UNIVERSITÀ POLITECNICA DELLE MARCHE

PHD RESEARCH SCHOOL IN ENGINEERING SCIENCES

INDUSTRIAL ENGINEERING CURRICULUM MECHANICAL ENGINEERING

A framework for an optimized product configuration

Author:
Miriam NARDELLI

Advisor:
Prof. Michele GERMANI

Curriculum Supervisor:
Prof. Roberto RAFFAELI

XXXII cycle

YEAR 2021

Declaration of Authorship

I, Miriam NARDELLI, declare that this thesis titled, "A framework for an optimized product configuration" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:
Miriam Nardelli

Date:
February 9, 2021

Acknowledgements

This work started in November 2016 as PhD course on “Mechanical Engineering” held at Polytecnic University of Marche, in Ancona, Italy. My work was carried out in the Mechanical Department of the Faculty of Engineering, under the supervision of Professor Michele Germani. This research has been possible thanks to the partnership between the academia and the industrial world.

This thesis contains the results of my activity in developing methodologies and frameworks for supporting design of Engineering-To-Order products. At the end of the course I am happy with what I have learnt and the results I have reached.

Firstly, I would like to thank my PhD advisor, Professor Michele Germani. In these years, I could see his enormous devotion for his work and I could admire his behaviour in human relationships.

A special thanks to Professor Roberto Raffaeli who followed and support me step by step in the development of this work. Thanks for the time he dedicated to me.

A special thanks also to Professor Paolo Cicconi who followed me especially during the first years of my PhD. Thanks for the time he spent to me.

I thank both Professor Raffaeli and Professor Cicconi for the stimulating discussions that led me to leave the purely mathematical field, from which I came, and introduced me to the engineering field.

Thanks to Hyperlean SRL that co-financed my doctoral scholarship. During these years the colleagues made me really feel part of the company, especially Silvia, Marco, Andrea, Christine, Angelica, Serena, Daniele, Matteo, Irene, Samuele and Eva.

Thanks also to my husband Fabio for always believing in me and for his fundamental support.

Thanks to my daughter Irene that shows me everyday the meaning of unconditional love and amazement for little things.

Finally, thanks to my families, my actual family, Fabio and Irene, my origin family, papà, mamma, Sara, Roald and Vanessa and the origin family of my husband, Tiziana, Claudio, Sabrina, Lorenzo e Anna, for being always with me.

Miriam Nardelli

Ringraziamenti

Questo lavoro è iniziato nel Novembre 2016 come un corso di Dottorato in "Ingegneria Meccanica" tenuto all'Università politecnica delle Marche, in Ancona, Italia. Il mio lavoro è stato portato avanti nel dipartimento di Meccanica nella facoltà di Ingegneria, sotto la supervisione del Professor Michele Germani. Questa ricerca è stata possibile grazie alla collaborazione tra l'univerità e il mondo dell'industria.

Questa tesi contiene i risultati della mia attività nello sviluppare metodologie e strumenti per supportare la progettazione di prodotti Engineering-To-Order. Alla fine del corso sono felice di ciò che ho imparato e dei risultati che ho ottenuto.

Vorrei prima di tutto ringraziare il mio PhD advisor, il Professor Michele Germani. In questi anni ho potuto notare la sua grande dedizione al lavoro e ammirare il suo comportamento nelle relazioni umane.

Un grazie speciale al Professor Roberto Raffaeli che mi ha seguito e supportato passo dopo passo nello sviluppo di questo lavoro. Grazie per il tempo che mi ha dedicato.

Un grazie speciale anche al Professor Paolo Cicconi che mi ha seguito specialmente nei primi anni di dottorato. Grazie anche a lui per il tempo speso per me.

Ringrazio sia il Professor Raffaeli che il Professor Cicconi per le discussioni stimolanti che mi hanno portato a uscire dall'ambito prettamente matematico, da cui provenivo, e mi hanno introdotto nell'ambito ingeneristico.

Grazie ad Hyperlean SRL che ha co-finanziato il mio dottorato. Durante questi anni i colleghi mi hanno fatto sentire pienamente parte dell'azienda, specialmente Silvia, Marco, Andrea, Christine, Angelica, Serena, Daniele, Matteo, Irene, Samuele e Eva.

Grazie anche a mio marito Fabio per credere sempre in me a per il suo supporto fondamentale.

Grazie a mia figlia Irene per mostrarmi ogni giorno il significato di amore incondizionato e lo stupore per le piccole cose.

Infine, grazie alle mie famiglie, la mia famiglia attuale, Fabio e Irene, la mia famiglia di origine, papà, mamma, Sara, Roald e Vanessa e la famiglia di origine di mio marito, Tiziana, Claudio, Sabrina, Lorenzo e Anna, per essere sempre con me.

Miriam Nardelli

Abstract

Nowadays market globalization has increased competitiveness and has forced companies to modify their design and production strategies. A key point is the development of products that respect and optimize as close as possible the individual customer needs (technical requirements) reducing at the same time cost during the phase of quotation. Industries specialized in products highly customized are called Engineering-To-Order (ETO) industries. The design of complex products, such as ETO ones, often requires the study of sub-problems. This practice is called modularization and it is one of the most applied design methods for ETO systems; however, it is necessary to integrate traditional tools with practices of design optimization to improve the development of a proposal managing the complexity. This integration is an interesting issue in engineering design. Design optimization, including configuration, is a common practice in industry 4.0 widely applied in civil and mechanical engineering. Different mathematical algorithms have been developed to support the optimization in engineering design.

Different commercial tools are available to support engineers during the design phase; however, many research scenarios are opened to find tools and methods to improve the design process and overcome some limits related to configuration and design optimization. The first difficulty is the will of company to formalize its internal knowledge, this can allow the past cases retrieval. One other limit is that the integration of geometrical modeling, simulations, analysis, and optimization concerns the interaction between different tools not linked with each level of the design phases. Moreover, these tools often require software customization.

The study of Constraints Satisfaction Problems(CSP) is a mathematical topic which can be applied for solving these engineering issues including also studies about energy efficiency and consumption of products, to enhance the product performance and reduce the cost, lead time and environmental impacts related to the product life-cycle. The strength of this approach is the speed on searching possible solutions.

This thesis proposes a design methodological approach to highlight how a CSP analysis can support the first phase of an optimization analysis, reducing the design space of solutions to be investigated and subsequently optimized. Moreover in this context is described a design tool that integrates a CS tool with model-based simulations in a collaborative design context.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	**A**rtificial **I**ntelligence
API	**A**pplication **P**rogramming **I**nterface
ASP	**A**nswer **S**et **P**rogramming
ATO	**A**ssemble-**T**o-**O**rder
BOM	**B**ills **O**f **M**aterial
CAD	**C**omputer **A**ided **D**esign
CAE	**C**omputer **A**ided **E**ngineering
CBR	**C**ase-**B**ased **R**easoning
CDS	**C**omputational **D**esign **S**ynthesis
COP	**C**ar **O**perating **P**anel
CSP	**C**onstraint **S**atisfaction **P**roblems
CTO	**C**onfigure-**T**o-**O**rder
DFMA	**D**esign **F**or **M**anufacturing and **A**ssembly
DL	**D**escription **L**ogics
DP	**D**esign **P**latform
DSM	**D**esign **S**tructured **M**atrix
DTC	**D**esign **T**o **C**ost
ETO	**E**ngineer-**T**o-**O**rder
ERP	**E**nterprise **R**esource **P**lanning
FBC	**F**eature **B**ased **C**osting
FEM	**F**inite **E**lement **M**ethods
GA	**G**enetic **A**lgorithms
HC	**H**ybrid **C**onfiguration
ICAD	**I**ndustrial **C**ontrol **A**nd **D**esign
KB	**K**owledge **B**ased
KBE	**K**owledge **B**ased **E**ngineering
KBS	**K**owledge **B**ased **S**ystems
LCC	**L**ife-**C**ycle **C**osts
LIP	**L**anding **I**ndicator **P**anel
LOP	**L**anding **O**perenting **P**anel
MFD	**M**odular **F**unction **D**eployment
OCL	**O**bject-**C**onstraint **L**anguage
OO	**O**bject-**O**riented
OOP	**O**bject-**O**riented **P**rogramming
PLM	**P**roduct **L**ifecycle **M**anagement
RFP	**R**equest-**F**or-**P**roposal
SAT	**B**oolean **S**atisfiability **P**roblems

SiB	**S**mall **i**s **B**etter
TRIZ	**T**heory of **I**nventive **P**roblem **S**olving
UML	**U**nified **M**odeling **L**anguage

To my whole family

List of publications

- Cicconi, P., Nardelli, M., Raffaeli, R., & Germani, M. (2020). *"Integrating a constraint-based optimization approach into the design of oil & gas structures".* Advanced Engineering Informatics, 45, 101129.

- Cicconi, P., Manieri, S., Bergantino, N., Nardelli, M., Raffaeli, R., & Germani, M. (2020). *"A constraint-based approach for optimizing the design of overhead lines".* International Journal on Interactive Design and Manufacturing (IJIDeM).

- Cicconi, P., Landi, D., Russo, A. C., Nardelli, M., Raffaeli, R., & Germani, M. (2018). *"A CSP-based design framework for appliances under energy labelling".* International Journal on Interactive Design and Manufacturing (IJIDeM), 12(4), 1243-1263.

- Nardelli, M., Cicconi, P., Raffaeli, R., &Germani, M. (2018). *"Supporting design tasks through constraint satisfaction tools".* In 15th International Design Conference-DESIGN 2018. The Design Society.

- Cicconi, P., Nardelli, M., Raffaeli, R., & Germani, M. (2018). *"A Design Methodology for a CSP-Based Optimization Approach".* In 25th ISPE Inc. International Conference on Transdisciplinary Engineering (Vol. 7). IOS Press.

- Nardelli, M., Cicconi, P., Savoretti, A., Raffaeli, R., & Germani, M. (2019, July) *"A knowledge based approach to support the conceptual design of ETO products".* In Proceedings of the Design Society: International Conference on Engineering Design (Vol. 1, No. 1, pp. 2417-2426). Cambridge University Press.

- Cicconi, P., Mandolini, M., Nardelli, M., & Raffaeli, R. (2019, September). *"Design Optimization: Tools and Methods for ETO Products".* In International Conference on Design, Simulation, Manufacturing: The Innovation Exchange (pp. 516-527). Springer, Cham.

Chapter 1

Introduction

1.1 Context of the thesis

Nowadays the market globalization, especially in the context of Industry 4.0, increases competitiveness and forced the industries to modify their design and production strategies [43]. Products loose their value very fast. Quality and excellence immediately start decreasing due to other competitive products on the market. Only with continual improvements companies can extend their products life, trying to adapt them to customer needs as much as possible. Therefore there is a trend of moving from mass production to mass customization. In this context, two types of approach can be compared to achieve specific customer's requests: Configure-To-Order (CTO) and Engineer-To-Order (ETO). Both approaches require a customer's order. While a CTO production is based on already existing configurations ([123], [90]), an ETO approach requires the development of engineering solutions before product fabrication [57]. ETO applications are necessary when customer's requirements cannot be fulfilled through a catalog of pre-configurated solutions. An important issue in ETO companies is the time necessary to provide a bid for the proposal submission [122]. Long engineering lead-times significantly affect the development and the cost of ETO products.

In the last 20 years, a lot of research has been paying attention to methods for reducing time and cost in the design of ETO products. Research topics concern Design for manufacturing and assembly [45], Feature Based Costing [75] and Design To Cost [33]. However, their application in industry is limited by the complex data analysis and the required knowledge formalization. On the other hand, other research issues such as Knowledge-Based Engineering (KBE) [65], Artificial Intelligence (AI) [7], and Object-Oriented (OO) design [91] are most applied in the industrial context. In fact, these methods enhance the reuse of design knowledge by automating repetitive tasks and optimization stages.

The trend of moving from mass production to mass customization changed production mentality approach in the way shown by Figure 1.1.

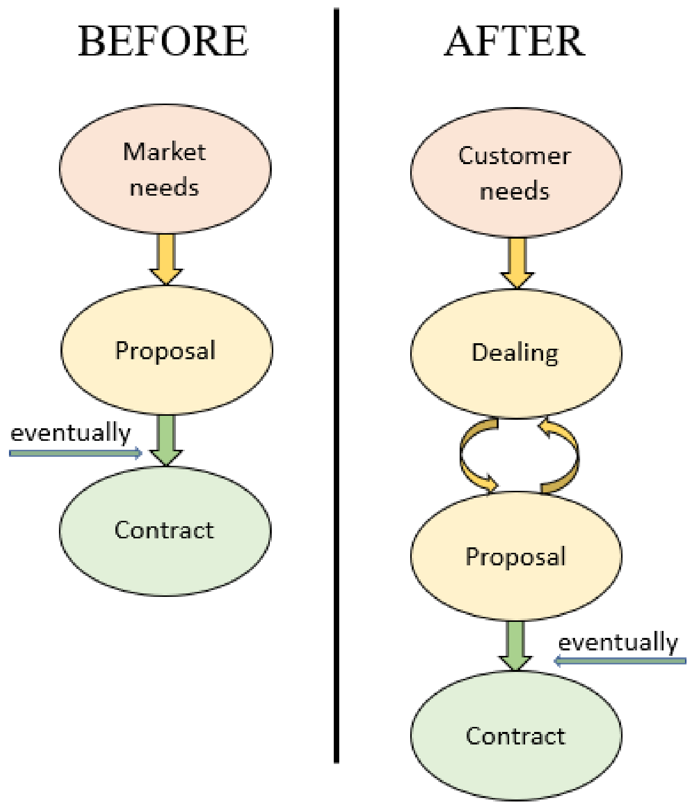


FIGURE 1.1: From mass production to mass customization: the change in the production mentality approach

In the first stage, it is important to prepare a reliable offer, this means to identify requirements and constraints within the Request-For-Proposal (RFP). This stage that in the past collected market requests now more often collects customer requests (ETO design). The RFP phase starts the early design phase. This phase is a preliminary analysis that has to take into account the draft project including geometry, layout, simulations and cost estimation. In the early design phase some activities require a priori choices, which lead to different alternatives depending on the chosen decision path. Moreover, several constraints, e.g. weight and overall dimensions, will be satisfied only after some activities of the design process have been performed; this complication leads to a rough configuration process that implies an inaccurate cost estimation. In fact, the cost estimation, assessed after the conceptual design stage, is mostly performed in an empirical and unstructured way.

Once the early design, called conceptual design, and the cost estimation stages are completed, the technical offer (the overall simplified product and often a drawing) inclusive of the economic offer, can be sent to the possible customer. In general, the customer evaluates the technical-economic offer, often comparing it to different offers of other possible suppliers. After this comparison many product characteristics and requirements are more precisely defined. This comparison that adjusts the proposal after dealing between supplier and customer is a technical alignment stages that requires much time and effort: product design and costs must be accurately revised. The company should answer to the new requests with another new offers, until the customer accepts the offer, which becomes a contract.

The awareness of these problems has led, from the 80ies, to the definition of some methodologies to assess cost before product realization to optimize product and process design, such as design for manufacturing and assembly (DFMA) approach [14], feature based costing (FBC) [120], design to cost (DTC) [33]. However the complex data analysis, knowledge structuring and the numerous resources to be involved in the project make their application in industry very difficult. On the other hand, different approaches as knowledge based engineering (KBE) [65], computational design synthesis (CDS) [20], artificial intelligence (AI) algorithms [72], object-oriented (OO) design and functional programming [91] have tried to standardize methods to easily capture, structure and reuse the design knowledge in a fast and easy way, by automating repetitive tasks and optimization stages.

Although there is the possibility of using such tools, it has been observed that many companies just base the process working by analogy on the basis of the expertise of senior designers and searching for similar past solutions. Then, product Bills of Material (BOM) is adapted and costs are updated accordingly. Such an approach mostly leads to some inaccuracies in the cost estimation, especially for ETO products. In fact, it has been seen that these products need a preliminary design phase, i.e. the conceptual design, before evaluating the product cost.

According to Pahl et al. Figure 1.2 it is useful to divide the planning and design process into four main phases: (1) planning and task clarification, (2) conceptual design, (3) embodiment design and (4) detail design; pointing out that it is not always possible to draw a clear borderline between these main phases, neither it is possible to avoid backtracking [79]. Thus, engineering design is a highly iterative process.

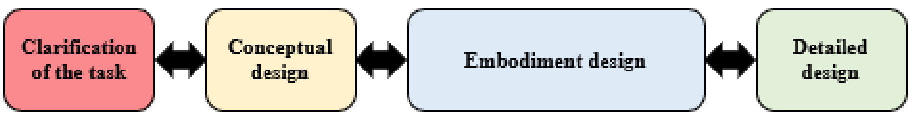


FIGURE 1.2: Design phases

Phase (1) leads to the product requirements list, the Request-For-Proposal (RFP) cited before, the later stages are based on this phase. While the conceptual design stage has to define the main functional solution, in a simplified way (concept); the embodiment design must specify the layout. Indeed, in many cases, also during the conceptual design stage, a preliminary layout assessment solution has to meet the customer requirements. Moreover, the awareness that decisions made in the conceptual design phase have a huge impact on the life-cycle costs (LCC) of the product, forced the designer to shift the knowledge required typically in the later stages (embodiment and detailed design) to the earlier stage (conceptual design) [23]. From this follows the importance of reusing the design process knowledge formalization at the conceptual design stage.

Beyond this aspect, conceptual design is also a very sensitive phase since requires cost estimation to evaluate variants against economic criteria. During the offer stage, customer requirements may change due to a negotiation phase in which costs and technical requirements are involved. In these cases, a rapid evaluation of the variants and their costs are needed.

In order to get rid of problems related to the conceptual design phase, exposed before, the industrial research gives a huge importance to modularity in particular starting from the 80ies and the 90ies.

Nowadays, the most used design strategies in ETO companies are Design Automation [89], Modularization [99] and Configurations [57]. To reduce time and cost in design phases and fabrication, configurations tools have been also implementing in ETO companies [106]. However, these tools are very difficult to be implemented if they have to provide new product solutions.

Despite such tools, it has been observed that many companies just base the process on poor empirical models, working by analogy on the basis of the expertise of senior designers and searching for similar past solutions. Formal knowledge is embedded from product documents, drawings and engineering dimensioning algorithms, while tacit knowledge, which is made of implicit rules, comes from the experience of people with technical expertise.

Knowledge representation can be classified into five categories: pictorial, linguistic, virtual, algorithmic and symbolic [78]. Chandrasegaran et al. show the various forms of knowledge involved at each stage of the design process and state that in the early design stages, knowledge representation is predominantly linguistic and pictorial [23]. This also explains the lack of tools able to support the conceptual design stage.

Knowledge formalization is a critical issue which involves capturing, representing, and reusing of past design cases [115]. Moreover the transfer of the knowledge about design requirements and constraints form senior designers to junior designers risks to be imprecise, approximate or incomplete [117].

So industries need a tool that should manage the company knowledge on the product life cycle, particularly the product design and manufacturing knowledge. Knowledge should be elicited and formalized, so that it can allow the past cases retrieval and could be a connection between customer specifications and the product configuration. Finally, the tool should allow a rapid definition of a new product configuration and its costs.

KBE applications are helpful to automate the design process avoiding the subjectivity of the cost estimator (1) and allow saving time for offer generation (2). KBE systems use stored knowledge for solving problems in a specific domain, making use of inference mechanism to answer the problem.

Even if methods have been already analyzed in literature for rapid product configuration and early cost estimation, [87] a lack of tools still exist in the context of ETO products, where traditional tools are not suitable to support the design activity including new tasks such as parameters optimization [30] and life cycle analysis. This thesis proposes an approach for the rapid definition of the product structure related to a new ETO product, including the early cost evaluation in configurations. The motivation of the work arise from the need to face the following cost estimation problems:

- Final cost is strongly affected by the subjectivity of the cost estimator.
- Elaboration of the technical proposal and the economic offer is time consuming.
- Expertise in a wide area is required.
- Difficulties to pass knowledge from senior designers to junior designers.
- Difficulties in retrieving information of past cases.
- Difficulty of connecting information, for example customer specifications and BOM.
- Difficulty of making the most economic choices during the product configuration.
- Difficulty of taking into account the product life-cycle within the offer.

The above problems lead to the need of a method and a tool of product configuration and early cost estimation that can be useful in different scenarios of a company: for providers and customers but also for different areas of the company itself like technical and commercial area.

The optimal solution, that is obtained from the product configuration, is that characterized from the best cost, the best time of production and the best performances.

In order to make the most economic choices, cost estimation and optimization criteria should be made step by step during the configuration process, being aware of the economic impact of every design choice, and should also guide the cost assessment by considering the entire product life-cycle [73]. Moreover, design choices should not be guided only by economic criteria, in fact several parameters should be considered during product configuration, starting from the conceptual design stage and not only during the embodiment design phase.

1.2 Research goals

In this context, the main focus of the research have been concentrated in developing a methodology and a related framework to support early cost estimation of engineering-to-order (ETO) products throughout the design phases; in particular on reducing time and effort in the offer stage, which allows a company to properly manage a greater number of offers in less time. During these three years the attention has been posed on topics like as optimization, modularity, product architecture, configuration and change propagation issues along the design process. In order to apply abstract concepts to design practice different approaches and tools have been proposed in literature; anyway concrete software solutions and application examples are still lacking or incomplete.

This work started with the understanding of the efforts and time consumed by the design of the draft project (offer stage) and the awareness that efforts and time spent could be not remunerated by the customer order. Consequently, the main focus of the research is on reducing time and effort in the offer stage, which allows a company to properly manage a greater number of offers in less time. Moreover configuration nowadays still closed principally to the technical department scenario (both for supplier and customer company) and this is also a limitation that not includes commercial and marketing departments. In facts often configuration softwares are not able to support an historical of orders and quotations, for example, and they are hardly able to communicate with the most common company tools. Finally during the configuration phase often it is not possible the human intervention that could help in particular to solve sometimes problems easy for humans and difficult for the machines. The awareness of these issues, which comes from the industrial world, is well documented in literature. The objectives of this work derived mostly from the results of an interview proposed and experimented in different companies that for each company reports the most important needs regarding the offer stage and the cost estimation.

The methodology and software focus but it is not limited on mechanically based products that include also non-mechanical systems, i.e. electrical, hydraulic, electronic, software and automation.

The framework is a computer-aided-design-based system that enable designers to easily represent the product platform, to structure the relations between modules defined at different levels of detail and, hence, to simulate, analyze and evaluate the impact of modifications during new product variant design activity. Moreover the design platform aimed at building a company knowledge repository, which can then be used to design new products and estimate their costs and at supporting the user in finding solutions for new products having different requirements. In particular, the support towards the solution aims to simplify the management of complex design problems characterized by a great number of variables to be assigned and by several design choices to be done. The support focuses on dividing the complex problem into more sub-problem more manageable (modules) and on identifying the design choices to be taken at the early stages. This knowledge acquisition differently from the past aggregates micro-contents both from engineering and programmers experts. This is an important novelty because these two areas are separately aware of the issues in their field but hardly communicate each other.

A rule-based method, a 3D product modeling environment and a database are the tools used to manage the framework, to achieve a graphical representation of the product variant architecture and to menage orders and quotations.

As regards the cost estimation, the aim is to extend the evaluation to the entire product life-cycle. In fact, the objective has been to go beyond the manufacturing costs, developing cost models that include product design, maintenance and running costs.

The research goals can be summarized in the following points:

- Development of a methodology able to support the designer from the configuration to the cost estimation phase in particular reducing time and efforts during the offer stage.
- Development of a framework able to support the methodology.
- Application of the framework in two industrial case studies.

1.3 Thesis overview

This thesis consists of three main parts. The first part, which corresponds to chapter 2, is the theoretical foundation of the research area. Literature about industry 4.0, knowledge based engineering (KBE), computational design synthesis (CDS), modularization and configuration, engineering-to-order products and their design, design structure matrix (DSM), constraints satisfaction problems (CSP) and optimization analysis has been reviewed.

Chapter 3 focuses on the definition of a method that support the designer towards an optimal solution. This is the actual contribution of the thesis: a methodology aiming to support the conceptual phase of product design or modifications during redesign process.

In chapter 4 and chapter 5, the final part, is presented the framework used for achieving the research goals above discussed. This is splitted in two chapters because in one it is described the early implementation and in the other the final development. The early implementation concentrates on the Design Optimization and has been preparatory for the final development that instead concentrates on the Design Configuration. The solver of the configuration problem implemented in the first implementation phase has been necessary to test and validate the results. Then this solver has been improved and has been posed into a more general context, the final development, even more adapt to the needs of a company that produces highly customized items. In this two chapter the case studies used to validate the method are also presented. For the early implementation is presented the optimization of an oil and gas structure; for the final development is presented the case study of a cabin operating panel system.

The definition of an approach and the correlated framework is the result of a broad literature review and the direct experience in developing such systems in collaboration with a company that has contributed to this doctoral scholarship, Hyperlean s.r.l., that is specialized in software solutions that support the product life-cycle management.

1.4 Author contribution

This thesis includes methods for supporting design process especially of modular products in the ETO context. The following features are believed to be original:

- Elaboration of an approach to formalize the knowledge of a company that includes more company scenarios.

- Elaboration of a methodology to represent modular product structure supporting product modifications unifying human and machine computation that includes a Design Structured Matrix (DSM)-based method assisted by the use of constrain satisfaction problems (CSP) to support the user in finding product modules.
- Elaboration of a configuration framework to be used as bases to develop configuration softwares in the context of modern design supporting systems and that is possible to be integrated to the most common company tools.
- Utilization of simplified 3D schemes as a means to represent product layout and also to manage product graphical modifications.
- Developing of customized solutions in some industrial contexts in order to reach efficiency and time/costs reduction in design and commercial department.
- Elaboration of an early cost estimation model that considers the product life-cycle.

Chapter 2

Background research

This chapter provides a background for the tools and methods used in this thesis and gives a literature review regarding similar approaches already existing.

The first sections of this chapter (Industry 4.0, Engineering-To-Order design, knowledge based engineering, conceptual design synthesis, configuration and modularization) constitute the theoretical background for the company knowledge formalization and representation described in the section of the method. The last sections (design structure matrix, constraints satisfaction problems and optimization analysis) describe some tools that have been used to support the design solution search.

2.1 Industry 4.0

The concept of industry from the first industrial revolution to the present days has undergone profoundly changes. Today we have arrived at the concept of the fourth industrial revolution and the current period of industrialization is called Industry 4.0.

A brief history of the industrial revolutions has been presented in Figure 2.1.

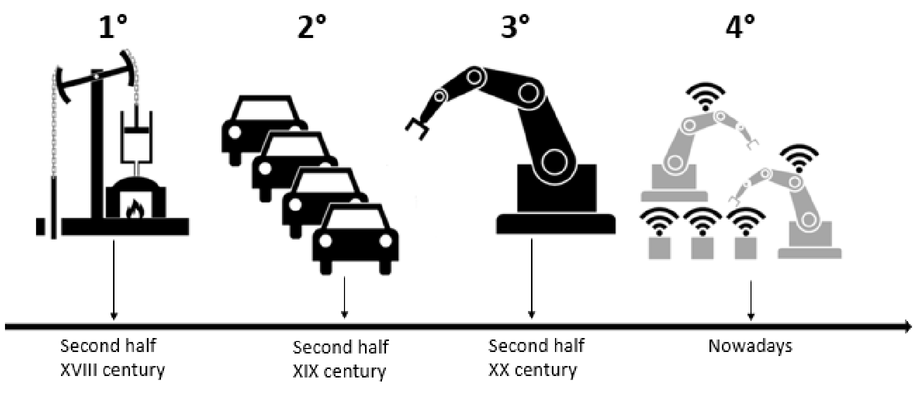


FIGURE 2.1: Industrial revolutions

The first industrial revolution starts in the second half of the eighteenth century and has been a process that industrialize the society from a prevalent context agricultural and artisanal introducing modern machines who worked with fuels obtained from new energy sources as water and fossil fuels. The sector most interested has been the textile-metallurgical one with the introduction of the flying fuse and the steam engine. This first revolution profoundly changed the socio-economic aspect of society at the time.

The second industrial revolution starts in the second half of the nineteen century with the introduction of electricity, chemicals and oil. During this period the areas of transports, communications and general production has been enormously modified to respond to the requests of the world commerce of mass production. The mounting chain is proper of this revolution.

In the second half of the twentieth century the advent and massive introduction of electronics, telecommunications and information technology in industry bring the world to the third industrial revolution. This revolution affects profoundly the manufacturing sector and starts to pose the world attention on pollution and green economy.

Nowadays the world has arrived at the fourth industrial revolution also denominated Industry 4.0. This revolution has a trend in industrial automation that integrates some new production technologies to improve working conditions, create new business models and increase the productivity and production quality of plants.

The name Industry 4.0 spread from an European initiative inspired to a German project that included investments in infrastructures, schools, energy systems, research institutions and companies to modernize the German production system and bring German manufacturing back to the top of the world, making it competitive globally [54].

The success obtained by the Germany persuaded many other Nations to pursuit this policy. However, several studies have estimated that the automation of different jobs will lead to a decline in employment of around 7 million jobs compared to just 2 million new jobs. The areas most affected will be administration and production. The losses will be partially offset in the financial, management, IT and engineering areas [10].

The Industry 4.0 is based on nine technologies defined pillars[98]:

- Advanced manufacturing solution: interconnected and modular systems that permit flexibility and best performances;
- Additive manufacturing: production systems which decrease the waste of materials;

- Augmented reality: vision systems to guide better operators in carrying out daily activities
- Simulations: simulations between interconnected machines to optimize the process;
- Horizontal and vertical integration: information exchange between all the departments of production process;
- Industrial internet: use of the internet for communication between elements of production;
- Cloud: implementation of all cloud technology like cloud storage including also the management of big data through open systems;
- Informatics security: open systems need to have an high degree of security since they have to respect all the company policies;
- Big data analytics: analysis of the big data in the open systems to permits forecasts.

As part of the Industry 4.0 modularity and configuration find a further importance than in the past, as the trend is towards highly customized products with also the end user participating in the product definition process. These two technologies, both used in this thesis, are parts of the pillars advanced manufacturing solution and horizontal and vertical integration since the last tool implemented is an interconnected and modular system that guarantees flexibility and best performances and permits the exchange of information between all the department of production process.

2.2 Engineer-to-order products

Engineer-to-order is a production technique that starts upon receipt of a customer order, whose engineering requirements and specifications are not known in detail. There is a substantial amount of design and engineering analysis required, these additional activities need to be added to product lead time.

ETO is essential for those companies with customers needing solutions that are expected to fit their own unique environment. It begins with concepts that do not have fixed designs and ends in a new, unique product. This could be any product, from special aircraft to a pair of jeans. But the typical ETO environment usually deals with the design and build of unique custom engineered complex machinery and industrial equipment where disciplines like mechanical, electrical, mechatronics, soft-ware, manufacturing and systems engineering are heavy involved. A business strategy based on ETO products often requires design tools to an efficient generation of the product variants [34]. Ready-to-use softwares do not exist since for every customer they will require a new customization. In this way the software will be such expensive that will be more convenient to design products without its help.

The optimization of ETO products is an interesting topic for the engineering world even if it carries issues as time consumption, subjectivity and not replicability. In fact, during the conceptual phase, the designer has to face the satisfaction of all the technical and geometrical request of the customer. The principal problem is that the final product is often defined by the experience of the designer that has to solve models which leads complicate time consuming routines without many helps from software's. Sylla et al. review recently this question focusing on how to extend configuration from Assemble-To-Order (ATO) products, that is a more standard process, to Engineer-To-Order (ETO) situations [106]. Companies that produce customized and modularized products [15] often reuse and adapt past solutions. The sales-delivery process of ETO products presents a considerable potential for design reuse [15]. As analyzed by Sabin and Wiegel, at the end of 90ies, the design of ETO variants can be seen as a case-based reasoning (CBR) process, where each new configuration problem was an adaptation solution of a product family model [93]. As a second step, modularity was applied in this context to reuse predesigned modules [104].

Johansson studied a method that could help manufacturing companies to manage and to reuse the engineering knowledge related to the ETO products [52]. Algorithms were implemented to retrieve the engineering knowledge from Computer-Aided Design (CAD) models and spreadsheets used in the design of the ETO projects. Camba et al. [19] developed methods to retrieve knowledge from CAD models by analyzing the dependencies between each component. A knowledge eliciting approach was also investigated to support the configurations of the virtual prototypes [86]. These approaches can be considered valid to support a connectivism in the context of the KBE; however, these approaches do not consider the relation between the geometrical configuration and results of the simulations. Even if the KBE tools are widely applied in this context, they are used only to manage the product configuration as a reuse system of the design knowledge. Therefore, such approaches work well with light level ETO products, where the term "light" means that the ETO product corresponds to a standard configuration with minor customizations. ETO solutions can be classified as "light" if concern minor customization or as "heavy" when the final solution must be completely engineered with a major level of customization [106]. As already highlighted by André and Elgh [5], the future reuse of models in the ETO approach needs to involve problem solving methods. Additionally, the design of the ETO systems should simultaneously involve the development of a transdisciplinary environment, which requires a cooperation between the design and analysis [5].

In the engineering world, products can be classified into two classes: Configuration to Order (CTO) products and Engineering to Order (ETO) products.

CTO products are created by vendor, the customer can only decide between a range of products but can not customize them. This type of service guarantees cheaper production costs but limits customer choice. Many softwares have been implemented to support this kind of production. some example of CTO products are shown in the Figure 2.2.

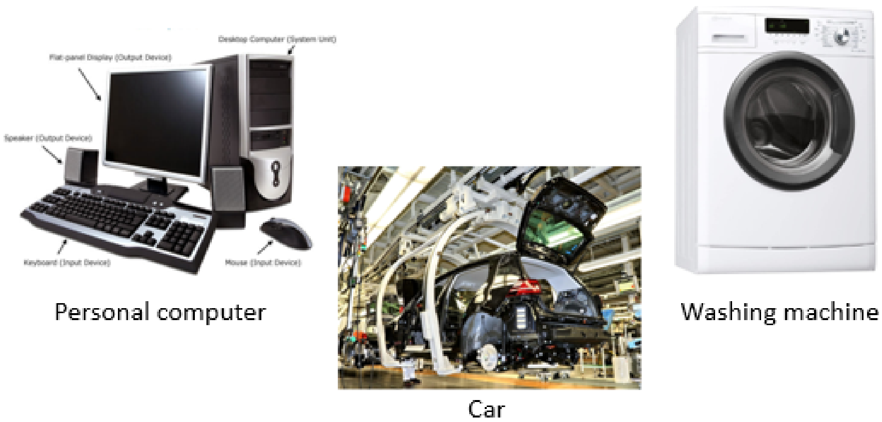


FIGURE 2.2: CTO products examples

ETO products are very different from CTO products due to the necessity of the engineering and detailed design phases. Figure 2.3 show some example of ETO products.

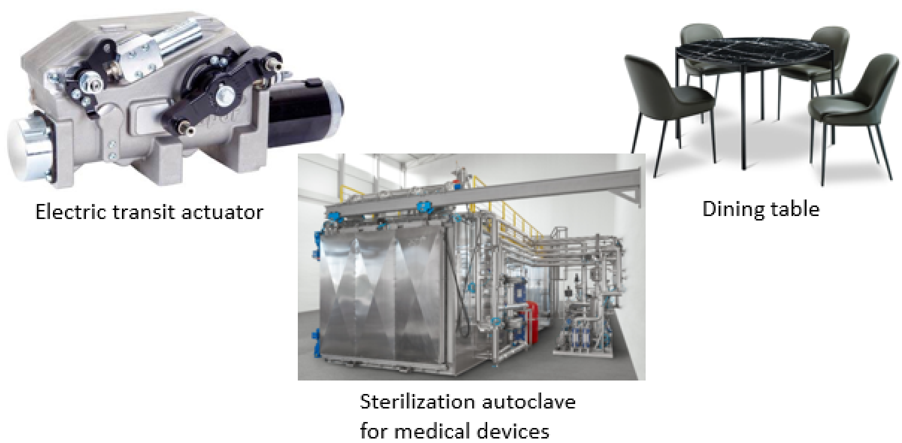


FIGURE 2.3: ETO products examples

The ETO production workflow (Figure 2.4) starts with a definition of all the technical and geometrical specifics by the customer. Then a prototype, physical or virtual, is realized in order to obtain a first cost estimation: this will consider the requirements that can be divided into product configuration and design specifications. On the prototype is made a performance analysis and, if it passes all the tests, this phase will end up with a proposal review. If the two parts, the vendor and the customer, agreed, the process of the embodiment design starts and finish with the field installation and the acceptance. The embodiment design consists in realize a CAD model that is modified until the customer does not accept the design. Then the model is realized and is installed. The product delivery process requires the design tools to support the decision making during the stage of the quotation preparation since an ETO company is often involved in many quotation processes [34]. In this context, when the design targets are not achieved, numerous time-consuming iteration loops are necessary to optimize the initial solution [81].

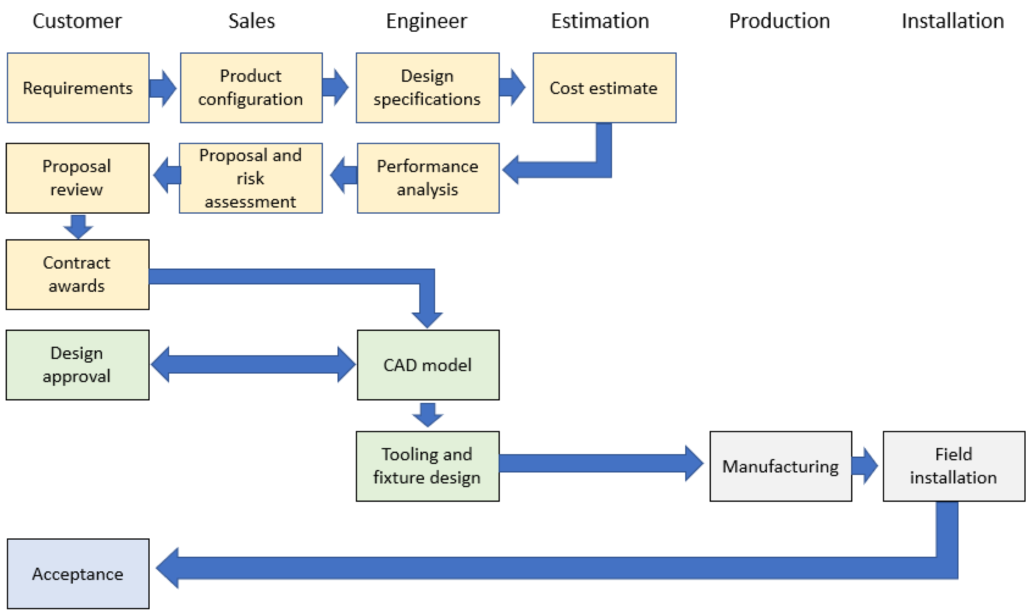


FIGURE 2.4: ETO production workflow

Several methods have been described in the literature to support the systematic design space exploration in the context of the model-based engineering. However, effective tools to capture, reuse, and represent the implicit knowledge in the exploration process of the design space are lacking. These tools must have the ability to learn from previous cases and developing knowledge based methods for guided assistance in decision making [118]. Gadeyne et al. described a graphical approach based on the SysML modeling language with the object constraints language for representing the model of the design space for a gearbox [41]. Their research aimed to support the design synthesis during the early embodiment design [9] to define and fix the product topology, components, architecture and connections. The definitions of these parameters at the early design stage exert a considerable influence on the final product cost [79]. Designers apply design rules, tables, formulas, and relations during the engineering design process. Artificial intelligence technologies such as rule based reasoning [24], constraint satisfaction problems (CSPs) [106], case-based reasoning (CBR) [126] are mainly adopted as the problem solving mechanisms for obtaining the configuration solutions. An ETO application was analysed by Cicconi et al. [30]. He proposes an approach to perform the configuration and the optimization of oil & gas ducts. However, his approach lacks a CSP module to reduce the variation of the parameters to be optimized. Lin et al. [63] proposed a TRIZ-based approach to support design when the solutions of optimization methods do not meet the objectives of problems to solve. His approach is focused on the analysis of system contradictions which extracts data from the simulations generated from the optimization loop.

All these studies highlight that the tools to integrate the configurations and optimizations into the same design workflow for products with a high level of customization are lacking. A reason is that a high level of customization cannot be managed using standard interfaces. The recent improvements in computational analysis have continuously enhanced the development of the design optimization practices. However, ETO structures require a design platform that can optimize the product at different levels, integrating several design tools to reduce time and cost.

2.3 Knowledge Based Engineering

An expert system (or Knowledge based systems KBS) is defined as a tool that stores and accumulates specific knowledge of different areas and generates solutions in a user interface to given problems [101].

Knowledge based engineering (KBE) is a subset of KBS and is a research field that comprehends methodologies and technologies to acquire, formalize and represent in an IT system, product and process engineering knowledge to achieve automation of repetitive design tasks and to implement multidisciplinary design optimization [115]. More specifically it tries to increase dynamic calculus of excels with generative geometry, ability to write reports, ability to increase automation and intelligence in CADs programs, teaching them design rules and engineering knowledge [91]. KBE has its roots in Artificial Intelligence (AI) and in Object-Oriented Programming (OOP) with a particular focus on product engineering design and downstream activities such as analysis, manufacturing, production planning cost estimation and even sales. The historical roots of KBE come from expert systems of the 1960s.

KBE typical application is in the design of aircraft. This is an application case that shows how this technique could be efficient in helping the designer to analyze and solve problems than must face frequently [129]. One of the hard challenges in the aircraft design is the modelling of the harness 3D routing. The difficulty stays in its intrinsic complexity, but also in the increasing of the design constraints and in its dependency on any little change. Zhu e al. [129] proposed methods based on KBE and optimization to reach minimum cost routing solutions that satisfy constraints of all relevant design rules.

In order to develop a KBE application, a previous step of knowledge formalization is required. It is important to collect both tacit and formal knowledge. Formal knowledge is embedded from product documents, drawings and engineering dimensioning algorithms, while tacit knowledge, which is made of implicit rules, comes from the experience of people with technical expertise. Knowledge formalization is a critical issue which involves capturing, representing and reusing knowledge of various forms. Several methodologies for knowledge capture and formalization are available in literature for KBE. One of the most well-known KBE methodologies is the methodology and software tools oriented to knowledge based engineering applications (MOKA). This methodology, based on eight KBE life-cycle steps and expressed in accompanying case-specific informal and formal models, is designed to take a project from inception towards industrialization and actual use [102]. Recently has been introduced the concept of Human computation [109] that can ease the knowledge formalization. Human computation makes arise constraints, that represents knowledge, gathering micro-contributions from experts.

A KBE system is a general-purpose tool which does not contain any knowledge about any specific domain. Developing a KBE application is mostly about writing code using a KBE programming language. State-of-the-art KBE systems provide the user with an object-oriented language, which allows modeling the domain knowledge as a coherent network of classes [91].

The language of KBE systems is characterized by a declarative style, i.e. the order in which object are declared is not relevant. Most of KBE languages are based on object oriented dialects of Lisp programming language [91] which was used in artificial intelligence research and implementation. The define-object operator is the basic means to apply the object-oriented paradigm in KBE application, allowing to define classes, sub-classes, objects and relationships of inheritance, aggregation and association.

One of the most important features of KBE systems is the integration with CAD environment, which allows to generate and manipulate geometry by means of the programming language. This reveals the close link between KBE and the fields of design automation and optimization. The focus of engineering design automation is on automating repetitive or routine design tasks, e.g. CAD models and dimensional drawings generation. For instance, Frank et al. [40] have presented a framework of engineering design automation for creating complex CAD models, which has been applied to some parts of cranes. Instead, the aim of design optimization is to find out a valid design solution, within a solution space, which minimizes or maximizes a given objective function (e.g. the weight of a steel structure).

Different contributions to design optimization problems have been provided, mostly regarding structural optimization. Cicconi et al. [29] develop a platform in which an optimization tool and a FEM (Finite Element Methods) analysis software have been integrated in order to support the automatic optimization of a steel structure. Solutions exploration has been done by means of a design of experiment (DOE) plan. In fact, DOE theory is a traditional way to explore possible solutions, which makes use of a multi-variation approach [68]. Verhagen et al. [115] made a review of KBE and identified the current shortcomings and the research challenges of this topic. In particular, the improvement of the methodological support for KBE has been identified as the most important research challenge.

2.4 Configuration

The configuration is defined as delivering customized products by means of mass/serial production, which is achieved by designing a configurable product, whose variants cover the needed variation. Configurable product is defined in configuration model. The configuration model represents the available components or modules, rules of their correct combinations, and rules on how to achieve the desired product properties for a customer.

The product configuration is an essential approach for selecting various components to constitute a customized product with the aim of meeting the specific requirements of a customer [127].

Configuration is a special case of design activity with two key features: the final product is assembled from instances of a fixed set of well-defined component types and the components interact with each other in predefined ways [93]. Component types are chosen from sets of alternative components, namely instances, characterized by properties. Constraints restrict the solution space, e.g. the way in which components can be combined each other. Sabin and Weigel [93] also assert that given a set of customer requirements and a product family description, the task of configuration is to find a valid and completely specified product structure among the alternatives the generic structure describes.

A configuration task also involves two different steps: representing the problem and finding a solution for the problem posed. Representing the problem refers to the description of the knowledge formalization, which is the most critical issue. The knowledge acquisition process provide the list of the selected component (BOM) as well as the product structure and topology (product architecture).

Configurators are the most common tools to support product configuration. They are softwares that guide the users through the configuration process, implementing rules and a knowledge base for a rapid selection of the most suitable product configurations [48]. Configurators are mostly thought to be used by customers for selecting the product that matches his needs, paying at the same time attention to make him aware of the costs of every choice made.

Generally, the CTO production is based on configurators because these tools support customers and designers in the selection of already defined solutions. The main advantage of this approach is the reusing of past solutions avoiding new design efforts related to simulations, engineering design, cost estimations, etc. In this context, products can be simple machines or even types of machinery. Felferning et al. [37] distinguish several types of product configurators basing on their different goals and on the level of configuration freedom. Otherwise, the design of more complex systems often requires an ETO approach with customization. The customization can be related to specific normative, boundary conditions, sizing, special requirements or other. Therefore, ETO products are often complex structures, types of machinery, and even plants. To reduce time and cost in design phases and fabrication, configurations tools have been also implementing in ETO companies [106]. However, these tools are very difficult to be implemented if they have to provide new product solutions. ETO configurators have a high level of complexity since the knowledge domain of ETO products is enormous and not fixed. In addition, the product configuration requires in these cases an engineering effort so that it is impossible to make the task automatic.

Each component is uniquely identified by its set of properties, describing its function and performance. Properties can be classified into attributes, resources and ports. Attributes are used to describe the features of a component, i.e. geometry, weight, functions. A resource is something provided by some components and consumed by others. Resources are used in configuration models to check for balance issues. Ports describe the way in which a component can interact with other components of the domain. Ports are used in a component to impose restrictions on the type and number of components that can be connected to it.

Artificial Intelligence (AI) techniques, like XCON system [64], have their foundations on configuration. XCON was a rule-based configurator, i.e.

production rules (if-then structures) were used to express configuration knowledge. The main problem of rule-based configurators is the interdependence between domain and problem knowledge, which causes enormous efforts in knowledge base development and maintenance [100]. The problems encountered with the use of rule-based configurators have resulted in the development of model-based configurators, in which the domain knowledge is completely separate from the problem-solving knowledge.

Nowadays, most of the configuration environments are component-based, e.g. SAP, Siemens, ConfigIT [67], EngCon [37], Tacton [76]. Moreover, configuration environments have been made more usable and are often integrated into enterprise resource planning (ERP Enterprise Resource Planning) systems, which are widely used in the industrial area.

Configuration is a research area that still alive also today, even if the scopes of the research are changed. Tiihonen and Felfernig [109] analyzed how personalization can be incorporated into mass customization. They arise the problem that mass configuration could confuse customer sometimes for the large variety of products available and sometimes because it not explained why during the configuration no product satisfies the filters (diagnosis is poor). In order to get rid of this uneasiness, personalization techniques should be applied. They introduce the concept of user model ranges that are capable of reduce the range of products to be presented to the customer. Examples of these techniques are collaborative filters that presents products already evaluated positively and content-based recommendation that analyzed purchased products from the customer and propose similar products. Another open point of configuration is how to get an open configuration, e.g. configuration should be open to more scenarios like seller, customer and technical sales. To answer to this last points Felfernig et al. [37] states that this can be possible if three concepts collaborate: community-based knowledge engineering, group-based configuration and flexible product enhancement. Community-based knowledge engineering is an approach that makes interacts between them engineers and programmers in order to complete each other knowledge. Group-based configuration expects that a group interacts in a configuration providing decision making mechanisms. Flexible product enhancement requires flexibility in the group decision making in order to get a configuration that respects, in part, all the opinions.

Finally also during configuration is desirable the human computation already introduced in Section 2.3 [109]. In the context of configuration it means that the user can interact during configuration if the process find a bottleneck that may became largely time consuming. The configurator proposes some alternatives that solve the bottleneck and the user should choose among these alternatives. This point, even if is in the last part of this section, is fundamental since traditional configurators and design synthesis systems are rigid and static. This thesis explores the possibility of a collaboration between designer and software in order to obtain more flexible solutions to enlarge the creativity space.

2.5 Computational Design Synthesis

Besides KBE and configuration, Computational Design Synthesis (CDS) is a research area focused on activities that automate the design phase in the production, generating automatically alternatives in order to avoid long and useless routine works [20]. Thus, a major goal of design synthesis is to support the creation of a large number of alternatives of high value, while in KBE and product configuration systems the product architecture is almost predefined within certain limits. In fact, the aim of CDS is to support designers at a crossroads without a known direction to pursue or when the range of possible solutions for a task is too wide to be graspable manually [18]. Some tools have been developed but they remain too time and memory consuming. Mueller et al. apply this method to design and optimize parts for additive manufacturing processes [70]. In his work he analyzed parameters of the 3D printing process and gathered the necessary information to understand variations of 3D printed structures. The resulting parameters led to build accurate part models and optimize, fabricate, and test them in the best way.

CDS is used in every Knowledge Based Engineering (KBE) application to automate design routines and to implement a multidisciplinary product design.

Chakrabarti et al. [22] made an overview of CDS research, it is divided in three major approaches: function-based synthesis, grammar-based synthesis and analogy-based synthesis.

Function-based synthesis includes several methodologies merged around the functional decomposition of solutions, taking designers through steps that help decompose a design problem and build conceptual solutions based on the product functionality. All these models try to find connections between functional aspects of product and behavioral, structural and environmental ones. In grammar-based synthesis, generative graph and spatial grammars are used to computationally encoding knowledge in order to create designs: it is defined a vocabulary and a rule-set that operates on it, the generated design language can be used to quickly generate standard or novel design alternatives.

For instance, Wyatt et al. [125] describe a product architecture as a network in which nodes represent components and links connections or other relations. The space of possible architectures is the complete set of possible combinations of elements that satisfy the constraints. Along the same lines, Müenzer [66] discusses an approach for the generic transformation of product concepts to Bond-graph-based simulation models in order to generate, explore and evaluate large solution spaces. The actual research in grammar-based synthesis is the automatic learning of grammar rules. In fact, engineering grammars would be more readily applied if rules could be learned and adapted on their own by the software.

Finally, analogy-based synthesis is defined as the cognitive process of transferring information from a particular subject to another one. In particular, the case base reasoning (CBR) is used to adapt old situations to new demands, old cases are used to interpret new solutions [56].

2.6 Modularization and product platform

This section reviews the concepts of modularity and product platform, which play a key role in mass customization strategies. The architecture of a product is the scheme by which the functional elements of the product (the individual operations and transformations that contribute to the overall performance of the product [114]) are arranged into physical blocks (parts, components, and subassemblies) and by which the blocks interact. Mostly, products belong to a family of products, i.e. product variants based on a common platform. Product family can be defined as a group of related products that share common features, components, subsystems and satisfy a variety of market requirements [36]. The industrial importance of modularity has grown in the last forty years, when mass customization has more and more replaced the traditional mass production.

Modularity is a fundamental part of the product platform thinking: it enables the development of product families using shared assets with a cost-effectiveness advantage [77]: it brings cost savings and other gains in terms of sourcing, manufacturing, and quality control.

Modularity is the standardization of building blocks in order to actuate product platforms. The first known type of modularity is the assembly based modularization where modules are physical assemblies. Having a modular structure helps to organizing the production, managing the life cycle of a product or managing the variants in product palette. However, the more interesting type of a product platform is formed of function-based modules. In customer variant product, the variation often is connected to product functionalities, so this kind of platform supports customer variation, which is an essential point in business today. Modules are defined as physical structures that have a one-to-one correspondence with functional structures [113]. Hölttä-Otto asserts that a module is an independent building block of a larger system with a specific function and well-defined interface. She also states that a module has fairly loose

connections with the rest of the system if the interconnections of the interfaces are carefully considered, this allows its independent development, outsourcing, manufacturing and recycling [50].
Modular products have been defined as machines, assemblies or components that fulfil various overall functions through the combination of building blocks (distinct function units) or modules [79]. Each module provides one or more well-defined functions of the product and includes several variants that deliver different performance level for the function(s) the product should accomplish.
These products can be modeled in 3D using product families in CAD tools. The use of a product family in the 3D modeling allows a wide range of variants to be generated by one unique parametric CAD document [124]. However, the definition of a product structure is still a challenging task in the modular design, as highlighted by Windheim et al. [124].
Modularity is also applied in ETO systems such as oil & gas plants [107]. A modular plant consists of many modules which include pre-fabricated units [44]. A dedicated design strategy is needed to exploit all the advantages offered by modularity in terms of cost and project risk reduction that must face frequently[129].

Regarding past research on ETO approaches, other topics have been investigated in the last two decades such as Design for Manufacturing and Assembly [45], Feature-Based Costing [75] and Design-To-Cost [33]. However, these applications for the ETO industry requires the definition of a Knowledge Base (KB) to formalize the implicit, explicit and tacit knowledge [65].

In developing a platform architecture, two alternative approaches can be used: a function-based approach or a component-based approach. While the first one takes a more general view to define common functions independent of any particular embodiment decision, the second one uses a priori knowledge of the most relevant components needed [77].

Höltta & Salonen [51] identified three systematic modularity methods: (1) function structure heuristic methods [103], (2) clustering of Design Structure Matrix (DSM) [83] and (3) the Modular Function Deployement (MFD) method [35].

The first method arises from the functional model of the product, that is defined following the guidelines of Pahl et al. [79]. A function structure is a functional decomposition, intended as a block diagram, both of all the functions of the product and also of the flows, of material, energy, and information, that are established between them. Functions represent the input/output relationships of a system, or of a product, that are intended to perform a task. A function is a pair of verb-noun, connected by flows of energy, material and signal. The functional structure of the product shown in Figure 2.5 is splitted into sub functions of lower complexity, which have to be connected through the flows of energy, material and signals.

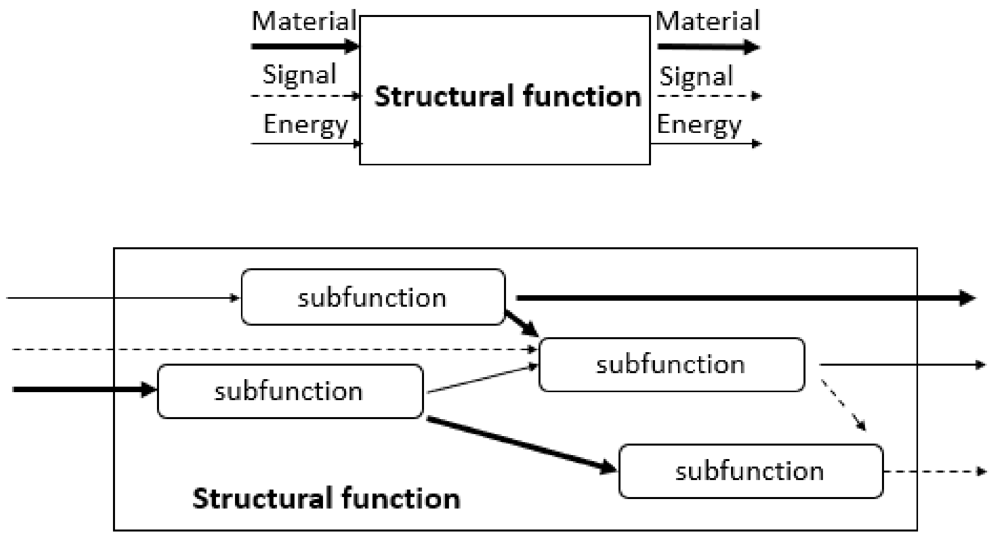


FIGURE 2.5: Scheme of a functional decomposition

Function-based representations are largely used during the conceptual design stage. Summers and Rosen [105] discuss three different modeling approaches with a focus on conceptual design and make a comparison of the types of information supported by these representations. The above cited function structure heuristic methods [103] are: dominant flow, branching flow and conversion-transmission pair; they identify groups of functions so that each group constitutes a module.

DSM is a method to manage complex system [114]: it can provide a compact representation of the product architecture in terms of relationships between its constituent components. Specific algorithms allow to identify modules within a DSM, a module in a DSM is a group of components, based on their interactions. While the function structure heuristic methods require function structures and are suitable for the function-based approach, DSM method can be applied without function structures using the component-based approach.

MFD [35] is also based on functional decomposition but in this method modularity drivers are also considered. Modularity drivers, like technology evolution, common units or styling, are then mapped against functions.

2.7 Design Structured Matrix

A design structure matrix (DSM) is a simple and compact visualization of a complex decisional system that supports innovative solutions to decomposition and integration problem [16]. DSM is a square matrix NxN used to represent the dependencies between N elements of a system. DSMs can be divided into static and time-based. Browning identifies four applications for DSMs:

- Component-Based or Architecture DSM (static): it organizes product architecture.
- Team-Based or Organizational DSM (static): it organizes teams of people.
- Activity-Based or Schedule DSM (time-based): it sequences activities.
- Parameter-Based DSM (time-based): it sequences parameters.

The component-based DSM is used to model system component relationships and to identify modules in product architecture. The team-based DSM is quite similar to the previous, but it is used to analyze inter-team interfaces, which provide the greatest leverage to improve the organization. Static DSMs are used for representing the interactions between elements, to identify several blocks of elements in which the interactions are maximized. Binary DSMs allow to represent only the existence of a relation, whereas numerical DSMs use a numerical value to show the strength of a relation. Once the components and their interactions are placed in the DSM, a clustering algorithm can be applied to group the components so that the interactions in the clusters are maximized and out of the clusters are minimized. Various clustering algorithms have been created with the goal of module definition in mind, e.g. Thebeau developed a clustering algorithm based on simulated annealing [108].

On the contrary, time-based DSMs are used to show dependencies between the elements of a system, i.e. activities or parameters. Time-based DSMs are generally binary where the presence of a 1 (or a X) in ij item means that the activity (or the parameter) i depends on the activity (or the parameter) j. If a 1 (or a X) is both in ij and ji items, the activities (or the parameters) are interdependent. Activity-based DSM is used to sequence a set of activities according to their dependencies and to identify blocks of activities which are affected by mutual dependencies. These activities must be solved in an iterative manner in order to proceed with the following activities. Parameter-based DSM is actually a low-level of the activity-based DSM. In fact, while an activity-based DSM can be used to model a design process, a parameter-based DSM can be used to represent the relationships between the parameters of the design process. In order to sequence an activity (or parameter)-based DSM, several partition algorithms can be found in literature, e.g. the reachability matrix method calls for finding a multi-level hierarchical decomposition for the matrix [119]. By means of partition tools, it is possible to reorder the DSM and eventually find the coupled blocks of parameters that are affected by mutual dependencies. Aggregation, decomposition and tearing are the possible alternatives to solve the coupled blocks [16]. While aggregation consists in collapsing two or more activities in order to remove interdependencies, decomposition attempts to explode coupled activities in lower-level activities which are not mutually dependent. Finally, tearing consists in removing dependencies and making assumptions in order to exit the loop of coupled activities (or parameters) and proceed with the process.

2.8 Constraint satisfaction problems

Generally, designers use and manage design rules, tables, formulas, and relations during the mechanical design process that can finally be explained by a set of rules. This set of rules in Mathematics is named Constraint Satisfaction Problem (CSP). A CSP problem could be defined as an object that is a mathematical model which reproduces the behavior of the physical system [47] represented from variables that must satisfy a set of constraints. The model to be applied is parametric; therefore, each variable can change is value in a defined variation range. A constraint on a set of variables is a restriction on the values that those variables can take simultaneously. Conceptually, a constraint can be represented by means of matrices, equations, inequalities or relations. These types of problems are tasks for research in Artificial Intelligence and Operational Research. They are extremely interesting in particular in the Computational Design Synthesis or in the Knowledge Based Engineering because they are a powerful mean to express engineering problems and pursuit solutions in product design, supporting the knowledge-intensive process [49]. There are numerous methods to solve this kind of problems and many other methods are studied in order to increase time speed and calculus precision. The more commons are backtracking, branch and bound and depth first search with their variations.

Backtracking is a technique that consists to search a solutions pruning the combination that do not satisfy some constraint [59]. It explores a graph tree remembering all the nodes already analyzed, in that way, if a path must be pruned, it can come back to the node that is not visited jet without the risk to move in a path already explored[112]. The backtracking algorithm has an exponential complexity, so it is not so efficient for not NP-complete problems[128]. Nevertheless, the algorithm integrates some heuristic techniques that allow to decrease complexity [25]. Figure 2.6 describes the backtracking search process.

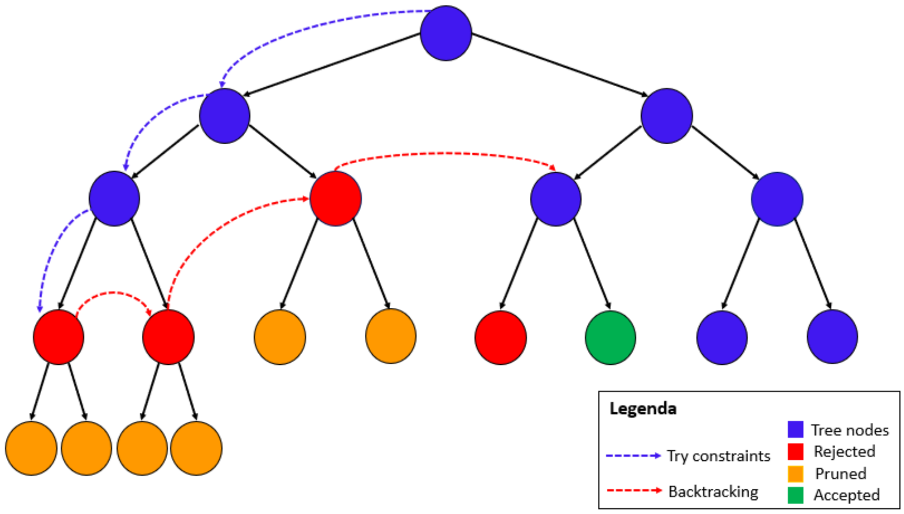


FIGURE 2.6: Backtracking search process

Depth first search is another technique of tree search, its particularity is that the algorithm can explore nodes far from the root without having visited the nodes of first generation [95]. The search strategy explores the graph reaching the deepest possible node in it. Once all the deepest nodes are analyzed, the algorithm comes back to explore all the previous ones. Figure 2.7 describes the depth first search algorithm.

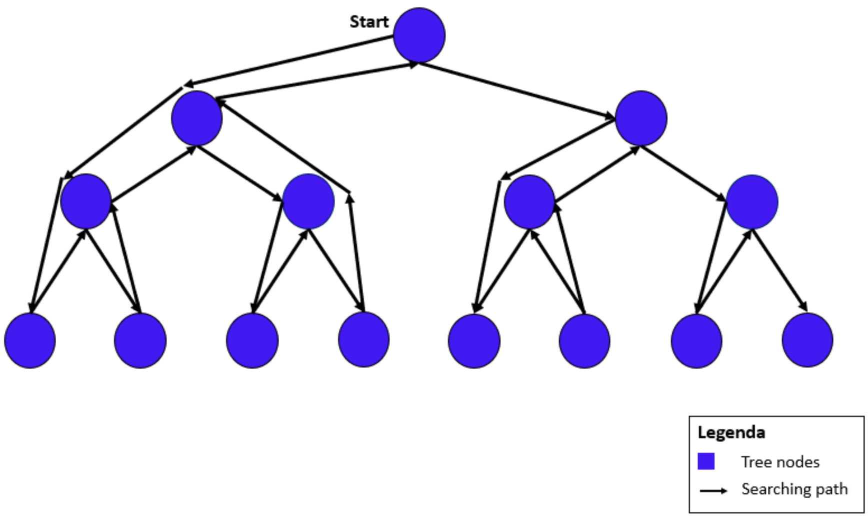


FIGURE 2.7: Depth first search algorithm

Branch and bound is a general technique particulary used to solve finite optimization problems. It starts, first of all, subdividing the original problem in sub-problems easier to solve. Branch and bound algorithms are called of implicit enumeration since they enumerate all the possible solutions and tries all of them, but some will be deleted proving their non-optimality [69]. Figure 2.8 describes the enumeration process of the branch and bound algorithm.

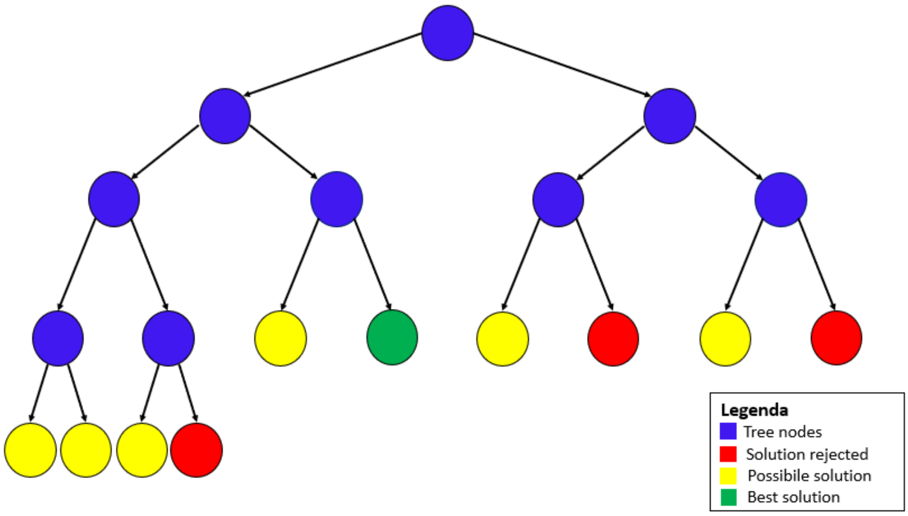


FIGURE 2.8: Branch and bound process

The CSP approach is a method largely studied in different fields of science. It is important in robotics; in fact, as already said, it is considered as a method of the Artificial Intelligence. Panesc et al. consider holonic coordination through a combination between the contract net protocol (CNP) and distributed constraint satisfaction problem (DisCSP) and how the designed method can be confirmed by appropriate models and analyzing tools [80]. Trabelsi et al. [111] applied a CSP method to support the preliminary design of a linear vehicle suspension system. This research, which was focused on the sizing of the system, proposes a comparison between conventional design methods and shows how the computation time related to CSP models is satisfactory. They developed a simple design principle for an untethered, entirely soft, swimming robot with the ability to achieve directional propulsion without batteries and on-board electronics. Mathematicians study the difficulties of these problems to help the resolution of them. In particular Butalov et al. focus themselves on weighted Boolean CSP and their complexity [17]. Carvalho et al. study the duality of non-uniform CSP by the use of non-linearity involving trees of bounded path-width [21]. Jonssona et al. present a general result for characterizing computationally hard fragments and, under certain side conditions, this result implies that polynomial-time solvable fragments are only to be found within two limited families of sets of relations [53]. Risk management also use CSP to find solutions for their problems. Gouriveau et al. extend with CSP the CBR interest: not only the case base will be a support to manage risks but also a support to manage decisions [46]. Leet et al. examines how data sharing has the potential to create risk for enterprises in retail, supply chain collaboration and proposes a new algorithm using CSP to remove sensitive knowledge from the released database based on the intersection lattice of frequent item sets [60].

CSP approach is largely used also in engineering in particular in the design phase and in the life management of the product. The design process of manufactured products has been taking into account technical, economic, and social aspects for about 15 years [39]. Generally, explicit and implicit rules are necessary to support a design phase [38]. In fact, the mechanical design is an interactive and synthesis process to define choices such as materials, dimensions, and shapes [38].

A CSP model implements all these knowledge representations as mathematical constraints. For this reason, a lot of research is focused on techniques and methods for design optimization. Fisher et al. proposed a product development process from the early phases to the embodiment design phases [39]. They enhanced the design collaboration and knowledge sharing to coordinate the design process in the first phases of the development of a new product. They also implemented a CSP tool into the design workflow to rapidly generate design solutions that could be also exchanged through the design process. However, they did not consider an optimization approach related to the CSP analysis.

A multidisciplinary design optimization advisor system [121] and the implementation of an ICAD generative model [92] was proposed in the design of aircraft applications. Lin [63] proposed a TRIZ-based approach to support design when the solutions of optimization methods do not meet the objectives of problems to solve. His approach is focused on the analysis of system contradictions which extracts data from the simulations generated from the optimization loop. Yang and Dong proposed a CSP approach to address the problems related to the configuration conflicts involving structural restrictions, configuration rules, and repair rules [127]. A framework for collaborative top-down assembly design using CSP is presented by Gao et al. [42] which allows a group of designers geographically dispersed to collaboratively conduct top-down assembly design of complex products with their design knowledge and experience in a computer supported cooperative work environment. Andrée et al. [6] introduce a design platform (DP) based on CSP that aims to support the development of customized products when traditional platform concepts do not suffice. Other studies focuses on managing design change order in a PLM platform using CSP (Ducellier et al. [31]). Other issues are investigated by Pitiot et al.[84] that also considers the problem of mix the ERP characteristics, that are also important in the engineering approach, with the characteristics of a simple configurator, they elaborate a method that preconfigure the products solving a problem that involves only non-negotiable requirements obtaining a solution space of reduced configurations and afterwards optimizing this reduced solution space. They focused on the optimization. These authors considered a constraint filtering

process in evolutionary algorithms, obtaining satisfactory results in terms of the time and convergence. Yvars [82] introduced a multi-objective deterministic CSP approach to solve a pareto bi-criterion optimization to reduce the size and the complexity of the design phase satisfying conflicting criteria. However, this approach does not consider the integration of the design workflow with other tools such as external configuration software or numerical FEM solvers. In the context of oil & gas applications, [27] describes an optimization workflow to reduce the weight of big steel structures. This approach considers the integration of FEM simulations into the optimization analysis. He proposed a research example focused on a bolt coupling [82]. His research used CSP methods to reduce the size and the complexity of the design phase. The use of FEM solvers increases the calculation time and requires the use of other kinds of optimization algorithms such as GAs (Genetic Algorithms) [27]. Raffaeli et al. [88] also studied CSP problems. Their focus was on mechanical systems such as Engineer-To-Order (ETO) products. These complex products require a more complex formalization of knowledge. He studied a CSP approach for a further implementation into a generic configuration system. Generally, design parameters in CSP problems must satisfy a collection of constraints related to standard, customer requirements, marketing requirements, etc.
Besides the constraint-based knowledge representation, graphical and logic-based knowledge representation have been developed. Both feature models and UML configuration models fall into graphical knowledge representation. Answer set programming (ASP), description logics (DL) and hybrid configuration (HC) are different approaches to perform a logic-based knowledge representation. In particular, ASP maps a logic-based configuration model into propositional logic theories, which are solvable by means of boolean satisfiability (SAT) solvers. Since ASP allows a component-oriented representation by using a first order logic-based knowledge representation, it is a valid alternative to CSP.
In the last 10s, the research on CSP has been focusing on the way to simplify the numerical problem. The motivation is related to the computational cost which affects the classical algorithms. In 2009, Vu et al. studied a new method to reduce the space of solutions. They used multiple enclosures techniques during the constraint propagation [116]. In 2013, Bistarelli et al. applied the concepts of substitutability and interchangeability in CSP with soft constraints [13]. In particular, a soft CSP is a solution problem which implements constraints that have a set of preference values. This technique is usually applied in solutions such as heuristics and adaptive. Anyway, the employment of soft constraints has improved the efficiency of branch and bound algorithms in CSP solutions.

Several tools and solvers have been introduced to solve CSP problems.The most used techniques are variants of backtracking, constraint propagation, and local search. The branching factor at each level of the tree is equal to the number of variables not yet assigned, multiplied by the number of values that each variable can take. Gecode [97] is a development platform which implements CSP algorithms. This open and free framework is a programmable toolkit for extensions based on C++. Gecode already contains many features but its strength is in the fact that it is programmable: it supports the implementation of new constraints, strategies and search methods [97]. Moreover, Gecode is efficient in time and memory consuming. It has been used to solve over 50000 test cases with good results. However, Gecode has some limitations: it does not support geometrical constraints, it does not support connections with database, it has also limitations with using constraints that involves strings and finally the connection with other software must be implemented with custom libraries. Münzer [66] proposed Gecode to implement part of a model to search design solutions derived from a set of specified techniques. His method consists in defining a metamodel of the problem and its constraints, finding the interconnections, assigning variables using Gecode and optimizing an objective function using simulating annealing. The approach for the resolution is based on ad-hoc algorithms that compare all found solutions at the same depth (with usually the same cost) to avoid running in infinite paths.

Other tools have been analyzed to solve CSP problems such as MiniZinc [60], FlatZinc, Choco, Cream and Jacop.

MiniZinc is a medium-level constraint modelling platform based on Zinc language [47]. It can be mapped onto existing solvers. Algorithms developed with MiniZinc are compiled in FlatZinc language which exploits the advantages of a library of pre-defined constraints.

FlatZinc is a low-level programming language which can directly interact with solver such as Gecode and others [110] to translate the CSP model into the language required by the defined solver.

Choco is another free open-source Java library dedicated to CSP problems [85]. The user can define its problem in a declarative way by stating the constraints that he wants to satisfy. Then, the related CSP model is solved by alternating constraint filtering algorithms with a search engine.

Cream is a Class Library for Constraint Programming; Jacop is a Java-based constraint solver.

These tools and others have all the same problem that they cannot be programmed; thus, plug-in and extensions are not allowed.

Nowadays, different commercial tools and mathematical algorithms are available to support the engineering design in fields such as mechanics, electronics, and civil. However, a lack still exists in the development of a flexible and agile design methods to support the optimization workflow [2]. Even if the design complexity can be managed using a KBE tool, CSP methods, and optimization algorithms, a lack of tools still exists in the development of a flexible and agile design method to support the optimization workflow [2]. The product configuration allows the past design solutions to be reused; however, the delivery of the ETO products requires an engineering analysis before closing the order with the customer. Therefore, the configuration tools can only support a predesign phase in the ETO context. Furthermore, the traditional commercial tools cannot support the designer from the early configuration phase to the product optimization phase including simulations and CAD automation.

2.9 Machine learning

Machine learning is a branch of artificial intelligence that collects methods developed in the last decades of the twentieth century in various scientific communities, such as: computational statistics, pattern recognition, artificial neural networks, adaptive filtering, dynamical systems theory, image processing, data mining, adaptive algorithms.... Machine learning uses statistical methods to improve the performance of an algorithm in identifying patterns in data. In the field of computer science, machine learning is a variant of traditional programming in which a machine prepares itself the ability to learn something from data, without explicit instructions [4].

Arthur Samuel, who coined the term in 1959 [94], identifies two distinct approaches. The first method, referred to a neural network, develops general-purpose machine learning machines whose behavior is learned following a reward-and-punishment-based learning routine. The second, more specific, method is to reproduce the equivalent of a highly organized network designed to learn only specific activities. The second procedure, which requires supervision and requires reprogramming for each new application, is much more computationally efficient.

Machine learning is closely related to pattern recognition and computational theory of learning [12] and explores the study and construction of algorithms that can learn from a set of data and make predictions about it, [55] by building inductively a model based on samples. Machine learning is used in those fields of computer science where designing and programming explicit algorithms is impractical.

Machine learning is linked to, and often overlaps with, computational statistics, which are concerned with making predictions using computers. Machine learning is also strongly linked to mathematical optimization, which provides methods, theories and application domains to this field. For commercial uses, machine learning is known as predictive analytics.

Computational analysis of machine learning algorithms and their performance is a branch of theoretical computer science called learning theory.

The main objective of machine learning is that a machine is able to generalize from its own experience: it must be able to carry out inductive reasoning. In this context, generalization refers to the ability of a machine to accurately complete new examples or tasks that it has never observed after having experienced a set of learning data[11]. The training examples are assumed to come from some probability distribution, generally unknown, that is considered representative of the occurrence space of the phenomenon to be learned; the machine has the task of building a general probabilistic model of the space of occurrences in order to be able to produce sufficiently accurate predictions when subjected to new cases.

Machine learning tasks are typically classified into three broad categories, depending on the nature of the "signal" used for learning or the "feedback" available to the learning system [74]. These categories, also called paradigms, are:

- supervised learning, in which examples are given to the model in the form of possible inputs and their respective desired outputs and the goal is to extract a general rule that associates the inputs with the correct output;
- unsupervised learning, in which the model aims to find a structure in the inputs provided, without labeling the inputs in any way;
- reinforcement learning, in which the model interacts with a dynamic environment in which it tries to reach a goal, having a teacher who only tells him if he has achieved the goal.

Another categorization of machine learning tasks arises when considering the desired output of the machine learning system: classification, regression and clustering.

In the classification, the outputs are divided into two or more classes and the learning system must produce a model that assigns the inputs not yet seen to one or more of these. This is usually addressed in a supervised manner. Spam filtering is an example of classification, where the inputs are emails and the classes are "spam" and "not spam". In regression, which is also a supervised problem, the output and model used are continuous. An example of regression is the determination of the cost of an engineering component knowing only on some features. In clustering, a set of inputs is divided into groups. It differ from classification in the fact that groups are not known before, this is a typical unsupervised task.

Even if machine learning techniques seems to be very promising they have some issues linked to them.

Since the training examples are finite sets of data and there is no way of knowing the future evolution of a model, learning theory offers no guarantee on the performance of the algorithms. On the other hand, it is quite common for such performances to be constrained by probabilistic limits. In addition to performance limitations, learning theorists study the time complexity and feasibility of learning itself. A computation is considered feasible if it can be performed in polynomial time.

For generalization, the complexity of the inductive hypothesis must equal the complexity of the function underlying the data. If the hypothesis is less complex than the function, then the model manifests underfitting. If the complexity of the model is increased in response, the learning error decreases. On the contrary, if the hypothesis is too complex, then the model shows overfitting and the generalization will be poorer [3].

The last problem is related to the number of the training examples. This is an unknown that depends principally on the inputs features of the model but it should be consistent to produce a relevant output, in the order of hundreds or better thousands. Such a large amount of training data is hardly to found and this affects the reliability of the prediction model.

2.10 Optimization analysis

Optimization analysis are based on searching the parameters configuration which optimizes the target objectives [63]. In this approach, constraints are defined as limits, which are represented through Boolean checks (if <condition> then <instruction-1> else <instruction-2>), which are generally evaluated at the end of an optimization workflow. The condition to be verified is a rule or a formula which regards the know-how of the product knowledge. However, if an optimization analysis requires the employment of heavy time-consuming simulations, the verification of constraints is applied after the iteration of many simulations. Therefore, the optimization workflow can also generate many not-verified solutions. This case is not efficient because it employs computing resources to calculate different not-valid configurations. Additionally, an optimization analysis requires the definition of a range of values for each parameter to be analysed. This is a manual phase and it requires the designer's know-how to configure each parameter range. Parametric template-models can help to develop a design automation workflow from configuration to optimization; however, the definition of the variation range is often an engineer's task and therefore a manual activity. Optimization tools can work with a CSP solver and allow finding out a valid solution that minimizes or maximizes an objective function. [71]

ETO products are more difficult to be optimized because they often are complex systems. Generally, the optimization in ETO solutions is parametric and parameters are related to normative and geometry. A modular approach can be extended to a multi-step optimization to reduce computing time and complexity [26].

2.11 Open issues

In conclusion nowadays configuration is a research area that still alive and has many research arguments:

- Configuration tool for ETO products is limited.
- Configuration should be open to different scenarios.
- Human computation should be allowed.
- Diagnosis of infeasible problems is limited.
- Idea of costs still remote.

The background research highlights a lack of methods and tools that efficiently support companies during the design phase especially in the configuration phase.

Chapter 3

The configuration approach

An overall configuration approach is proposed during this thesis to support the reuse of company knowledge to design and to estimate the prices of new ETO products.

The design of ETO structures requires the development of several engineering subsystems associated with the structural layout, machinery, accessories, etc. During this context, a constraint-based optimization aims to fill the gap between the collaborative design, knowledge domain, simulations and optimization during the first design phase of ETO products.

The design study is targeted on two actions: Request-For-Proposal (RFP) and Proposal Submission. The approach aims at supporting the designers in a rapid and reliable definition of an embodiment project in order to participate to a Proposal Submission.

A company operating within the business of ETO solutions is usually involved in many quotation processes. Even if objectives and targets are defined with the specific customer, an optimization activity is fundamental to achieve a high product quality at a reduced cost.

The aim of my research is the definition of methods and tools to provide the design phases during two of its fundamental aspects: optimization and configuration.

Design optimization is an engineering design methodology using a mathematical formulation of a design problem to support selection of the optimal design among many alternatives.

Configuration design is a design where a fixed set of predefined components that can be connected in predefined ways is given and an assembly of components selected from this fixed set is sought that satisfies a set of requirements and obeys a set of constraints.

Even if these two macro-themes may seem to be disjoint, for my research work they have been both fundamental and in particular optimization design has been preparatory to configuration design. In fact the study and the application of the first one culminates in the development of a framework that concentrates principally on a constraints solver engine and stressed it to test its capacity to produce reliable result quickly.

This solver improved have been reused then in a more general context, the design configuration, in order to support the designer in more general cases. Around this solver, that is the principal engine of the configuration phase, have been developed a software able to support better both the technical and the commercial scenarios of a company.

In both cases, optimization and configuration, the aim regards the reduction of time in mechanical design using feasible and automatic design tools. These tools implement knowledge base formalized in rules, formulas, and constraints. Synthetically, it consists of the subsequent steps:

- Knowledge formalization and representation.
- Support towards a design solution.
- Cost estimation.

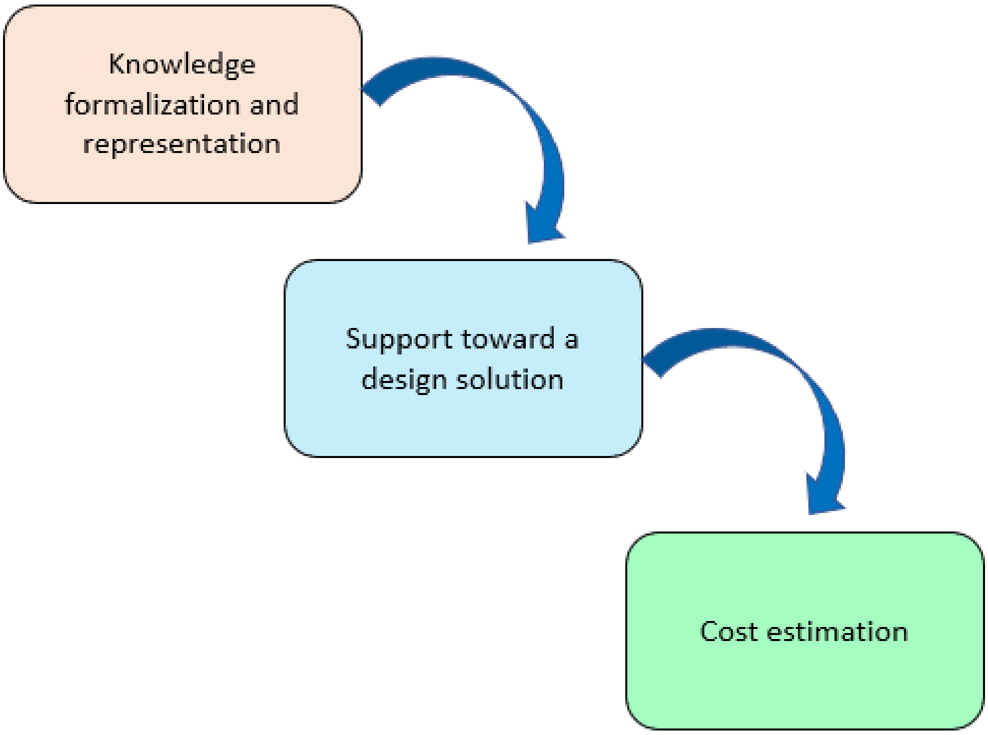


FIGURE 3.1: Scheme of the approach

The first step is a preparatory phase where the company knowledge is formalized and represented in order to be reused. Past project data and product knowledge are embedded in a database of product variants and the product architecture is represented by a meta-model in terms of blocks and dependencies. By combining in numerous ways blocks and dependencies, it is possible to get different product variants, aiming at satisfying product requirements. The second part of the method supports the user to find a technical solution that meets the product requirements. This design workflow presents a collection of stand-alone tools that may be integrated inside the life cycle employing a Knowledge Base which consists of a repository of rules, formulas, documents, past projects, etc. A DSM is proposed for representing and managing the dependencies between each attribute. Therefore, the resulting problem is split into several sub-problems. Each sub-problem has a reduced size and concerns the assignment of a limited number of variables. A parameter-based cost estimation follows the variables assignment. Finally, the cost evaluation relies on cost models which are retrieved by past project data.

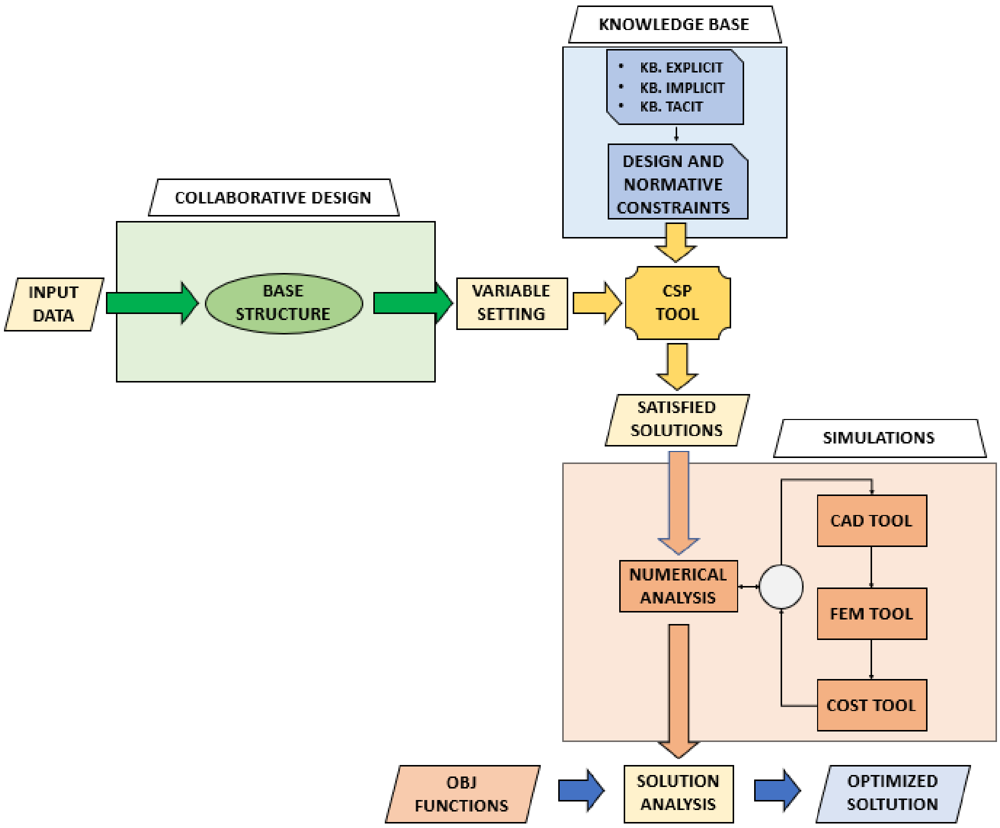


FIGURE 3.2: Proposed design framework

Analyzing the design platform, as shown in 3.2, the Collaborative Design domain concerns the study and coordination of each design sub-problem within the design team. The collaborative approach also aims at searching the common and specific parameters to be involved into the constraint-based optimization of each configuration.

The Knowledge Base domain concerns the collection and sharing of rules, formulas, and constraints related to the engineering design of the ETO product to be analyzed. The constraints related to the engineering design are mainly focused on the normative checks and conditions. The other constraints are related to the validation rules based on the expert knowledge, and these constraints can refer to the formalization of the implicit and tacit knowledge. An example can be the rules of validation such as the check of the weight balance for the steel structure. The Design Constraints enhance the design synthesis, avoiding infeasible solutions. The level of the Design Constraints concerns the validation rules focused mainly on the domain of the technical expertise. Therefore, these constraints refer to the formalization of the explicit and implicit knowledge. This aspect ensures that the generated set of solutions is consistent using the knowledge and rules formalized in each constraint. Here, the CSP tool can be seen as a bridge between the three domains: Collaborative Design, Knowledge Base, and Simulations 3.2.

The Simulations domain corresponds to the employment of a virtual prototyping approach to evaluate the structural feasibility and behavior. The definition of the boundary conditions and the evaluation of the results are shared along the entire collaborative design team. In particular, this phase corresponds to the detailed design. The Simulations level also involves the definition of a set of Normative Constraints, which are focused on the normative checks and loading conditions as check conditions to be applied into the simulation loop.

As software toolkit used to develop the framework, has been used Visual Studio 2015 community edition. The programming language is principally Visual Basic, but for some classes has been adopted also C#, both based on the Framework.NET 4.0. Then some Application Programming Interface (API) of existing libraries have been imported to ease the communication with external tools, for example Microsoft Excel or Gecode.

3.1 Knowledge formalization and representation

A Knowledge Base is proposed to collect each rule, formula, and data related to the product design. This repository of knowledge can interact with commercial softwares that manage databases using the Application Programming Interface (API) tools. The Knowledge Base is connected to each stage of the proposed design, to integrate each input and output into a common design platform.

The functional analysis is the first phase for enhancing the formalization and representation of a Knowledge Base related to a product or a system. This approach is compliant with the principles of Design-To-Cost since it manages the product value from performances to manufacturing costs.

The first point of the knowledge formalization is preparatory but also the most complicate, the designer or the team of expertise must formalize the problem. They must focalize all the technical and geometrical specifics that the final product must have. This is the most difficult part because often designers base their design on previous experiences. The experience of working alone has an inconvenient: sometimes something that works is not the best solution for the problem. Maybe the product can be optimized in terms of costs, green sustainability or efficiency. To overcame this issue a solution can be working in a team of experts: connecting different contribution can help to bring out company knowledge to be optimized in a more easy and quick way. The first step consists of an effort from each expert to translate constraints that are only in his mind into mathematical rules that a computer can understand and elaborate. He has also to divide problems into variables, their range of variability and constraints. On the basis of what we explicate till the usage of the software seems to be inconvenient due to the big amount of manual work. On the contrary this part is fundamental since it permits to save the model and to reuse it only changing some initial parameter. The other advantage in using this approach is that, once formalized, the model allows to transmit a company knowledge that can remains equal through the next generations and can only be improved refining the model.

Collaboration of experts should not involve only mechanical experts, but also informatics ones. This is also a key point in the collection of Knowledge since each part is aware of strengths and limits of its field. Mechanical experts master the real problem and with a relative little effort can be able to extract the abstract problem but they often are not aware about problems like numerical inconsistencies; on the other hand Informatics master programming rules and control memory problems but they have difficulties to abstract the real problem.

Company knowledge can be formalized in different software tools such as CAD, CAE, PLM, and ERP systems. However, this level of knowledge it is mostly related to its explicit side. Tacit and implicit knowledge, which is the most difficult to be elicited and represented, can be represented in rules and algorithm which are usually implemented in customized Knowledge-Bases Systems.

The proposed framework, described in the following section, interacts with different software tools following these functionalities: acquire and manage the product requirements; support the user towards the product configuration; support the user in defining a simplified 3D layout and define cost models for an early cost estimation.The proposed framework is to be intended as a part of an integrated environment, which links product knowledge coming from design, manufacturing, marketing, maintenance gathered from the company departments: technical, production, service, commercial and purchasing departments. Managed knowledge extends to the product life cycle, enhancing standardized interfaces to acquire information. Moreover, the proposed framework needs many information from other tools. For instance, PLM systems provide product structure and product variants classification. ERP systems give information about materials and resources used in the past projects, that is necessary to estimate the new product cost. Indeed, vendor catalogs and standard parts databases contain technical and economic information that are useful to build cost models.

3.2 Support towards a design solution

The aim of this section is to describe how the proposed framework, that is the application of the method in Figure 3.2, can support the user for searching one or more optimal solutions moving from product requirements, tentative product architectures, parts dependencies. Optimization can be based on one or more parameters, e.g. the product cost.

The framework shares geometrical data with CAD tools, to provide a preliminary 3D layout. Geometrical attributes of the product are used to generate parametric CAD models having a simplified geometry. The simplified 3D layout, which is the output of the conceptual design phase, becomes the starting point of the embodiment design phase. CAE tools receive information from the framework such as product attributes or a preliminary 3D layout while they give results coming from preliminary design optimization and performance verification analysis.

Finally, using API libraries or other communication means, the integration can be extended to a large variety of customized company tools, such as performance models or simulation software, including existing spreadsheets for component dimensioning and evaluating possible scenarios.

The knowledge formalized in the previous step is now elaborated and solved using the following tools: the Design Structured Matrix (DSM) and the Constraint Satisfaction Problems (CSP).

3.2.1 Constraint typologies

Three constraints levels can be highlighted: Configuration Constraints, Design Constraints, and Performance Constraints. This classification aims to investigate the different types of constraints inside a typical mechanical design, which is based on configuration and optimization phases. In particular, Configuration Constraints are those validation rules and formulas which are mainly related to the implicit knowledge of the engineer. An example could be the compatibility analysis between two options; thus, this level introduces constraints to evaluate the product structure of a configuration. The level called Performance Constraints is focused on normative checks and conditions such as structural limits to be applied; therefore, this level mainly concerns the explicit knowledge. On the other hand, the

Design Constraints level highlights validation rules mainly focused on the domain of the technical know-how. These constraints can refer to the formalization of explicit and implicit knowledge. Examples can be validation rules to check the weight balance of assemblies into a steel structure. Another example can be rules to evaluate and limit the number of different types of components, such as the types of beams into an assembly.

Summarizing, if Performance and Configuration Constraints are mainly related to normative and setup rules, Design Constraints are focused on practical satisfaction rules which depends on the engineer's know-how. In this context, Configuration Constraints are usually implemented in configuration tools such as filtering and validation rules, and Performance Constraints are defined as checking conditions into optimization loops after simulation analysis. The collection of Design Constraints is often defined into previous two levels; however, the proposed approach highlights the necessity to use them in an intermediate level.

3.2.2 Steps to solve the constraints

Once the configuration rules are defined, these rules are used to create the Design Structured Matrix (DSM) associated to the problem. The DSM is a matrix which contains the parameters, e.g. the variables of the rules, and traces the relationship of dependencies between the parameters. A parameter is dependent from other parameters if there is a rule, or a set of rules, in which the dependent parameter is a combination result of the other parameters.

The Figure 3.3 shows the structure of the class used in the software to represent the parameter.

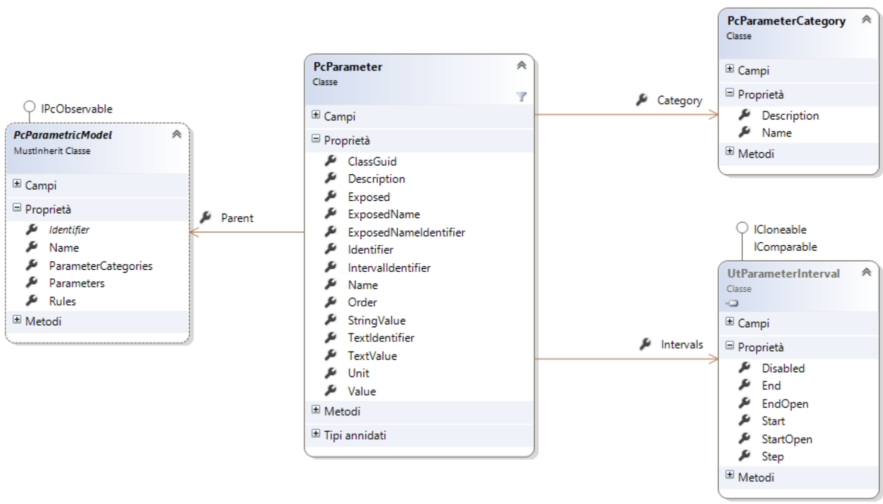


FIGURE 3.3: The structure of the class for the parameter

A parameter is part of a parametric model composed by parameters and rules. The parameter is characterized by its value that may be numeric or a text, it has a category and it is defined for certain intervals.

To solve the CSP constraints five steps have been identified:

- Instantiation of the DSM.
- Clustering of the DSM.
- Partitioning of each cluster.
- Clusters resolution.
- Checking for constraints satisfaction and global DSM.

The instantiation of the parameter-based DSM (step 1) is automatic because it is defined in the constraints. The introduced dependencies generally constitute an intricate and complex network with an elevate degree of coupling between elements. Such complexity can be identified as the main difficulty a designer is facing when trying to figure out a preliminary solution in the early phases.

Clustering the DSM allows to divide the problem into simpler sub-problems of reduced size, which are expected to be much more manageable. This approach is close to the designer perspective, which is used to dealing with problems of limited size. Relationships between the sub-systems as well as input and output data flow are highlighted and monitored. Clusterization is based on a simulated annealing optimization procedure derived from the one proposed by Thebeau [108]. The advantage of such approach lies on the capability of controlling the number, size and composition of the clusters based on custom algorithm parameters and functions.

Once the problem has been divided into sub-problems, each of these can be solved separately. However, clusters maintain couplings between them and should be solved respecting the order in which they appear in the clustered DSM. In order to solve each sub-problem, a partitioning algorithm is applied to each cluster (step 3). The partitioned DSM shows all the product parameters sequenced according to their dependencies, thus it is possible to know when a parameter must be defined in order to proceed with the design process. The partitioning algorithm groups around the diagonal those attributes that need to be defined in the same design step, as highly coupled [81]. Thus, the obtained parameters sequence allows minimizing the iterations during the phase of the determination of the parameters.

In general, the partitioned DSM shows several blocks of coupled parameters, i.e. the parameters that are affected by mutual dependencies. In order to determine these parameters and proceed with the design process, a priori choices are needed. A priori choices are properly characteristics of the design process, e.g. the material selection of a part. Usually assumptions must be verified after an iteration. For example, a constraint on the weight of a part can be verified only after several choices have been made, i.e. the material selection. In order to assess what parameters are the best candidates for a priori choices, an algorithm has been proposed. The algorithm aims to minimize the number of parameters to make assumptions and suggests the parameters to the user. In addition, the method serves to identify the steps in which human choices are needed.

The algorithm tries to remove dependencies from each one of the coupled parameters and, after new partitioning, verifies if triangularization condition is satisfied. Triangularization condition of the DSM means the possibility of determining all the parameters of the block without other parameter assumptions. If only one parameter is not enough to reach the condition, algorithm searches for pairs of parameters, triplets, and so on.

Cluster resolution, i.e. variable assignment, is done using CSP solver techniques. Design process can therefore be considered as a CSP under a set of requirements, assumptions and design limits [66]. If dependencies among the coupled parameters include geometric relations, the problem cannot be solved by a CSP solver and needs the human intervention. In this case, the user should define a plan of experiments to explore the solution space and return the parameter values that the system cannot calculate. Human intervention can be also requested in solving a cluster that is estimated to be too time consuming. The framework in these cases interacts with the user that, if he is enough expert, can solve the cluster at glance.

Now will be shown a pseudo-code of the resolution algorithm based on the backtracking technique:

Preliminary steps:

- Prepare the mapping variable-list of rules that will be named ***variableRules***.
- Prepare an array, named ***valuesIndices***, as long as the number of variables, initialized at -1, that indicates where the algorithm is arrived to process the rules.
- Prepare an array, ***valuesCount***, as long as the number of variables, initialized at 0, that take into account the count of the possible values of a variable.

Pseudo-code

```
DO (searching solution loop)
   'initialization of the CSP solution as long as the number of variables initialized at -1
      FOR v=0 to variablesCount
            Consider the variable at position v and retrieve its admissible values
            Consider the rules, not already solved, that are related to the selected variable

            'admissible variable value loop
            valueIndex = -1
            FOR i = valuesIndices(v) TO valuesCount(v) -1
                  valuesIndices(v) += 1
                  IF (all the rules in variableRules(variables(v)) are satisfied) THEN
                        valueIndex = i
                        EXIT FOR
                  END IF
            NEXT

            IF valueIndex<0 THEN
                  'the backtracking is applied, e.g., the algorithm come back of one variable
                  valuesIndices(v) = -1
                  v -= 2 (this is the decrement because the NEXT increments of 1 unit)
            ELSE
                  'the partial solution is saved
                  solution(v) = valueIndex
            END IF
      NEXT (variables)
WHILE (solutions research)
```

FIGURE 3.4: Pseudo-code of the backtracking algorithm

The searching of solutions ends when a valid solution is found, i.e. variable assignment complies with all dependencies within the block. However, it is possible to proceed with the exploration of the solution space in order to find other valid solutions, with the aim of minimizing or maximizing an objective function, e.g. the product cost. In this thesis, cost optimization has been applied in order to make an offer more competitive, but other parameters can be used for optimization, e.g. the product weight, performance or environmental impact. Although variable assignment is mostly done manually, the framework monitors the compatibility of each assignment with the set of dependencies. When all the parameters have been assigned and verified, a solution for the entire problem has been found.

In the tool implemented by the authors there is also the possibility to add a pre-optimization function, that do not perform a total optimization, but it orders the results obtained for some parameter that in general is the cost of the model or the time to finish all the operations on the model.

3.2.3 CAD automation level

The next step described in the proposed Knowledge Base is the CAD Automation level. Designers can use tools to generate parametric CAD models from the define Product Structure. Each geometrical parameter can be edited considering the set of values defined by the previous analysis. The levels of detail are related to the set of CAD templates used for CAD automation. However, a simplified level of detail is suitable to achieve fast simulations.

3.2.4 Optimization phase

Once the problem is solved the results can be exported in a file. For the Design Configuration the works of the framework can stop here, for the Design Optimization the works must continue in the following way. The report file represents the cases that are given to the FEM analysis to be optimized. In this phase the results obtained by the previous analysis are settled into a FEM analysis tool, selected from a list of available commercial tools, that will optimize the problems in terms of performance. Genetic algorithms or similar can be used into the design workflow to optimize the geometrical parameters of the analyzed ETO systems. At the end of this analysis is obtained the model optimized in terms of cost and performance that respect all the customer requirements.

The approach considers a multi-objective optimization including the cost reduction as an objective together with the structural behavior. The tool for the evaluation of the product cost should be customized on the specific product. Therefore, this tool can be parametric and developed by the design team of specific ETO products, or a commercial software solution with customized functions. The difference between these two solutions depends on the level of accuracy to be achieved.

The proposed Optimization level also presents an integration with the Knowledge Base. The interaction concerns design constraints to be applied to the optimization workflow and the formulas used for cost analysis and analytical calculations. Finally, decision-making tools are necessary to complete such design workflow because the result of the multi-optimization study is not a final configuration of parameters but a Pareto front which represents a set of optimized solutions. Every point of the Pareto front are optimum solutions, but the decision-making process has to select only one optimum case. A strategy is to change second objectives in design constraints, in order to obtain a one-level optimization and reduce the complexity of such analysis. As a second strategy, threshold values can be applied to each objective in order to reduce the points of the Pareto front. The connection between decision-making tools and Knowledge Base is related to the definition of threshold values and constraints to be considered. The decision-making process can be also supported by the analysis of past-projects stored into the database of the design platform.

3.3 Cost estimation

The cost estimation method is based on past projects data to evaluate the cost of new ETO solutions. Since the costs are attributes of the product, depending on blocks attributes and relationships, they are stored for a possible future reuse. For these reasons, the building of a cost database must precede the cost estimation activities. The database building consists in the acquisition and classification of past data from previous projects.

For cost prediction some machine learning techniques has been taken into account. Some test has been performed using techniques of supervised learning such as neural networks, but the database created was too poor of training examples and consequently the model for the cost estimation was not very reliable: the error of the prediction was about of the 20%.

For this reason in this work has been considered a deterministic model to perform an early cost estimation. A parametric method has been applied for a rapid evaluation along the configuration process. In fact, since a block is described by a set of attributes, the cost of each block can be expressed as a function of its attributes (such as weight, length, area, etc.. . .). These "blocks" can be produced within the company or purchased by a supplier. In the former, cost structure is known, while in the latter, only the total cost is known. For this reason, cost estimation follows two main paths distinguishing purchased materials from produced parts. The proposed cost estimation method for purchased parts is based on catalogs and on regression models, which can be derived by fitting the costs of past purchased codes using one or more relevant parameters (e.g. the power of an electric motor and the output torque for a gearbox). Cost estimation for produced parts are deducted by more complex cost models based on cost structure, allowing a better management of the company resources involved in the manufacturing and life cycle processes[96].

Chapter 4

Application in the Oil & Gas field

During the first two years, my research focused on the study of the background basis about configuration and the development of ETO products and propose a prototypical software application connected to Minizinc based on Gecode toolkit solver. In this approach, while Gecode runs in the background, an interface shows and manages the CSP problem.

This framework concentrates on the aspects of design optimization. Design optimization is an engineering design methodology using a mathematical formulation of a design problem to support selection of the optimal design among many alternatives.

The case study, that has been presented to validate the method, has been focused on the optimization of a 700 ton oil & gas structure.

The results of this chapter have been described in one of my articles [28]. The original article have been enriched with the addition of some parameters to analyze.

The results show that this approach reduces of several days, about two days against about seven days, the optimization design workflow since the analysis of the CSP problem decreases the cases that are given to the FEM analysis in order to optimize a less amount of possible cases infeasible because of some condition.

This experimentation did not involve any company expert as an already known case study has been revisited with this new methodology.

4.1 Development approach

The collaborative design platform in Chapter 3 (Figure 3.2) is proposed to support the design of ETO products by integrating a constraint-based optimization approach. The approach is defined for the design of generic ETO products with the problem of the structural analysis. In particular, the methodology is applied in the context of large steel constructions. Focusing on modular steel structures, the design of a structure can include the study of a set of structures related to different conditions. In the context of the oil and gas constructions, if a steel construction is built and then shipped as a plug and play unit, the collaborative approach aims at defining the optimal base structure to be used in three configurations related to the operation condition, the sea transport and the land transport. Furthermore, these three structures will have a common module related to the Base Structure, which consists of a primary structure (main frame or supporting structure) and a secondary structure.

The CSP model implements a set of variables, values, and constraints. The variables are defined during the Collaborative Design level, and each variable can take a range of values. The constraints are described using mathematical functions. The definition of each constraint is related to the formalized knowledge base. In this context, the CSP tool implements the rules to solve the Design Constraints before launching any simulation loop. This approach can reduce the number of solutions to be evaluated in a further simulation phase, which involves a time consuming analysis. Figure 4.1 describes the optimization workflow, which includes the CSP tool and Simulations. Considering the design of steel constructions, the simulation activities correspond to FEM simulations and cost estimation. The CSP tool is employed to reduce the space of solutions. Moreover the Simulations node performs a deeper investigation of each resulting solution,looking for the optimum solution in terms of the cost, weight, and structural behavior. The Simulations level also involves the use of a parametric CAD model to allow a FEM model and a cost analysis to be solved.

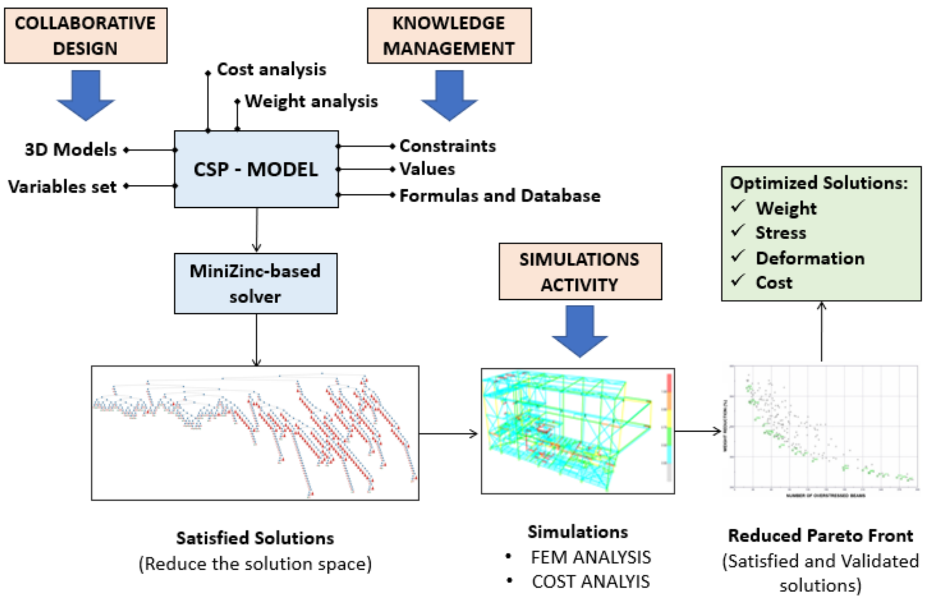


FIGURE 4.1: The optimization workflow, that integrates the CSP tool and the FEM solver.

The proposed methodology, explained in Chapter 3, was implemented into a platform tool to support the multi objective optimization of a steel structure for oil and gas applications. To solve the constraint-based problem, the CSP tool has been implemented in Visual Basic.NET using MiniZinc, which is a Gecode platform toolkit that aims to simplify the formalization of the problem avoiding issues like programming languages. The resulting tool consists of a CSP Solver and GUI Interface. The CSP Solver uses the Gecode code to find the problem solution, and it works in the background. Furthermore, the GUI Interface shows the visual commands for the management and definition of the parameters, constraints, and values. This framework, that use a knowledge domain already formalized and an existing library that do not request an apposite implementation, as been implemented in about two months.

Figure 4.2 describes the software architecture to solve the CSP problems.

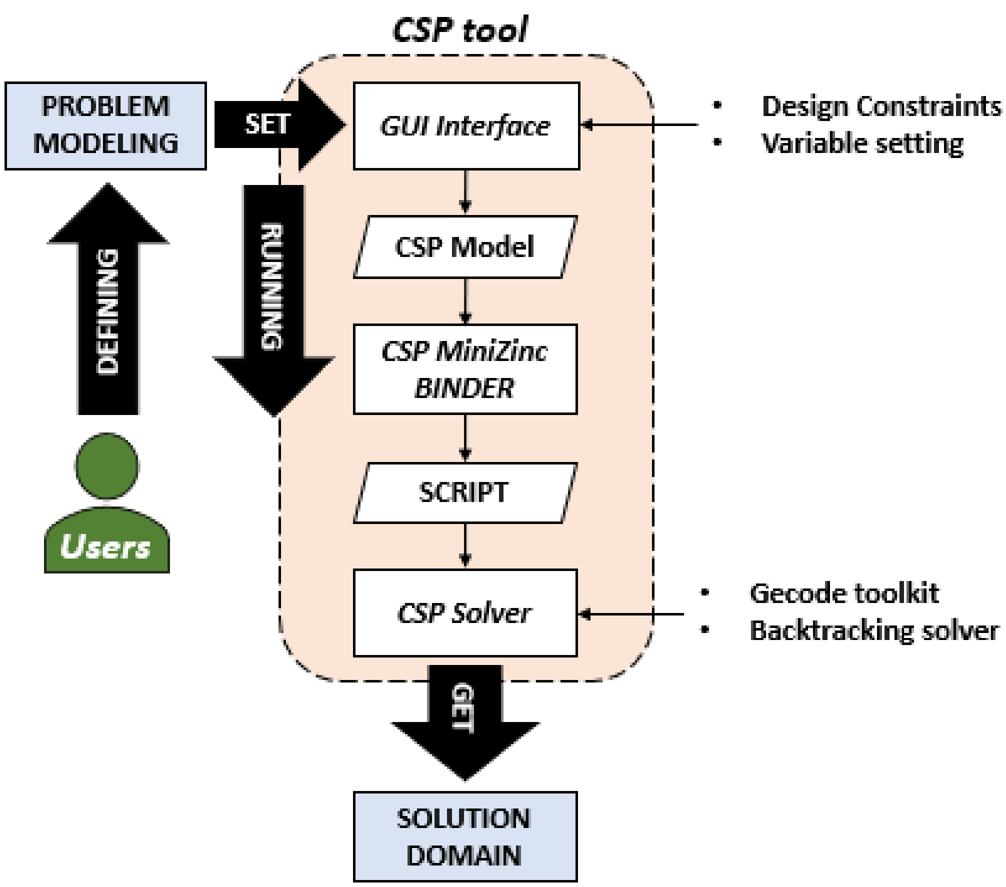


FIGURE 4.2: Software architecture of the CSP tool.

This tool implements a knowledge-base which is formalized in a CSP problem, so in rules, formulas, and constraints realted to the engineering design. The first step is the definition of the CSP problem. This task pertains to the difficult formalization of the technical and practical knowledge in the constraints, which means translating an aspect that is often only in the mind of the designer into a computational representation involving variables and functions. In this context, as already said, while the explicit knowledge is easy to translate into mathematical constraints, the implicit and tacit knowledge are difficult to be elicited and formalized. All the technical and geometrical specifications must be rewritten in a universal language that anyone can read. Therefore, the approach considers the implementation of a formalism to provide the input definition. The user can add constrains compliant to the defined formalism. The formalization of the technical knowledge, which is required to define constraints in the CSP definition, as already underlined, is a difficult phase since design routines are often based on experience. They are not related to standard procedures.

Then a simplified Graphical User Interface (GUI) has been defined for the management of the CSP problem and design objectives (second step). Figure 4.3 shows the proposed GUI, which consists of four levels: Variables, Constraints, Solver Method, and Results.

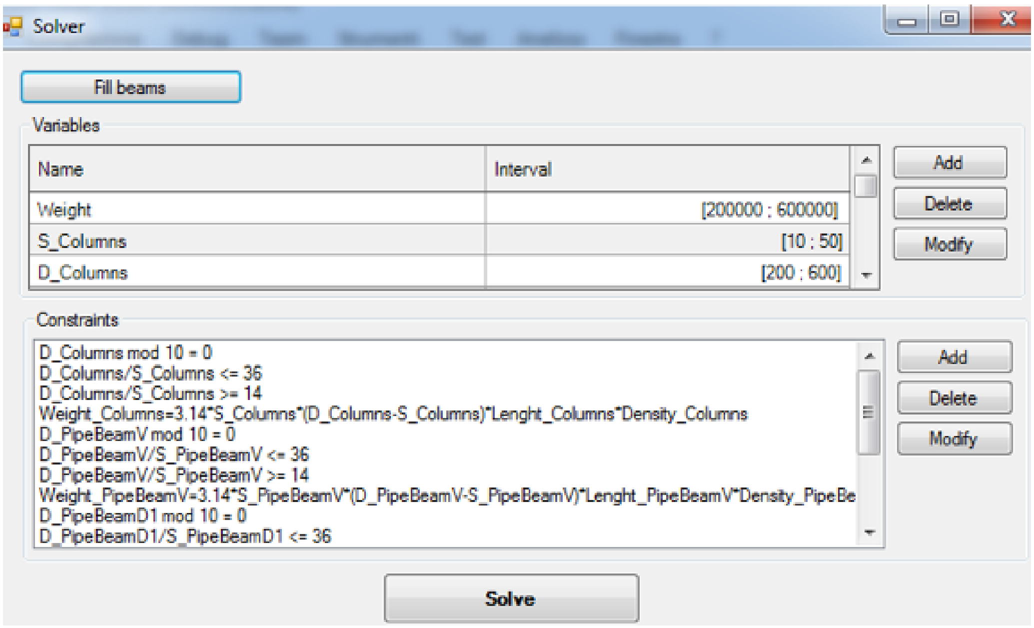


FIGURE 4.3: The GUI Interface.

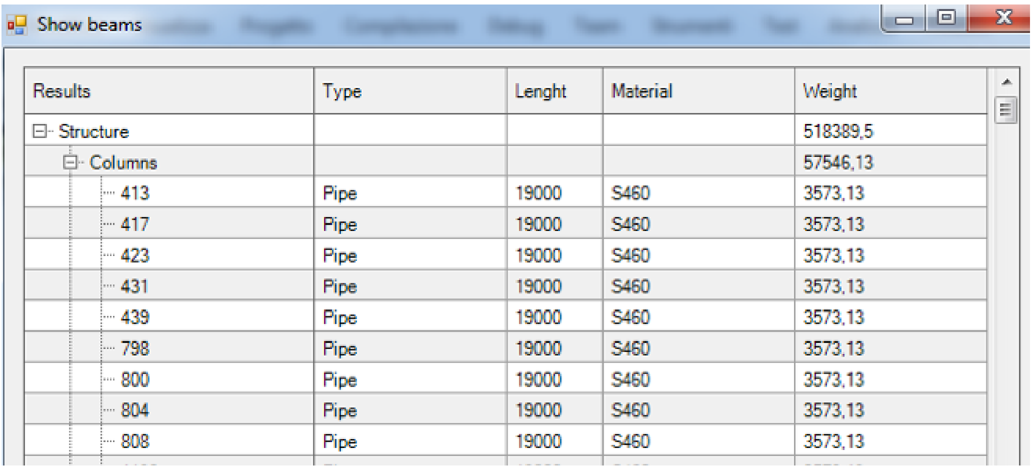
The first level concerns the settings and management of variables using a table. The Constraints level regards the definition of the mathematical rules to be verified in the CSP calculation. In particular, each constraint is edited using a mathematical representation of formulas and rules. This definition is an object-oriented (O-O) representation of the product structure. The generic classes are defined in Visual Basic.NET to support the O-O representation of a product structure. In particular, a generic set of must inherit classes is implemented. These classes provide the parent-child relationship between each component of the product structure. This O-O representation includes the definitions of the properties, methods, variables, and constraints. In the third level the solver method to be applied is defined and the forth level shows the results. After pushing the Solve button (Solver Method level in Figure 4.3), Gecode is invoked. The CSP definition is translated in Gecode language for the computing. In fact, the proposed application uses Gecode algorithms to solve CSP problems. Gecode elaborates the instructions to be solved, and then returns results.

In this Chapter, the set of classes to describe the product structure has been termed as the Beam Module. This project inherits the generic classes defined in the CSP model to represent the specific design problem related to the steel structures. The classes related to the Beam Module classify the definitions of the standard beams, pipes, and built-up shapes. These classes also include the variables related to the shape geometry of each beam. The procedures to calculate the cost and weight of such a steel construction were implemented in each element. The CSP MiniZinc Binder is a tool that was developed to translate the input data into a MiniZinc structure before running the CSP solution using Gecode. While a user can add/edit constraints and variables using the GUI Interface without using any specific programming language, the CSP MiniZinc Binder implements the methods to write constraints as a MiniZinc script for solving the CSP problem. Figure 4.4 shows an example of a MiniZinc code for the description of certain design constraints. When the user pushes the button "Solve", the CSP Solver invokes Gecode using the MiniZinc script.

```
constraint D_PipeBeamD2/S_PipeBeamD2 <= 36;
constraint D_PipeBeamD2/S_PipeBeamD2 >= 14;
constraint 3.14*S_Columns*(D_Columns-S_Columns)*Lenght_Columns*Density_Columns+3.14*S_PipeBeamV*(D_PipeBeamV-
S_PipeBeamV)*Lenght_PipeBeamV*Density_PipeBeamV
+Area_BeamD0MBT500*Lenght_BeamD0MBT500*Density_BeamD0MBT500+Area_BeamD0MBL400*Lenght_BeamD0MBL400*Density_BeamD0MBL400+Area_BeamD1MBT400*Lenght_B
eamD1MBT400*Density_BeamD1MBT400+3.14*S_PipeBeamD1*(D_PipeBeamD1-
S_PipeBeamD1)*Lenght_PipeBeamD1*Density_PipeBeamD1+Area_BeamD0SB300*Lenght_BeamD0SB300*Density_BeamD0SB300+Area_CrossBeam*Lenght_CrossBeam*Densit
y_CrossBeam+Area_CrossBeamDeck*Lenght_CrossBeamDeck*Density_CrossBeamDeck
+Area_BeamD2MBL500*Lenght_BeamD2MBL500*Density_BeamD2MBL500+Area_BeamD2MBT500*Lenght_BeamD2MBT500*Density_BeamD2MBT500+Area_BeamI*Lenght_BeamI*De
nsity_BeamI+Area_Supports*Lenght_Supports*Density_Supports+3.14*S_PipeBeamD2*(D_PipeBeamD2-S_PipeBeamD2)*Lenght_PipeBeamD2*Density_PipeBeamD2<=
600000;
```

FIGURE 4.4: MiniZinc language: example of constraints.

The test case proposes the modeling and optimization of a steel structure, which consists of different groups of beams; each group is characterized by its type and structural function. A tool is developed using Visual Basic.NET to directly interact with the CAD model during the optimization analysis. A function called "Fill Beams" reads the data and automatically gathers all the information necessary to solve a minimization problem for the weight and cost of the structure (Figure 4.5). This import includes some preliminary variables and constraints such as the dimensions of the possible beams and the functions to calculate parts of weights and costs of the module, these features have been implemented in VB.NET. After the data import, the designer can use the GUI Interface to implement additional constraints, variables, value domains, etc.


Show beams

Results	Type	Lenght	Material	Weight
Structure				518389,5
Columns				57546,13
413	Pipe	19000	S460	3573,13
417	Pipe	19000	S460	3573,13
423	Pipe	19000	S460	3573,13
431	Pipe	19000	S460	3573,13
439	Pipe	19000	S460	3573,13
798	Pipe	19000	S460	3573,13
800	Pipe	19000	S460	3573,13
804	Pipe	19000	S460	3573,13
808	Pipe	19000	S460	3573,13

FIGURE 4.5: Module structure imported from SAP2000 and exported in the .XML format (lengths are in meters and weights in kilograms).

The proposed test case considers an analytical calculation to estimate product cost and weight applied to steel structures used in oil & gas applications. Therefore, the cost and weight models are included into the CSP model. The constraints related to the engineering design of a steel structure are mainly focused on normative checks and conditions such as the structural limits to be applied. Other constraints are related to validation rules based on the technical know-how. These constraints can refer to the formalization of explicit and implicit knowledge. Examples can be validation rules to check the weight balance of assemblies into a steel structure. Another example could be rules to evaluate and limit the number of different types of components such as beams into an assembly. The cost analysis is performed using an analytical solver, which analyzes the model related to the Base Structure to recognize the information from the CAD model. The weight evaluation, which corresponds to the multi objective function, refers to the weight of the operative structure, including the primary and secondary structures. In particular, the weight value is analyzed using the CAD model. The configurations related to the sea and land transports have different weights related to a different set of specific accessories used for the corresponding transportation. However, this additional equipment has not been evaluated in this research.

The interaction between the CSP solver and the simulation software is realized using the code developed in Visual Basic.NET. This interaction has been performed with VB.NET and the steel structure's geometry. For the structural analysis, SAP2000® is employed as the FEM solver. The geometry has been simplified through a trusses model. SAP2000 provides an application programming interface (API) to develop a background control of the geometry, boundary conditions, loading combinations, calculation settings, checks and results to perform the connection between the CSP model and the interactive geometry. In fact, during the calculation of the CSP solution, the section dimensions of each beam and column were directly changed into the 3D geometrical model, in order to generate the updated trusses model for the analytical and numerical computing. Therefore, using Visual Basic.Net and these API utilities, a plugin application is developed to open a SAP2000® document, load/edit the CAD model, upload the parameters and geometry, and simulate the current model configuration. As reported, an .XML structure is delivered for each FEM simulation. Tekla® is used as the CAD system for the representation of the steel structures. A simplified geometry of the structure is generated by the CAD assembly model. This simplified geometry is then uploaded inside the FEM solver for the structural simulation. The output of the CSP tool is a reduced space of solutions, which is the domain in which the optimal set of parameters is searched. The CSP analysis represents the generation of the domain of the satisfied solutions to be evaluated within the simulation activity (Figure 4.6). Each satisfied solution can be considered as a set of values that solve the constraint satisfaction problem. Next, each set of data is imported into the Model-Based Simulation module to launch the numerical analysis in order to check the structural behavior. This task is automated using Visual Basic.NET and the API library provided by SAP2000®.

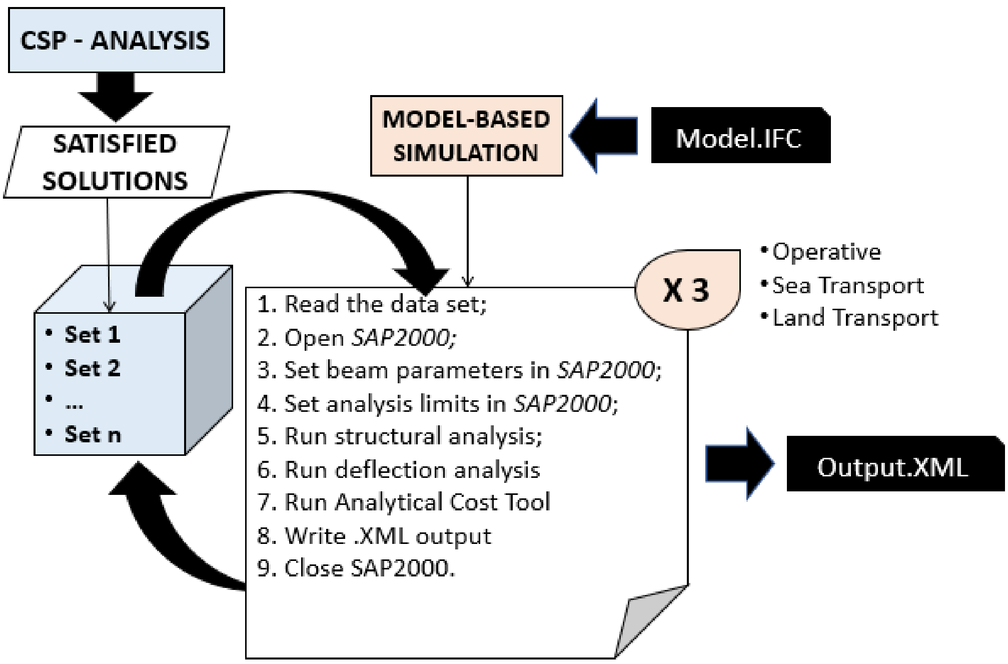


FIGURE 4.6: FEM and cost analysis: simulation workflow.

A tool is developed to read the data set related to each CSP solution and write this information into a .XLS file. This format is chosen for the management of the structured data, which contains the information related to each group of the beams, such as the geometrical section, height, material type, structural limits, and deflection limits. Next, all this information is used to set the solution parameters inside the SAP2000® model and run the relative analysis. All these steps are highlighted in Figure 4.6. These steps are repeated three times, one for each condition (operation, sea transport and land transport); in fact, each condition has a specific configuration of loading conditions to be applied. For each analysis, an output file is exported in the .XML format. This file contains the information related to the product structure with the results of the FEM analysis for each simulation. The .XML file (output) contains a general section, which includes the information of the number of beams with an overstressed status or an overdeflection set (Table 4.1). These data are related to each structure and each simulation. A second set of data contains the information pertaining to the stress and overstressed values achieved for each beam, including the information regarding the deflection state.

Output Value	Structure	Simulation type
Total number of overstressed beams	Primary; Secondary;	Operation; Sea transport; Land transport;
Total number of overstressed beams over 120 %	Primary; Secondary;	Operation; Sea transport; Land transport;
Total number of overdeflected beams	Primary; Secondary;	Operation; Sea transport; Land transport;

TABLE 4.1: Scheme of the output values for each FEM simulation.

The assembly of the structure is analyzed as a collection of groups. Each beam can be included in only one group of beams. A group of beams has items with the same cross-section. Moreover, the limits for the overstress and deflection conditions are defined for each group of beams. In particular, the deflection limit indicates the maximum value of the deflection allowed in the middle of the beam. For example, a L/100 limit describes the value to be considered for the maximum deflection, where L is the longitudinal length of the beam. This information is added by the designer using the GUI interface.

4.2 Case study: Oil and gas structures

The test case is focused on the design optimization of a large steel construction that weights approximately 700 t. The objective of the analysis is to reduce weight and cost during the early design phase. This steel structure is used in oil and gas systems called modules. A module is the smallest functional unit with its equipment, machines, and steel structure. Therefore, a single module can contain power generation units, gas compression units, and process equipment for oil and gas applications. The structure of a classic module consists of steel beams used for internal support of machines and equipment. These modules are prefabricated and later shipped to be situated in a place that is different from the construction site.

Modularization is a common design strategy in several engineering fields [86] such as oil and gas [29] where the products are pre-assembled and shipped. The modularity approach for oil and gas plants reduces the overall cost and the delivery time.

Regarding the design of steel structures with a CSP approach, the collection of the proposed constraints has been analyzed and discussed with an expert team of designers, in collaboration with an Italian enterprise which produces oil and gas solutions.

The steel structure that supports each module is analyzed in this test case. The overall structure can be divided into the primary frame (also known as the supporting frame) and the secondary frame (Figure 4.7). The secondary level of the structure provides the support for minor equipment. In general, the secondary frame includes braces to increase the stiffness of the construction.

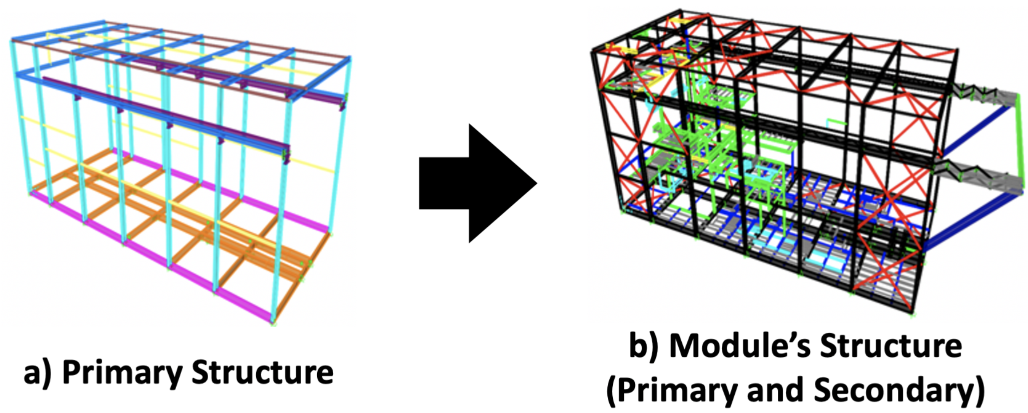


FIGURE 4.7: Representation of the primary structure (a) and the full module structure (b) (primary and secondary structures).

The classification of the primary and secondary structures is described in Figure 4.7). The Primary Structure is more important because it is composed of the elements that provide the structural resistance. If a failure affects one of these elements, the construction could collapse. In contrast, the Secondary Structure is composed by elements that cannot cause the collapse of the construction. A third category also exists, called the Tertiary Structure; however, it is not considered in the proposed design optimization workflow. This category includes the stairs, ladders, handrails, floor, roof, wall plates and maintenance/access platforms; the design of such elements is usually considered in the embodiment design phase.

Primary Structure	Secondary Structure
All main frames (columns, deck frames)	Braces
Gantry crane	Beams supporting floor
Monorail runway and supports for crane	Pipe supports
Stiffeners	Miscellaneous structure

TABLE 4.2: Classification of Primary and Secondary Structures.

4.2.1 Product structure

The product structure has been analyzed at the beginning of the research activity. The unified modeling language (UML) has been used for the representation of the product structure. Figure 4.8 shows the operative structure related to the analyzed steel construction. In particular, a composite structure diagram is applied to build the CSP model with an object-oriented representation of components, variables and constraints. In Figure 4.8 constraints are added to the diagram using the object constraint language (OCL) formalism.The highlighted diagram has been established using the Papyrus software and could help the designer to understand the structure of the existing systems and the formalization of blocks, variables and constraints. This diagram describes the relationships between the different components in terms of associations, dependencies and references (Figure 4.8). Moreover, the composite structure diagram adds the internal structure of the configuration and relationship of the parts. Therefore, each component is represented as a graphical class element.

As shown in Figure 4.8, the main component is the Operative Structure, which is the configuration of the steel structure in the operation phase. The correlated structures, known as the Sea Structure and Land Structure, inherit both the Primary Structure and Secondary Structure from the Operative Structure. However, additional information is necessary for modeling the phases of the shipping and road transport. The shipping and road transport are described and simplified using two abstract interfaces. The multiplicity of each relationship involving a structure is 1. The multiplicity is 1 to N for the groups of beams and the collections of the types of beams.

The geometrical variables are related to the geometry of each beam section. Two different types of beams are involved in this research: commercial beams and built-up beams. While commercial beams have hot

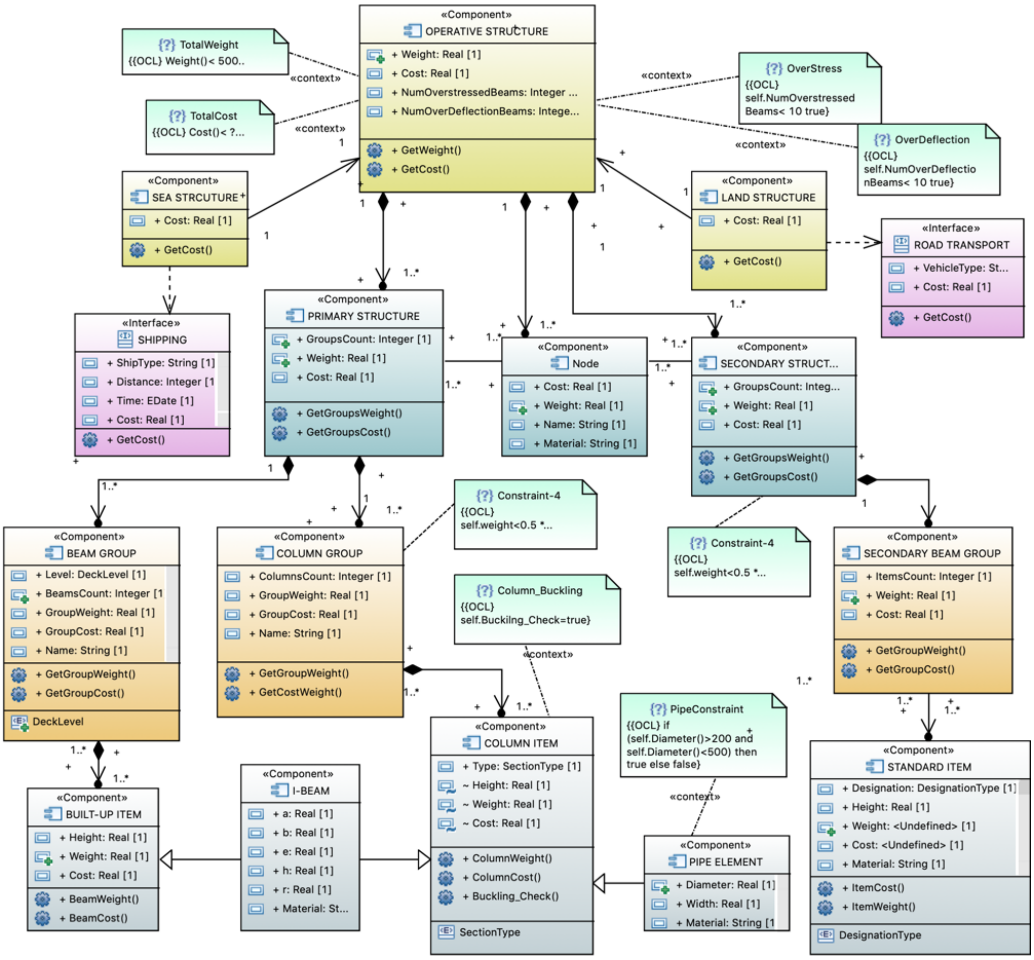


FIGURE 4.8: The composite structure diagram related to the analyzed product structure.

rolled laminated profiles with fixed cross-sections, the built-up beams are customized welded metal plates to shape the I beams (also known as double T type beams) or pipe beams (such as cylindrical columns). This paper considers standard beams for the Secondary Structure and built-up beams for the groups related to the Primary Structure. In this way it is possible to obtain more degrees of freedom in the design optimization of the Primary Structure. In fact, while a built-up section can be described using up to 4 geometrical variables, a standard beam is described only by a designation label, which can be seen as one variable related to the main geometrical parameter.

The generic representation of a beam object includes the parameters related to its section and length (here termed as height). The O-O representation of the product structure of such ETO structures is a collection of beams and nodes. Considering the classification highlighted in Table 4.2, the specific steel structure is divided into functional groups such as the "Central Columns", "Lateral Columns", "Main Deck - Traversal", "Main Deck - Longitudinal", "First Deck", "Second Deck - Traversal", "Crane Supports", "Piping Support", "Braces – First Level", and "Secondary Beam Deck".

4.2.2 Variables and constraints

An object-oriented model of the product structure is implemented inside the CSP Tool. In this model it is possible to define a set of variables 4.1 and constraints 4.2 for each component. The definition of such values and objects is an interactive process that involves the engineering team.

$$V = (V_1, V_2, \dots, V_n) \tag{4.1}$$

$$C = (C_1, C_2, \dots, C_n) \tag{4.2}$$

A CSP problem 4.4 is composed of:

- V: variables.
- C: constraints.
- D: variable's domains.

Continuous or discrete domains are defined for each variable. A set of n domains 4.3 is defined for solving the CSP problem 4.4. While a continuous domain concerns intervals of $\mathbb{R}$, a discrete domain is an enumerative and finite set of values (which can be Boolean, String, Integer etc.). Therefore, in this research, a mixed constraint domain is applied due to the conjunction of discrete and continuous constraints for the variables.

$$C = (D_1, D_2, \dots, D_n) \tag{4.3}$$

$$P = \{V, C, D\} \tag{4.4}$$

In terms of the CSP model, this thesis considers the arc-consistency approach [61] realized using the filtering process of the conjunction of constraints. The arc-consistency is used to guarantee the local consistency of the problem. Since the arc-consistency is 2 consistent, the node consistency (1 consistency) is guaranteed according to the definition of the k consistency provided by Kumar [58].

The solution search engine used in this case study is a simple backtracking algorithm combined with the constraint propagation method. This choice is due to the fact that dimension of the space of the solutions to be investigated is not so large. The use of the backtracking algorithm is necessary due to the lack of the proof of the global consistency. In particular, this algorithm is already implemented into the Gecode framework used for solving the CSP problem. Generally, backtracking, even if it is an algorithm dated, it is widely used in many recent research works [62] both in its base form or combined with evolutionary heuristics techniques.

The variables employed in the proposed test case are the geometrical representation of the steel structure. In particular, these variables correspond to the sizing of each beam section. As already said, two types of beams are involved in this study: standard and built-up beams. A subclassification corresponding to the built-up beams, which can be I section beams (double T) or cylindrical pipes, is established. The standard beams are described considering a list of different types. Therefore, the resulting variable for a standard beam is only one. However, the domain of the values related to each standard beam is defined by the designer on the basis of the type of the group of beams to be optimized inside the overall structure. The geometrical definition of a standard cross-section is collected in the database of the catalogs, instead the dimensions of the built-up sections can be defined by the designer. Table 4.3 and Table 4.4 show an example of a list of standard beams involved in the design process of the Secondary Structure.

Description	Material	Section height (h)	Type
HE200A	S355	200 mm	I shape
HE260A	S355	260 mm	I shape
HE300A	S355	300 mm	I shape
HE400A	S355	400 mm	I shape
HE500A	S355	500 mm	I shape
HE700A	S355	700 mm	I shape

TABLE 4.3: I Shapes standard.

Description	Material	Outside diameter	Thickness [mm]	Type
168.3X6.4CHS	S355	168.3 mm	6.4	Circular Hollow Sections
219.1X6.4CHS	S355	219.1 mm	6.4	Circular Hollow Sections
273.1X6.4CHS	S355	273.1 mm	6.4	Circular Hollow Sections
273.1X12.7CHS	S355	273.1 mm	12.7	Circular Hollow Sections

TABLE 4.4: Circular hollow sections standard.

Figure 4.9 illustrates the geometrical parametrization related to each type of built-up section, so for I beams and cylindrical pipes. On the left side, the I beam section is represented with 4 parameters: beam height (h), web thickness (a), flange thickness (e), and flange width (b). On the right side, the cylindrical section is represented with 2 parameters: main diameter (d) and pipe thickness (s).

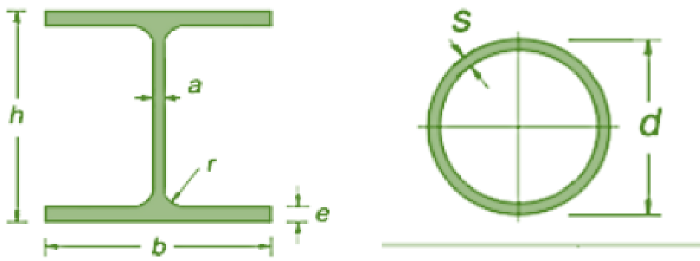


FIGURE 4.9: Example of parametrization for two different beam sections: I shaped beams and cylindrical pipes.

In this thesis, no parameter is assigned for the action related to the grouping of beams. The design team defines the setting of each group of beams in terms of the items included. Therefore, the design optimization is focused on the geometrical dimensions of each beam section. The Design Constraints were analyzed in an early phase and discussed with an expert team of designers. These constraints are focused on validation rules, which are mainly related to the domain of the technical expertise, and the constraints are described and formalized into the CSP problem using an analytical approach. These constraints refer to the formalization of the explicit and implicit knowledge. An example could be the validation rules to check the weight balance of the assemblies in a steel structure. Another example could be the rules to evaluate and limit the number of different types of components, such as the types of beams into an assembly. Therefore, the resulting list of constraints can be classified into four types:

- critical buckling;
- pipe sizing;
- weight ratio and balance;
- specification limits.

Finally, the Design Constraints are described with several examples that involve the practical satisfaction rules, which depend on the engineer's expertise.

Critical buckling

The set of constraints related to the critical buckling pertain to the ratio between the section area of each column and its vertical height, which is the length of the beam item. These constraints aim to avoid any possible buckling problem during operation, including the relative instability problem of the steel structure. The buckling problem involves the vertical structures, such as steel columns, when subjected to a compression load. This condition is characterized by a sudden sideways deflection; therefore, the cross-section of each column must be optimized to avoid this effect. The use of built-up sections for each group of columns allows each double T section and pipe to be optimized for the customized application.

Since carbon steel and stainless steel are high resistance materials, the relative double T beams have a small cross-section area. Therefore, these beams can be affected by compression problems. The buckling constraints are related to the specification provided by AISC 360-05, which regulates the limit factors for the width and thickness of a built-up I shaped beam. In particular, Eq. 4.5 describes the limiting width-thickness ratio (b/e) for a compression I beam under the condition of flexural compression:

$$\lambda_r = 0,95\sqrt{((K_C E)/F_L)} \tag{4.5}$$

where K_C is the coefficient for slender unstiffened elements 4.6, E is the modulus of elasticity (200.000 MPa in case of steel), and F_L is the calculated stress used in the calculation of the nominal flexural strength. In particular, F_L is calculated as being 70% of the specified minimum yield stress (F_y) of the compression flange (for example, for steel S355, F_L = 0,7 *x* 355 Pa). The K_C term is also evaluated using Eq. 4.6, and its value must be among the range expressed in 4.7.

$$k_c = \frac{4}{\sqrt{\left(\frac{h}{a}\right)}} \tag{4.6}$$

$$0,35 \leq k_c \leq 0,76 \tag{4.7}$$

The circular hollow sections (pipe beams) under uniform compression have a different limiting width-thickness ratio formula 4.8, which limits the D/s ratio. More constraints related to the pipe beams are analyzed in the following part.

$$\lambda_r = 0,11E/F_y \tag{4.8}$$

These limits of slenderness are critical for I shape beams. This type of constraint ensures that the design solutions are free of the buckling problem before high computational cost FEM analysis is performed. In this way, a slender beam with a possible buckling problem is eliminated during the CSP analysis. The limits highlighted in this section were modified by the experts to provide a more restrictive condition for the described constraint.

Pipe sizing

The rules related to the pipe sizing limit the usage of non-suitable pipes in terms of the stability and static resistance. A mathematical formulation is used to describe the ratio between the diameter and width 4.9. Moreover, a second limit is applied to the diameter range 4.10. In this constraint the designer can reduce the range of possible values for each pipe section (for example, a possible diameter range can be between 400 mm and 600 mm).

$$14 \leq \frac{D}{s} \leq 36 \tag{4.9}$$

$$D_{min} \leq D \leq D_{max} \tag{4.10}$$

Weight ratio and balance

This set of Design Constraints is related to the weight ratio and balance between each group of beams to avoid non suitable configurations. A list of such constraints is presented as an example in Table 4.5. The terms C1, C2, ..., C5 represent the numerical value for each constraint definition.

Specification limits

This set of design constraints involves the limits related to the maximum values of the weight, cost, number of different beam sections, etc. Therefore, in this thesis, the maximum weight and the maximum cost are the examples of two specification limits. The CSP solver can analyze the sum of the costs and weights related to all the groups of beams. Additional cost and weight factors are considered for the assembly, nodes, and transportation. Therefore, using the proposed analytical approach, the constraints related to the specification limits can evaluate if a set of parameters can satisfy these general limits. These constraints can be evaluated because an O-O model is implemented into the CSP Tool.

Example of constraint	Description
Constraint 1	the weight of the main deck (at zero level) is greater than that of the structure of the first deck, which Is greater than the weight of the second deck x C1;
Constraint 2	the weight of the main deck (at zero level) is greater than C2 x the weight of the equipment and machineries;
Constraint 3	the weight of the main deck (at zero level) is less than C3 x the total weight of the steel construction;
Constraint 4	the weight ratio between the secondary and primary structures is less than C4;
Constraint 5	the weight ratio between the groups of columns and primary structure is greater than C5;

TABLE 4.5: A list of the constraints related to weight ratio and balance.

4.2.3 Cost modeling

Two levels of cost analysis are considered in this thesis: parametric and analytical.

Parametric cost level

The parametric level is used during the CSP analysis to evaluate the constraints related to the cost of the structure. As analyzed previously, the cost evaluation is a constraint classified as a specification limit. This approach for costing means that should be applied a method that involves an analysis using only simple formulas based on the product parameters such as weight, sizing and material [32]. The CSP solver considers the cost related to the Primary and Secondary structures. This cost evaluation refers to the analysis of each group of beams. The total cost is the sum of the costs evaluated for each beam. In the first level only a few parameters are considered: beam weight, material (m), section type (st). Eq. 4.11 describes the cost related to one beam, as implemented for the first level of the cost analysis.

$$Cost_{Beam_I} = W_B \cdot SC_{(m,st)} \tag{4.11}$$

The term W_B in 4.11 represents the weight of the element in kg, and the term $SC_{(m,st)}$ is the specific cost related to the selected beam. This specific cost is evaluated in €/kg; its value is not constant and depends on the type of section (st) and the type of material (m). Table 4.6 presents an extract of the cost database used for the first cost evaluation implemented into the CSP model.

Section Type	Material Type	Specific Cost [€/kg]
HE200A	S355	0,75
HE200A	S275	0,70
HE500A	S355	0,77
HE500A	S275	0,72
273.1X6.4CHS	S355	0,85
...	...	...
BUILT-UP: Circular Shape	S355	0,68
BUILT-UP: I Shape	S355	0,70

TABLE 4.6: Specific cost [€/kg] related to the section types and materials.

Analytical cost level

The analytical method considers the analysis of each elementary task related to the product fabrication [32]. Therefore, this approach considers tasks such as cutting, welding and assembly. The level of detail depends on the availability of formulas and database for each test case. Eq. 4.12 shows the total cost (evaluated in €) related to the steel construction described in this Chapter. The cost is the value analyzed after the FEM simulation. The term $Cost_{STR}$ is the cost related to the construction of the steel structure in the operation condition. Thus, this term includes the cost related to the Primary and Secondary structures with the relative set of nodes and beams. The term $Cost_{TRA}$ represents the cost of the transportation. This parameter includes the sea and land transport 4.13. Finally, $Cost_{ERE}$ is the cost related to the construction erection and installation on site. This cost involves only the steel structures, without considering any equipment or machinery, and it is correlated to the structure self weight (W_{ERE}) and a specific cost per weight (SC_{ERE}), as highlighted in 4.14.

$$Cost_{TOT} = Cost_{STR} + Cost_{TRA} + Cost_{ERE} \tag{4.12}$$

$$Cost_{TRA} = Cost_{SEA} + Cost_{LAN} \tag{4.13}$$

$$Cost_{ERE} = W_{ERE} + SC_{ERE} \tag{4.14}$$

For the transport cost ($Cost_{TRA}$), Eq. 4.15 and 4.16 describe the approach for the sea and land transport. In particular, the transportation cost includes a fixed cost ($Cost_{FIX}$), a variable cost related to the distance (d), a specific cost (SC_{SEA} and SC_{LAN}) related to the vehicle, and a cost related to the additional structures necessary for the transportation ($Cost_{STR-SEA}$ and $Cost_{STR-LAN}$). The fixed cost is related to the accessories and operators involved in the transportation phase. The cost related to the additional structures concerns objects such as braces that are added into the Sea and Land Structures to limit the deflection due to the transportation phases. The cost of braces is an additional item because they are objects not included into the CAD model that represents the Operative Structure. In particular, the braces are disassembled before the installation.

$$Cost_{SEA} = Cost_{FIX-SEA} + (SC_{SEA} \cdot d_{SEA}) + Cost_{STR-SEA} \tag{4.15}$$

$$Cost_{LAN} = Cost_{FIX-LAN} + (SC_{LAN} \cdot d_{LAN}) + Cost_{STR-LAN} \tag{4.16}$$

The cost related to the steel construction ($Cost_{STR}$) is analyzed as the sum of three parts:

- Raw material cost
- Fabrication cost
- Assembly cost

The raw material cost corresponds to built-up beams, standard beams and nodes 4.17. In particular, built-up beams and nodes involve a set of sheet metal plates that compose the final shape after a welding process. The fabrication cost pertains to the manufacturing of beams and steel nodes. The beam fabrication includes the cutting ($Cost_{CUT}$) and welding ($Cost_{WEL}$) costs for the built-up profiles; however, the fabrication of standard beams includes only the cutting cost because the shape of these beams is produced using a hot rolling process 4.18.

$$Cost_{MAT} = \begin{cases} \sum_i^N Cost_{PLA} \; for \; nodes \; and \; builtup \; beams \\ \sum_i^N Cost_{STD} \; for \; standard \; beam \; sections \end{cases} \tag{4.17}$$

$$Cost_{FAB} = \begin{cases} \sum_i^N Cost_{CUT} \; for \; all \; items \\ \sum_i^N Cost_{WEL} \; for \; standard \; beam \; sections \end{cases} \tag{4.18}$$

Eq. 4.19 describes the formula that represents the cost related to the cutting process, which includes the preparation of the plates for nodes, built-up beams and standard beams.

$$Cost_{CUT} = \frac{\left(l_{EDG} \cdot \frac{1}{v_{EDG}} \cdot HC_{CUT}\right)}{3600} \tag{4.19}$$

The cutting process involves the cutting edge (l_{EDG}), which is related to the geometry of the part to be manufactured, the cutting velocity (v_{EDG}) and the hourly cost (HC_{CUT}) of the cutting process used. Using a similar approach, Eq. 4.20 introduces the calculation of each welding with the welding velocity (v_{WEL}) and the related hourly cost (HC_{WEL}).

$$Cost_{WEL} = \frac{\left(l_{EDG} \cdot \frac{1}{v_{WEL}} \cdot HC_{WEL}\right)}{3600} \tag{4.20}$$

While the geometrical data were acquired from the CAD model, the other costing parameters were previously defined into a database. The average values were selected for parameters such as the cutting and welding velocity, specific costs and other parameters. Figure 4.10 describes an example of the I shaped beam with the welding edges highlighted.

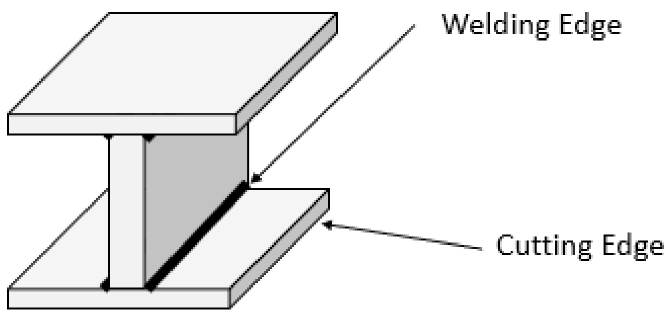


FIGURE 4.10: Example of an "I shaped" beam.

The assembly cost is considered as a function of nodes and beams (4.21) that uses empirical parameters analyzed from past projects. This cost depends on the number of nodes (n_{NOD}) and beams (n_{BEA}) respectively multiplied for the specific assembling cost (€/kg) defined for each node (SC_{NOD}) and each beam (SC_{BEA}). All the data related to the number and weight of each element are retrieved from the CAD geometry.

$$Cost_{ASS} = (n_{NOD} \cdot SC_{NOD}) + (n_{BEA} \cdot SC_{BEA}) \tag{4.21}$$

4.2.4 Loading conditions

The steel structure of a module is typically subjected to static loads, dynamic loads, land transport loads, sea transport loads and fatigue loads. The modules are prefabricated in a construction site and later shipped by sea to the destination for the installation. Therefore, the weight optimization of such ETO constructions is a critical issue. The simulation tools based on an FEM analysis are used in the design of the steel structures to allow the investigation of the behavior of mechanical structures with virtual prototypes. Figure 4.11 shows the simplified FEM model, which reproduces the steel module with the Primary and Secondary structures. In this chapter, the structure of the model is simulated using SAP2000®. The geometrical simplification considers 1D beams and does not include the Tertiary structure.

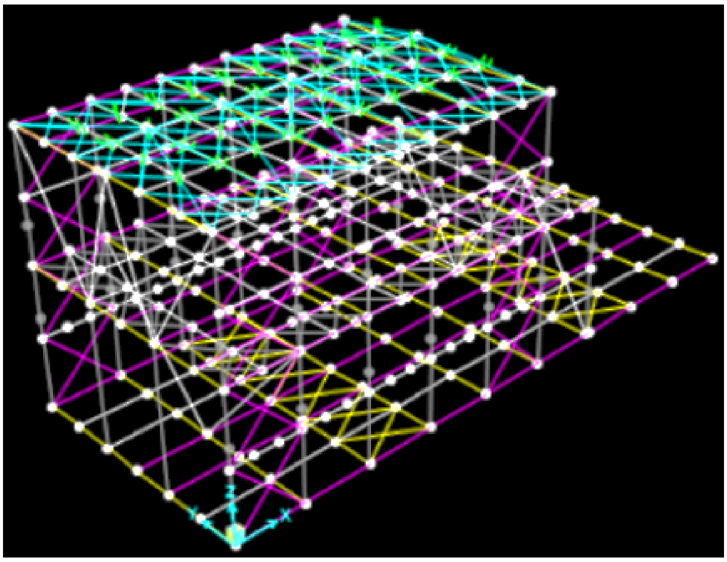


FIGURE 4.11: The structure of the module including the primary and secondary structures, as modeled using the FEM solver.

The specific load conditions to be considered during the design process are related to the installation phase (geographical region, local conditions, normative rules and regulations, etc.). In this context, the resistance limits for a structure to be designed are defined and agreed with the customer on the basis of specific normative standards. For example, the ASCE/SEI 7-05 [8] is a typical standard used for the definition of the design loads and cases to be applied in the structural analysis for the North American areas. The strength criteria for a structural analysis are defined by normative standards such as the ANSI/AISC 360-10 [1]. Generally, the design loads considered for a steel construction inside a national/international normative standard are as follows:

- Dead loads: the total weight of the structure with all the equipment, machineries, and accessories.
- Live loads: variable loads related to the use of the structure and their accessories such as a crane.
- Wind load: the action of the wind in each direction (+X, -X, +Y, -Y).
- Seismic load: the simulation of the dynamic load related to a possible earthquake. Response spectra are particularly useful tools for analyzing the performance of structures and equipment under earthquakes.
- Operative pressure: the nominal operative pressure inside the structure (if present).

Different normative standards propose a combination of loads to be evaluated in different design cases. Eq. 4.22 shows a typical load combination with a dead load ($Dead$), wind action in the $+X$ direction ($Wind_{+x}$), live forces in the $+X$ direction ($Live_{+x}$), and seismic load as the response spectra in the X direction($Seismic_{ResponseSpectra-x}$). The coefficients ($A$, B, C, and D) are calculated on the basis of the related normative standards to be applied in this specific case. The design of a steel structure includes the analysis of different load combinations with different numbers of loads applied. Each combination provides different values for the highlighted coefficients. Combinations and coefficients are related to the specific normative standard.

$$A \cdot Dead + B \cdot Wind_{+x} + C \cdot Live_{+x} + D \cdot Seismic_{ResponseSpectra-x} \tag{4.22}$$

4.3 Multi Objective Function

A multi objective function is used to evaluate each simulation in terms of the four targets related to the weight, cost and the number of overstressed and overdeflected beams. This objective function is described in Eq. 4.23:

$$f_W \cdot Cr_W + f_C \cdot Cr_C + f_{OS} \cdot Cr_{OS} + f_{OD} \cdot Cr_{OD} \tag{4.23}$$

where:

- f_W = the weight factor assigned to the weight reduction (first criterion)
- Cr_W = first criterion related to the weight reduction
- f_C = the weight factor assigned to the cost reduction (second criterion)

- Cr_C = second criterion related to the cost reduction
- f_{OS} = the weight factor assigned to the overstressed beam reduction (third criterion)
- Cr_{OS} = third criterion related to the overstressed beam reduction
- f_{OD} = the weight factor assigned to the overdeflected beam reduction (fourth criterion)
- Cr_{OD} = fourth criterion related to the overdeflected beam reduction

The described function 4.23 considers four criteria, follows the quality characteristic small is better (SiB). The SiB characteristic means that a small value is better and achieves a greater score in the function calculation. The description of the SiB approach is described in Eq. 4.24, where $value_{MAX}$ and $value_{MIN}$ represent the maximum and minimum values allowed in the optimization analysis. These values are defined by the designing team. For example, the designer can define as a target a weight reduction of approximately 20% considering the maximum and minimum values. The terms such as f_W, f_C, f_{OS}, and f_{OD} are the factors that regulate the weight of each factor into the calculation of the objective function 4.23; the sum of these factors is 1.

$$Cr = \frac{value_{MAX} - value_i}{value_{MAX} - value_{MIN}} \tag{4.24}$$

4.4 Results

This section presents the results related to the design optimization of the steel structure described in the previous Section. The optimization is based on a 700 t construction that obtains a weight reduction of approximately 10% with a cost reduction of approximately 12%. The reference structure and the loading conditions are generic and not related to an existing construction. Table 4.7 lists the variables defined and analyzed in the constraint analysis using the CSP tool. A number of 20 parameters were selected to represent 10 groups of beams. More than 60 million combinations were analyzed and elaborated using the CSP tool in approximately 1 hour and a quarter.

The result of the constraint-based analysis performed using the CSP tool by employing the MiniZinc and Gecode toolkit is described by the tree view represented in Figure 4.12. Green elements represent the satisfying solutions, red elements represent a combination of variables pruned because they do not satisfy some constraints, the yellow element represents the optimal solution. A total of 62 satisfied solutions exist. These solutions

Group Type	Group Name	Section Type	Variable list	Variable count
Primary Structure	Column 1	Circular Hollow	D, s	2
	Column 2	Circular Hollow	D, s	2
	Main Deck Traversal	I beam	b, h	2
	Main Deck Longitudinal	I beam	b, h	2
	1° Deck Traversal	I beam	h	1
	1° Deck Longitudinal	I beam	h	1
	2° Deck Traversal	I beam	h	1
	2° Deck Longitudinal	I beam	h	1
	Crane support	I beam	h	1
Secondary Structure	Piping support	Standard beam	h	1
	Braces First Level	Standard beam	h	1
	Braces Second Level	Standard beam	h	1
	Secondary Deck 1	Standard beam	h	1
	Secondary Deck 2	Standard beam	h	1

TABLE 4.7: Variables related to the test case for the CSP analysis.

are ordered by increasing weight. The domain of solutions is further analyzed into the optimization loop by using the numerical FEM tool and the described analytical cost tool.

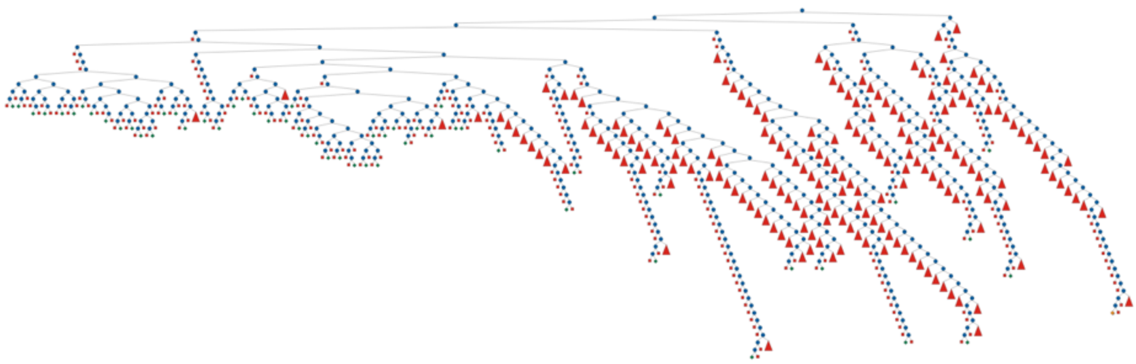


FIGURE 4.12: A graphical solution of the CSP problem.

Figure 4.13 shows the result of the optimization after the simulation phase. The simulation phase lasted in approximately 20 hours. Each possible solution is evaluated in the simulation loop by using the multi-objective function defined in the previous Section. The graph exhibits a parabolic behavior, which maximizes the solution with a better compromise among cost, weight and number of resulting overstressed and overdeflected beams. In this case study, the optimum solution is represented by a configuration that achieves a weight of approximately 630 t.

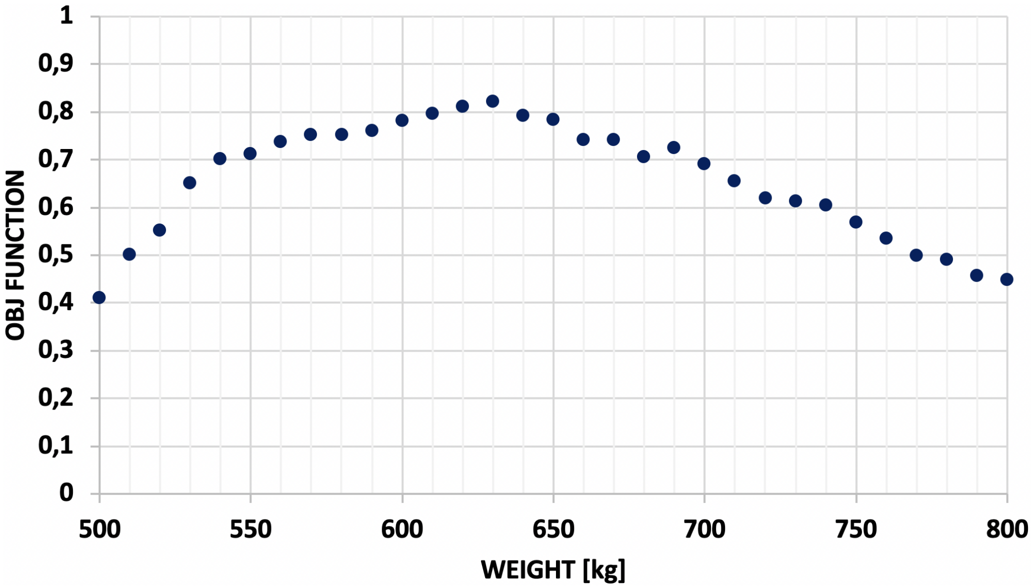


FIGURE 4.13: The relation between the OBJ function and the weight of the structure.

This configuration provides a limited number of non-verified beams that must be reinforced during the engineering detail phase. Heavier solutions are not suitable because they increase the cost of the structure. However, lightweight solutions increase the number of non-verified beams with overstressed and overdeflection problems.

Chapter 5

Application in the configuration of lift's panels

During the last year, my research concentrates on the development of a final generic software tool to support an industrial company to design products according to different requirements. The software is a valid support for the configuration of the ETO products of the technical department, for orders and quotations management for the commercial department and finally for the proposal submission for the customer.

The software can manage such aspects leveraging company design knowledge stored in a database. This framework is based on the aspects of design configuration.

The case study presented to validate the method and the framework concentrates on the configuration of lifts' panels.

The case study presented in Chapter 4 and the one presented in this Chapter apply and validate the same method. They have been both fundamental for my research and they are correlated since the solver tested and stressed in the Chapter 4 has been improved and incorporated in the framework presented in this Chapter. This final framework has been thought to be more general in order to be an advanced starter point to be reused for support the configuration of other ETO products.

The results shows that using the solver based on the CSP problem of Chapter 4 combined with a preliminary Design Structured Matrix analysis to split into sub-problems the original problem make the configuration phase agile and rapid, moreover the possibility of human intervention of the administrator directly on parameter and rules let this framework to be open and improved directly by the user without further work of the programmer.

5.1 Development approach

The collaborative design platform proposed in Chapter 3 (Figure 3.2), that performs Human-Computer interaction, is proposed to support the design of ETO products by integrating a constraint-based optimization approach. The approach is defined for the design of generic ETO products with the problem of the normative and dimensioning analysis. In particular, the methodology is now applied in the context of lift's panels.

This framework to be developed takes about one year and three months, about three months to acquire the knowledge mandatory to build a usable program and about one year to develop the program. The one year of development comprehends also the correct implementation in the company. In fact the framework has been build with the continuous feedback of the company: every time some group of new stand-alone functionalities has been added to the software a portable tool has been deployed in order to be tested from the company. This time has been also so long since it included the standardization of some parts of the program, as the managing of orders, quotations and product structure, that can be reused for further applications in other contexts and reduced of about six months the new developments.

This has been the time to built a program self consistent and usable in a company context, after this implementation other improvements have been discussed with the lift's panels company involved in this case study in order to make the software more compliant with their needs. These improvements have been developed in the part non standardized of the framework since they where specific requests not applicable in other contexts.

The CSP model implements a set of variables, values and constraints defined during the Collaborative Design level like in the previous Chapter, moreover the previous solver as been improved by combining it with a preliminary Design Structured Matrix analysis to split into sub-problems the original problem. This is fundamental to make the configuration phase agile and rapid.

The solver has been improved by including a Constraint Satisfaction Problems (CSP) solver based on backtracking newly implemented within the company (the solver developed in the Chapter 4 uses the freeware software Minizinc based itself on a free resolution kernel called Gecode). The CSP solver works with a solution library that have been developed in Hyperlean.

As in the previous Chapter, this CSP model considers the arc-consistency approach [61] realized using the filtering process of the conjunction of constraints. The arc-consistency is used to guarantee the local consistency of the problem. The solution search engine used in this case study is a simple backtracking algorithm combined with the constraint propagation method. As in the previous Chapter, this choice is due to the fact that dimension of the space of the solutions to be investigated is not so large. The use of the backtracking algorithm is necessary due to the lack of the proof of the global consistency.

The final solver that is obtained works creating from the configuration rules the Design Structured Matrix (DSM) which is useful to understand the relationship between the parameters, which are the variables of rules. This is fundamental to divide the initial problem into several sub-problems. In practice, once the DSM has been created, it is triangulated in order to understand the various clusters of dependent configuration parameters. For the independent parameters the solver that starts is the one that was already present in the company, for the parameters that instead depend on each other in the assignment of the values the new CSP solver starts. The solver algorithm is better explained below.

DSM = { *parameters ordered and clusterized* };
DSM = $\bigcup_{i=0}^{n}$ *Cluster(i) with Cluster(i)*= { *some parameters of the DSM*};
Cluster(j)∩*Cluster(k)* = ∅ $\forall j, k = 1 ... n$;

For each *parameter* ***in*** *DSM*
  If *parameter* ***IsNot*** *assigned* ***then***
    If *cluster of parameter has only one element* ***then***
      -Find all rules that contain parameter;
      -Assign a value to parameter with the solver already present in Hyperlean;
    Else
      -Find all rules that contains the parameters of the cluster in which is contained parameter;
      -Solve rules with the new CSP solver based on backtracking and assign the result values to the parameters of the cluster;
    EndIf
  EndIf
Next

The final solver has been implemented using the programming languages C# and VisualBasic. The part regarding the DSM uses some classes specialized in the manipulation of matrices and a method based on the simulating annealing, a stochastic technique, for the clustarization. The part regarding the new CSP solver is instead based on the backtracking technique. The process (reminded in Figure 5.1) consists to search a solutions pruning the combination of parameters that do not satisfy some constraint. If a path must be pruned, it can come back to the node that is not visited jet without the risk to move in a path already explored. Although the algorithm is not so efficient for not NP-complete problems since it has an exponential complexity (Chapter 3, Section **Constraint satisfaction problems**), for the purpose of this thesis it has been considered to be sufficiently fast thanks to the decomposition of the initial problem in many sub-problems. This trick, experimented in the following case study, makes the resolution of the CSP almost instantaneous.

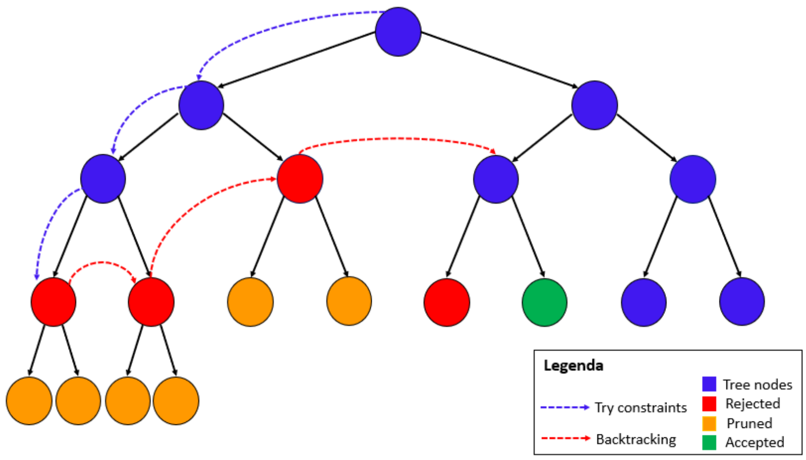


FIGURE 5.1: Backtracking search process

Figure 5.2 describes the configuration and optimization workflow, which is all included in the configuration software. The CSP tool is employed to reduce the space of solutions. The optimization integrated in the CSP tool finalize the assignment of the variables and parameters.

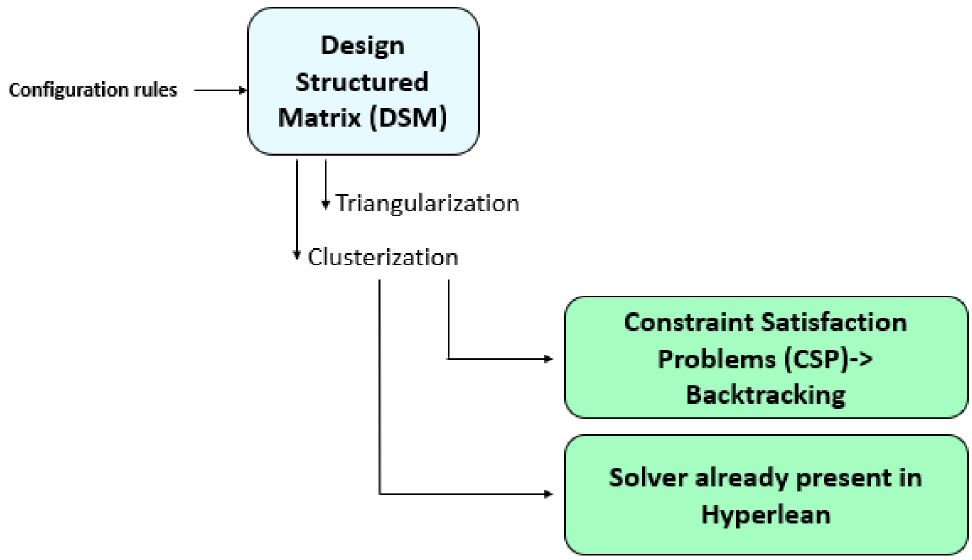


FIGURE 5.2: The optimization workflow integrated in the CSP solver.

The resulting tool consists of a kernel to manage the configuration, in particular the parametric model, the CSP Solver and the GUI Interface. The CSP Solver uses libraries already present in the Hyperlean company improved with some techniques explained before to decrease the solver time. The solver together with the kernel works in background and gather the result in the GUI Interface, giving also a graphical visualization of the paramatric model in order to ease the user during the configuration workflow. Furthermore, the GUI Interface shows the visual commands for the management and definition of the parameters, constraints, values and the management of the commercial part. This last part is based on the information exchange between the database and the kernel of the software.

Figure 5.3 describes the software architecture.

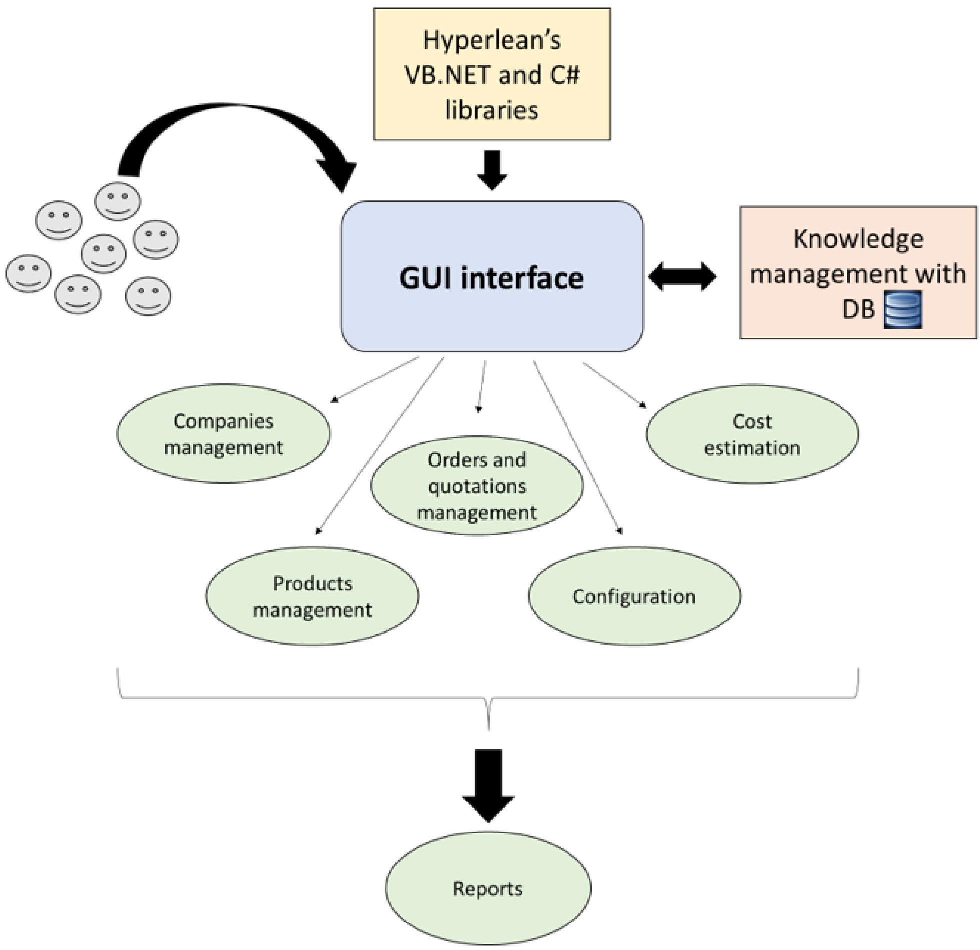


FIGURE 5.3: Software architecture of the tool.

Then a Graphical User Interface (GUI) has been defined for the management of the CSP problem, the design objectives and the commercial part (second step). Figure 5.4 shows the proposed GUI, which consists of three levels: order and quotation management, configuration management and product management. The first level concerns the management of orders and quotations, here the user is able to create new orders and quotations, modify and delete them. The configuration level, posed at the center of the interface to underline the central role of this part, regards by the side of the user role the choice of parameters to configure the ETO product and by the side of the administrator role, the definition of the mathematical rules to be verified in the CSP calculation. Each constraint is edited using a mathematical representation of formulas and rules. This definition is an object-oriented (O-O) representation of the product structure. The generic classes are defined in Visual Basic.NET language to support the O-O representation of a product structure. In particular, a generic set of must inherit classes is implemented to perform the generalization of the software predisposed to the specialization. These classes provide the parent-child relationship between each component of the product structure. This O-O representation includes the definitions of properties, methods, variables and constraints. The configuration level shows also a 3D graphical representation that uses graphical classes to obtain the rendering of the model. In the third level a product tree shows the products of the catalogue filtered by the role visibility.

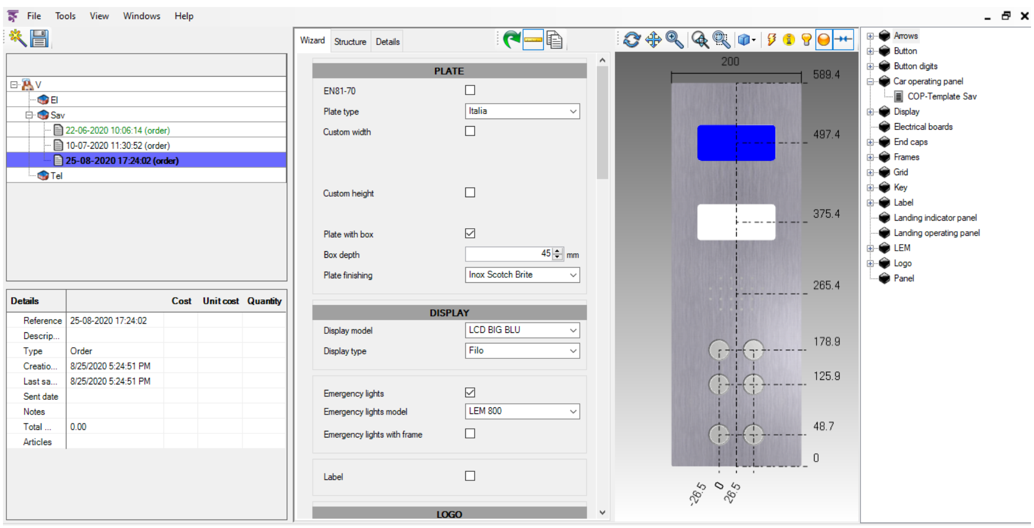


FIGURE 5.4: The final GUI Interface.

The proposed test case considers an analytical calculation to estimate product cost of lift's car operation panels applications.

The constraints related to the engineering design of a car operating panel are mainly focused on normative checks and conditions such as the dimensional limits to be applied. Other constraints are related to validation rules based on the technical know-how. These constraints can refer to the formalization of explicit and implicit knowledge.

The cost analysis is performed using an analytical solver, which analyzes the model related to the Base Structure to recognize the information from the CAD model.

The assembly of the structure is analyzed as a collection of groups. For the car operating panel these groups are: the plate, the display, the emergency light, the floor buttons and the service buttons. Some example of constraint can be: each group must be rightly outdistanced from the others, some choice must be of default (like the presence of end cups with some type of plate) and some choice must exclude some other choice (like the presence or not of frames for some types of display).

5.2 General purpose

The software has two different logins which depend on the role of the user. The administrator login gives access to all the functionalities of the software: the management of products, pricelists, configuration parameters and rules, the associations of products to the customers and the general database. The customer login gives access to less functionalities: the configuration of a product and the preparation of orders or quotations to sell to the supplier in order to obtain the proposal submission. This exchange is managed by the database that is shared between supplier and customers.

The software has been developed on libraries build in the company that co-finances my PHD, Hyperlean. Libraries are written in C# or in Visual basic using Framework.NET 4.0 and the final software is written in Visual Basic also using the Framework.NET 4.0. The products can be visualized through a viewer also developed by Hyperlean.

5.2.1 Administrator workflow

The initial work of the administrator is always to formalize the technical knowledge in order to obtain configurable products to present to the customer. This can be possible using a language not very difficult of mathematical rules. The Figure 5.5 shows the potentiality of the administrator login.

The administrator have a complete management of the database, in the database all of the knowledge formalization and the customer management are contained. The GUI interface simplifies the work to modify the database using apposite forms. The administrator can directly act on parameters and rules this let this framework to be open and improved directly by the user without further work of the programmer. This is an example of human intervention. Moreover the administrator can add or

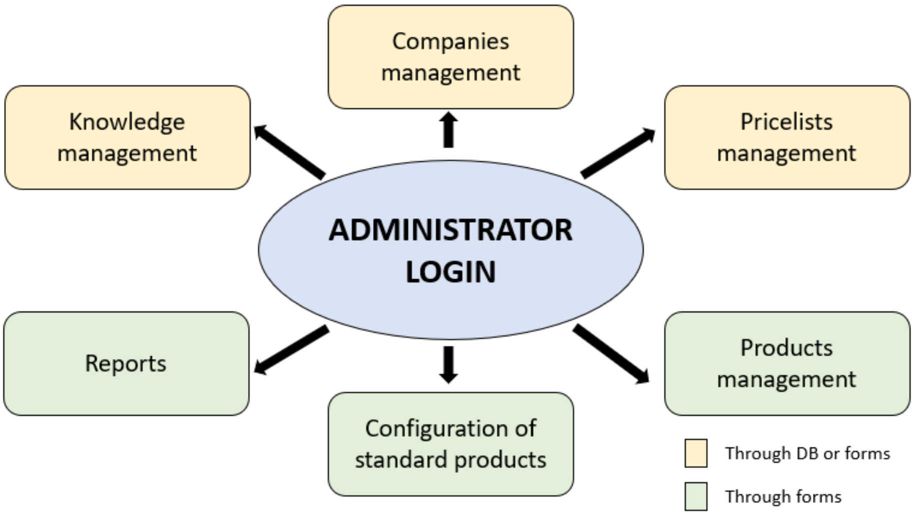


FIGURE 5.5: Potentiality of the administrator login.

remove customer companies and he can associate a pricelist to each customer to the cost estimation of standard products. He can also manage the products (Figure 5.6), adding or removing them (the addition use the STEP format), and making some product visible to some customer instead to another customer.

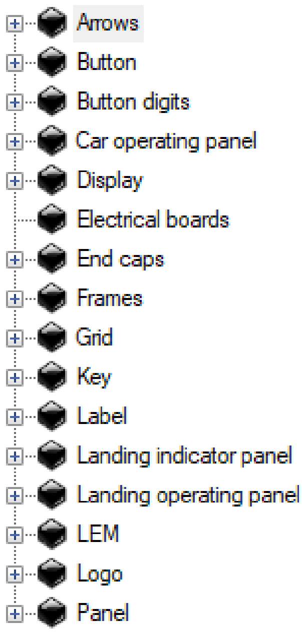


FIGURE 5.6: The part of the GUI interface dedicated to the products management.

The administrator can also prepare some configurable standard product to add to the database (Figure 5.7), this is important to ease the creation of the order or the quotation of the customer.

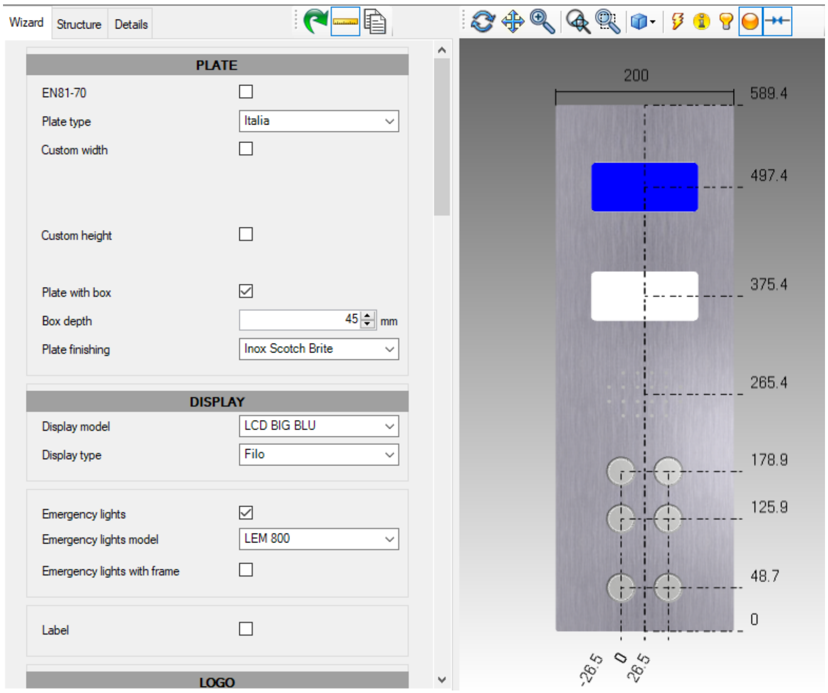


FIGURE 5.7: The part of the GUI interface dedicated to the configuration management.

Once the customer has send orders and quotations the administrator is able to evaluate them and to print the reports.

After the formalization and the products addition the administrator must prepare a new customer area. He must follow the following steps:

- add the customer;
- add the pricelist for the customer, if it is not already present;
- preconfigure template products that the customer orders in general;
- make visible products that the customer can order.

Once the customer area is prepared he can give the software to the customer with a user login.

5.2.2 Customer workflow

The possible software uses for the customer are outlined in Figure 5.8

Once the knowledge is formalized, so all the parameters and rules have been inserted in the database, if a configurable product is open in the software, the configuration of the product can starts. The user chooses all the parameters to configure the products and the solver, using the rules inserted from the administrator, assigns the parameter's values. The user, once the configuration has been finished, can add the product to is order

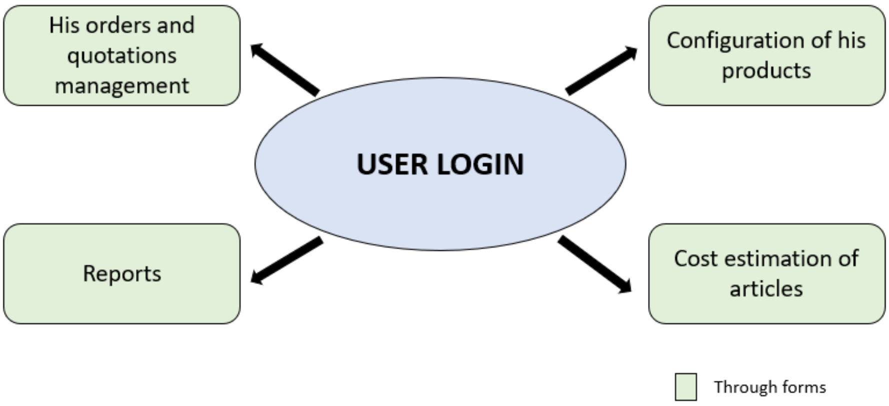


FIGURE 5.8: Potentiality of the user login.

and now the product is an article with its cost associated. The cost is derived from the pricelist of the customer and from parametric rules for mechanical processing on the products when are assembled, principally welding processing, mounting processing and laser cutting processes.

The order or the quotation can be modified until it is not send to the seller (Figure 5.9).

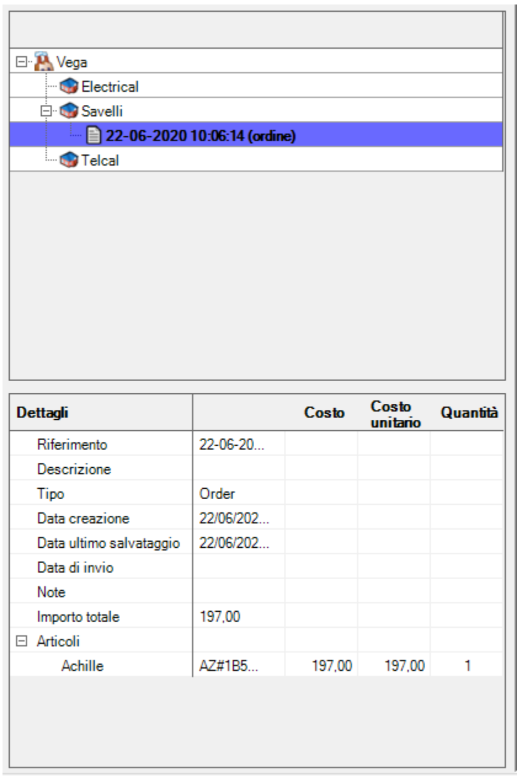


FIGURE 5.9: The part of the GUI interface dedicated to the orders management.

Finally also the customer is able to print the report.

5.3 Case study: lifts' panels

The general software has been specialized, as already said, for a company that works in the field of lift's panels. In particular it can configure: car operating panels (COP), landing operating panels (LOP) and landing indicator panels (LIP). COP is a panel posed at the internal of the lift, LOP is the panel to call the lift, LIP is the panel that shows the position of the lift.

The knowledge of the company has been formalized in parameters and mathematical rules that have been settled into a shared database which is used from the configurator engine to import the mathematical problem, the CSP problem, that will be settled by the solver described in the previous section.

The products of the company have been imported also into the database in order to have a graphical representation in both case: if they are individually opened or if they are assembled with the rest of components.

At the end of the configuration the user, which have a pricelist associated, can save the article into his order or quotation and he can see the final cost associated to the configurable product. The order or quotation can be modified until the user decides to send it to the seller.

This section will concentrate on the part related on the configuration of a COP. The car operating panel is an interface in an elevator used to select the floor to which passengers wish to travel, consisting of a car button for each landing served by the elevator, a display showing the floor passage, an emergency stop light, an alarm button, door-open and door-close buttons and a number of key switches used by building management, maintenance and emergency personnel. In Figure 5.10 it is represented an example of a configurated COP obtained by the software.

According with the company, the COP has been divided in the following modules:

- plate
- display
- logo
- grids
- additional boards
- floor buttons
- service buttons

FIGURE 5.10: Car operating panel (COP) rendering produced by the software

5.3.1 Plate

The plate is a part of a machine sheet metal that contains all the components of the panel.

Different models can be chosen: they can be simple rectangles obtained from a sheet metal or they can be models obtained from a bent sheet metal so with a certain thickness. Models with thickness can contains all the electronic parts without the need of an excavation in the wall in where they are mounted.

Models with thickness have a standard width since the ends are closed with caps with predefined lengths. For this models dimensions can variate but they must satisfy numerous constraints.

However, as can be seen from the part of the wizard dedicated to the plate (Figure 5.11), not only the parameters dedicated to sizing are variables, but also parameters dedicated to a possible box containing electronics or wiring necessary for the customer, for finishing plate and fixing the plate can be varied.

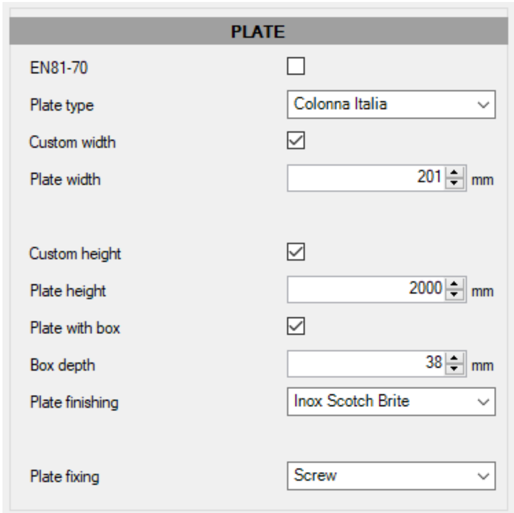


FIGURE 5.11: The wizard part related to the plate

In some cases, a standard plate may be requested by the customer. This simply means that the components must meet certain standards with respect to the height at which they are located; for example the height positioning of the first button or the height range within which the screen must be found.

5.3.2 Display

The display is that part of the panel containing a screen, most often digital, which shows the scrolling of the elevator floors or service communications, for example in the event of breakdowns.

The main parametrization of a display is shown in the Figure 5.12.

There are also several models here that differ in features such as: supporting or not supporting an image, having the possibility of having a frame including the choice of frame color and background coloring.

An emergency light can optionally be associated with a display, which generally depends on the chosen display. A label can be added on an emergency light.

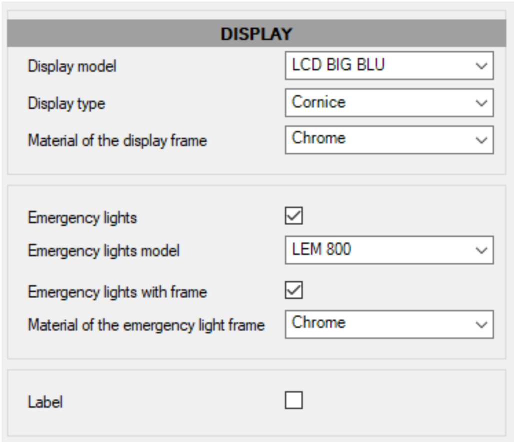


FIGURE 5.12: The wizard part related to the display

5.3.3 Logo

The logo is a label containing a text chosen by the customer that can be positioned "under display" or "above display" or "under buttons". The customer can define a rectangular area (width / height / position) where the logo will be placed. It can be engraved with a pantograph or laser marked.

The wizard part concerning the logo is represented in the Figure 5.13.

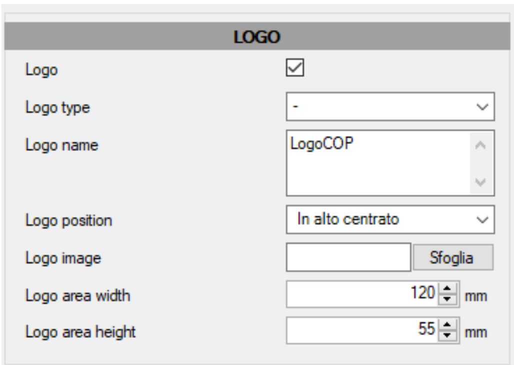


FIGURE 5.13: The wizard part related to the logo

5.3.4 Grids

The grid is a part that visibly shows only holes but behind it contains its electronics and is the point from which the possible sounds emitted by the panel come out. Some grids have a geometry, however even grids without geometry have their own associated dimensions in order to accommodate the electronic part of the component.

5.3.5 Buttons

The buttons are that part of the panel dedicated both to the choice of the floor and to service functions such as alarm, door opener, door closer and call.

To manage this duplicity, the wizard contains two parts dedicated to the buttons: the floor buttons and the service buttons respectively represented in Figure 5.14 and Figure 5.15.

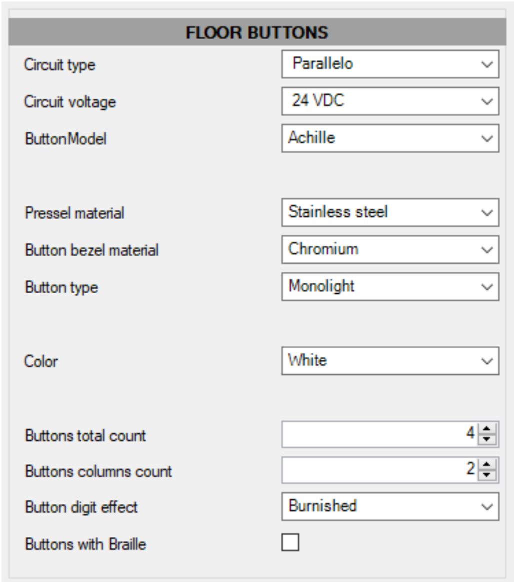


FIGURE 5.14: The wizard part related to the floor buttons

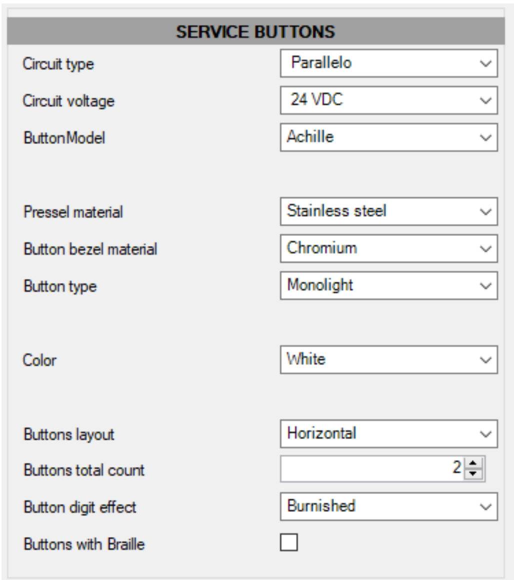


FIGURE 5.15: The wizard part related to the service buttons

All that is chosen for the floor buttons is automatically transferred to the service buttons to ensure uniformity in the choice, however the customer is left free to give different characteristics to the service buttons if he wishes.

For the buttons it is possible to choose the type and voltage of the electrical circuit that compose them for the electronic part, the button model, the various materials and colors for the parts that compose them, the total number of buttons and whether or not they are buttons with braille.

In addition, for the floor buttons the user can also choose how many columns of buttons he wants.

This last feature is the only one that differs for the service buttons, in fact it is not possible to choose the number of columns as these buttons go in a single row, however the user can choose whether to put this row vertically or horizontally.

To enter the digit or the symbol desired by the customer, you must press the button on which you want to set or change writing in the graphical area where the COP is represented and a detail tab will appear dedicated to this choice (Figure 5.16).

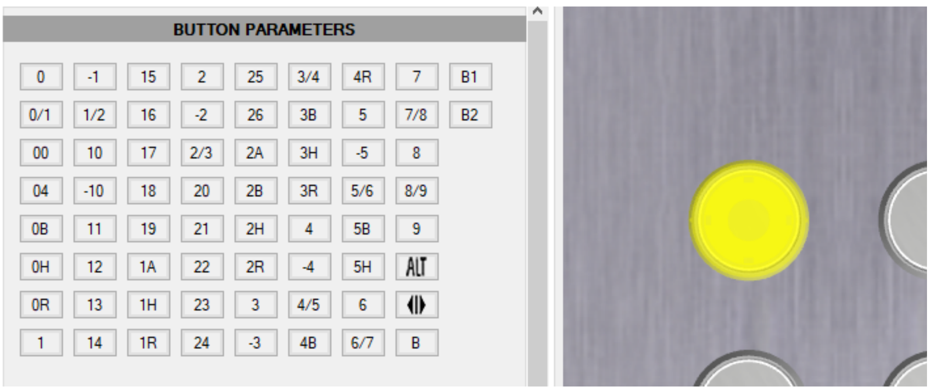


FIGURE 5.16: The details of the button

5.3.6 Cost estimation

The cost estimation for this case study is visualized in the order part and is obtained from the sum of two contributions:

- costs of components;
- cost of machining.

Each component that is standard, like a button or a display, has a pricelist code also obtained form the resolution of some rules. From this code the software is able to search the cost of the component in the pricelist associated to the user.

For the components that are not standard and are obtained from the configuration, like a COP, the final cost of the product is the sum of all the standard codes that are contained in the assembly and the sum of all the operations of machining as laser cuts, welds or mountings in the plate part.

Eq. 5.1 describes the formula that represents the cost related to the cutting process of the plate in order to obtain the necessary space to host the other components.

$$Cost_{CUT} = \frac{\left(l_{EDG} \cdot \frac{1}{v_{EDG}} \cdot HC_{CUT}\right)}{3600} \tag{5.1}$$

The cutting process involves the cutting edge (l_{EDG}), which is related to the geometry of the part to be manufactured, the cutting velocity (v_{EDG}) and the hourly cost (HC_{CUT}) of the cutting process used. Using a similar approach, Eq. 5.2 introduces the calculation of each welding with the welding velocity (v_{WEL}) and the related hourly cost (HC_{WEL}).

$$Cost_{WEL} = \frac{\left(l_{EDG} \cdot \frac{1}{v_{WEL}} \cdot HC_{WEL}\right)}{3600} \tag{5.2}$$

The mounting cost is considered as a function of the components of the plate that are assembled with the plate using this machining operation (5.3). This cost is simply a sum of the components (C_i) respectively multiplied for the specific mounting cost (€/kg) defined for each component (MC_i).

$$Cost_{MOU} = \sum_{i=1}^{N} (C_i \cdot MC_i) \tag{5.3}$$

The operations associated to the COP represents a cost model for this type of products.

5.3.7 Use case customer size

An example of usage of the software from the customer's point of view will be shown now.

The user must first log in at the end of which, if done correctly, the window shown in Figure 5.17 will be opened.

On the left the user will find the tree with his quotations/orders while on the right the products will appear, enabled for the user by the administrator and divided by categories.

FIGURE 5.17: Main window shown after login

At this point, the customer can decide whether to modify a quotation/order not yet sent (the quotations/orders sent are marked in green) or whether to create a new quotation/order.

Suppose that the customer chooses the way to create a new order, to obtain this he must right click on his company name in the tree at the top left and click **New order** (Figure 5.18).

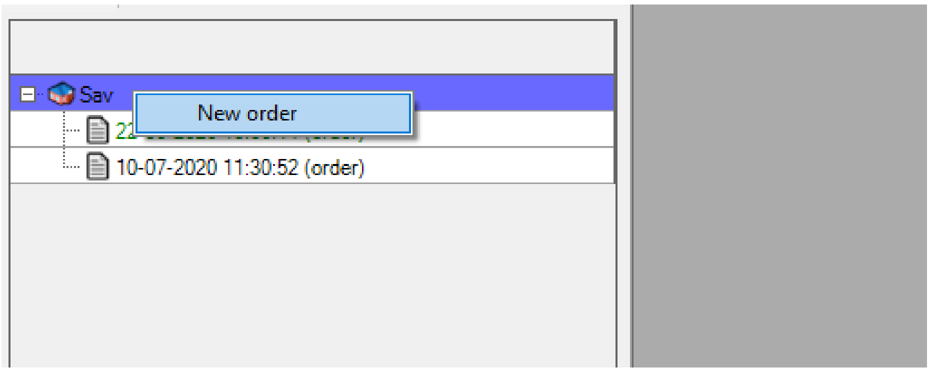


FIGURE 5.18: How to crate a new order

He must then enter all the data necessary to create the new order in the window that appears (Figure 5.19) and press the OK button.

Now the new order has been added to its list of quotations/orders and has been automatically made active, this can be understood from the fact that the name of the order is written in bold type (Figure 5.20). To activate any other quotation/order, he must simply double click on the desired quotation/order.

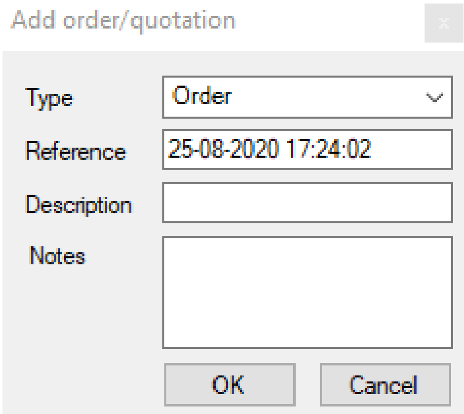


FIGURE 5.19: Order details

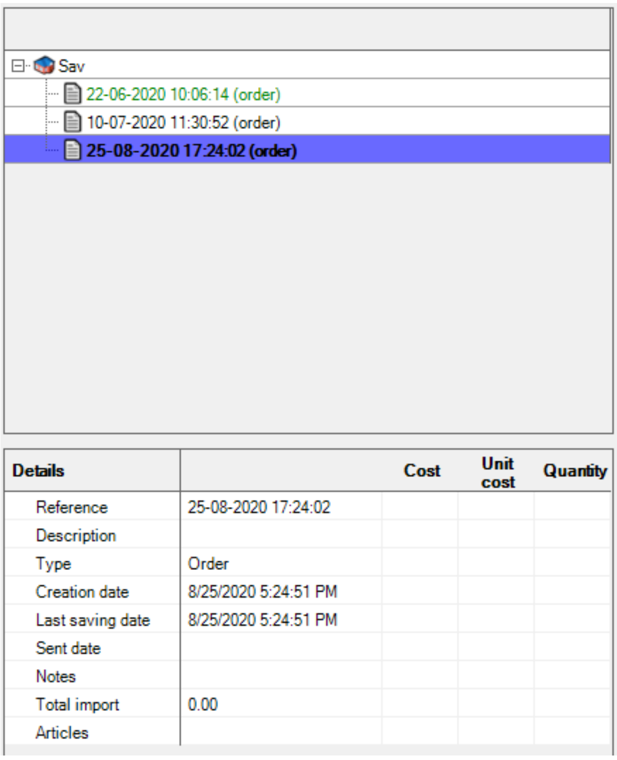


FIGURE 5.20: Main windows after order creation with the new order active

Once the new order has been created, the customer can choose which type of article to add:

- he can configure a new car operating panel (COP) / landing operating panel (LOP) / landing indicator panel (LIP) from scratch;
- he can order a COP / LOP / LIP previously configured by the administrator without the possibility of modifying it, i.e. a template;

- he can start from a COP / LOP / LIP previously configured by him or from a template, duplicate and modify it by reconfiguring it;
- he can order replacement parts of COP / LOP / LIP that can find in the product's tree configuring them to make them suitable for his needs.

The first and third type of articles will now be shown in more detail carrying two example of car operating panel (COP).

To configure a COP from scratch, the user must press on the button with the magic wand in the upper left of the main window (Figure 5.21).

FIGURE 5.21: Button "New wizard"

A window will then appear to decide the beginning configurable product between COP, LOP and LIP and what name the user desires to give to the product(Figure 5.22).

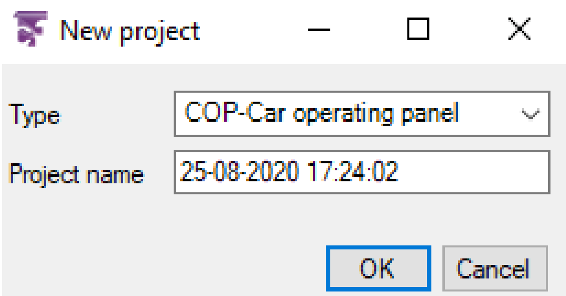


FIGURE 5.22: Details of the new configuration product

Once the OK button is pressed, the COP configuration wizard will automatically compose in the center of the main window as shown in Figure 5.23.

The user wizard is divided into two tab pages. The first tab is called exactly "Wizard" and it is the one that helps the user in the general configuration of the desired article. The second, called "Details", is used to configure the component parts of the COP in a more specific way.

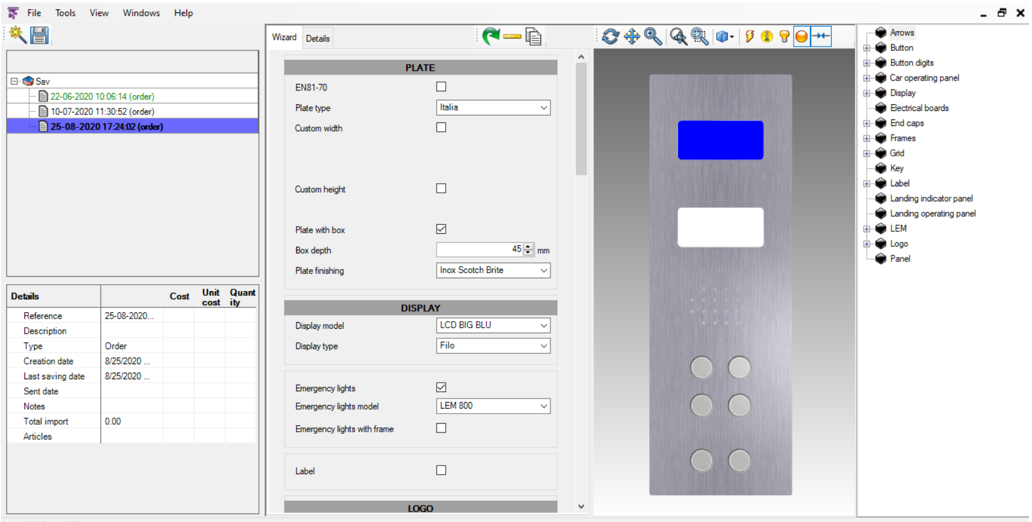


FIGURE 5.23: Main windows after opening the configuration wizard

The configuration flow is designed as follows: first the user must go to the "Wizard" tab and then to the "Details" tab to manage configuration parameter.

The graphical part of the configurator, the viewer, is not automatically updated every time a parameter that affects the COP graphics is changed in the tab "Wizard", this is due to the fact that the creation of the COP geometry takes a long time and would considerably slow down the flow of the configuration. The total updating of the geometry must be manually triggered by the user with the green arrow located at the top right of the "Wizard" tab (Figure 5.24). On the other hand, when the user is changing a parameter that affects the graphics of a component in the "Details" tab, the viewer is instead updated automatically since updating a single component is much faster than updating all the components of the COP simultaneously.

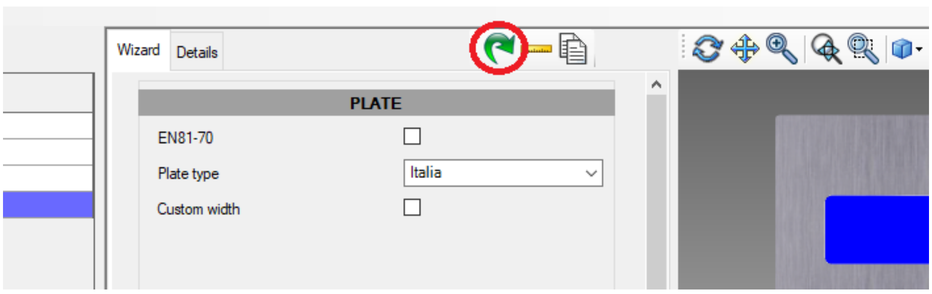


FIGURE 5.24: Button "Redraw panel geometry"

At this point the user can proceed with the configuration. In the "Wizard" tab he will find the parameters for configuring the COP displayed in a logical order and divided by modules that have been explained in the previous subsections. Each parameter has a variable value that is calculated by connecting it to a series of rules that act in background.

Parameters and rules are found in the database and are the result of the overall knowledge formalization work carried out in advance by the Hyperlean developer team, in which I have actively participated, together with the team of the company of lifts' panels involved in the project which was made up of reference figures from both the technical and commercial department.

Each time a parameter is changed in the configuration wizard, the CSP solver, explained in the section **Development approach**, is invoked and then the solver engine, starting from the DSM associated with the problem, built by seeing the interdependencies between parameters within the rules, is able to understand which other parameters are involved in the variation of the initial parameter. The solver then finds the rules associated with all the parameters involved and solves them.

Using this method of resolution, the calculation times are drastically reduced since the solver operates on a limited number of rules, therefore the iterations and evaluations to be made are much less than solving all the rules relating to the COP each time.

Once the user has made his choices in the "Wizard" tab, he can press the green arrow at the top right of the tab to graphically view the resulting configured COP. If the user wants to change something, he is free to do that and when he wants to see the updates he must simply click the green arrow at the top right of the tab again. Once the user is satisfied with the configuration obtained, he can choose the most specific characteristics of the components.

For example, by pressing a button in the viewer, on the left the tab pages will switch automatically from the "Wizard" tab to the "Details" tab (Figure 5.25). At the top of the tab the user can find all the digits available for the chosen button model and he can choose which digit to place above the specified button. Below the panel of the digits other parameters ,that modify the highlighted button, are represented, i.e. the color of the collar.

When all configuration choices are completed, the user, by pressing the "Save" button at the top left of the main window (Figure 5.26), next to the button with the magic wand, goes to save both the product as an article in the order and the product itself in the products tree (Figure 5.27).

The article in the order will be found in the table under the order tree as shown in Figure 5.27. The article has a cost associated. This cost is derived from the cost of the basic products combined with the processing costs of the COP plate. The costs of the basic products are obtained from the pricelist that the administrator has previously associated with the user.

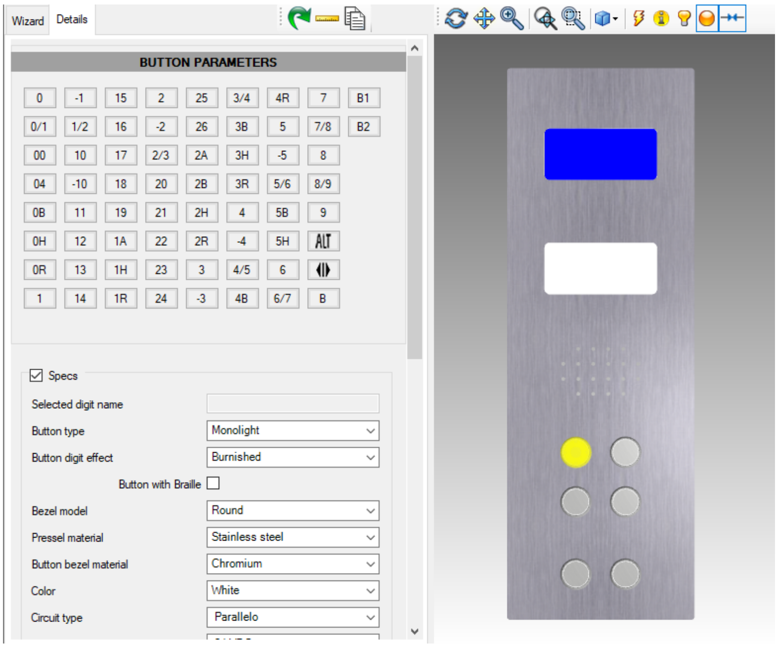


FIGURE 5.25: Buttons' tab details

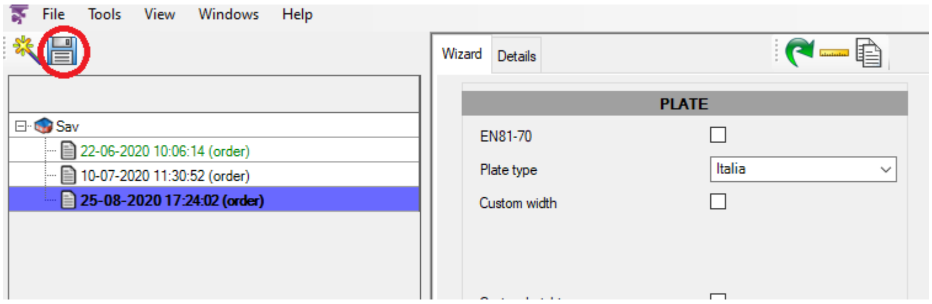


FIGURE 5.26: Button "Save"

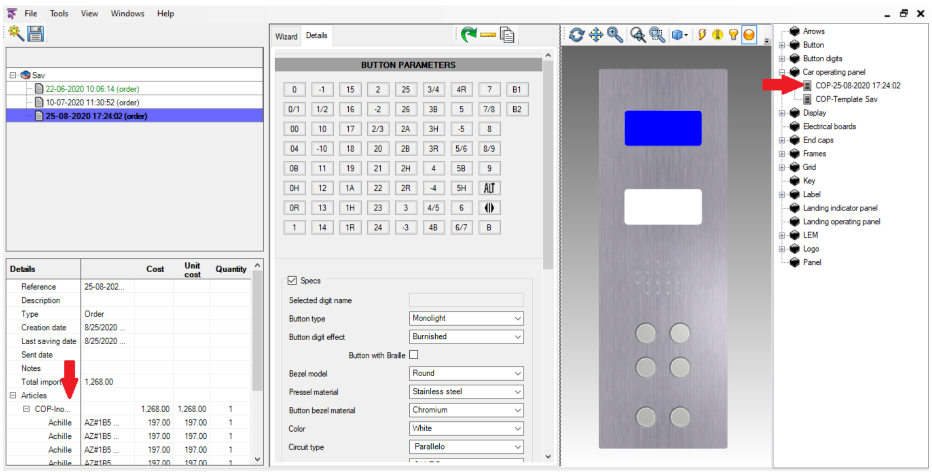


FIGURE 5.27: Where are saved the products: in the order and in the products tree

Once the order is completed, the user, who is part of the client company, can send it to the supplier company in order to be evaluated and can print the report.

The last but not the least potentiality of the software is the workflow of ordering a configured COP starting from a template. Usually the administrator, during the construction of a client area in the software, generates a series of templates products that are generally mostly requested by the customer to ease the order phase. The customer user when opens the software and starts to compile a order can add directly one of his templates, that obviously the administrator make visible to him or he can think that, the template is something that is very similar to his desire but he would like to bring some little change, like a floor button digit or the color of the collar of the service buttons or the color of the frame of the display, gun can instead of silver. This workflow is possible following these steps: he must open the template that is more similar to his desire, the template is disabled since it is a product released from the administrator so not modifiable, this can be seen in 5.27. Then he must press the button "Duplicate" posed on the top right of the tab "Wizard" and he must wait some seconds that the product is duplicated and subsequently opened. Once the duplicated product is ready the customer can modified it in the way he prefers and than save the product as an article in the order (5.29).

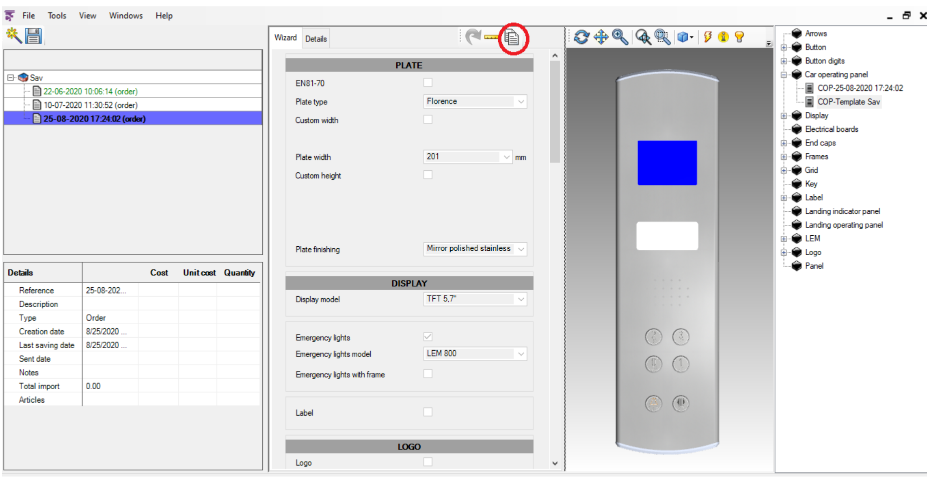


FIGURE 5.28: Graphical interface of a template product and the position of the button "Duplicate"

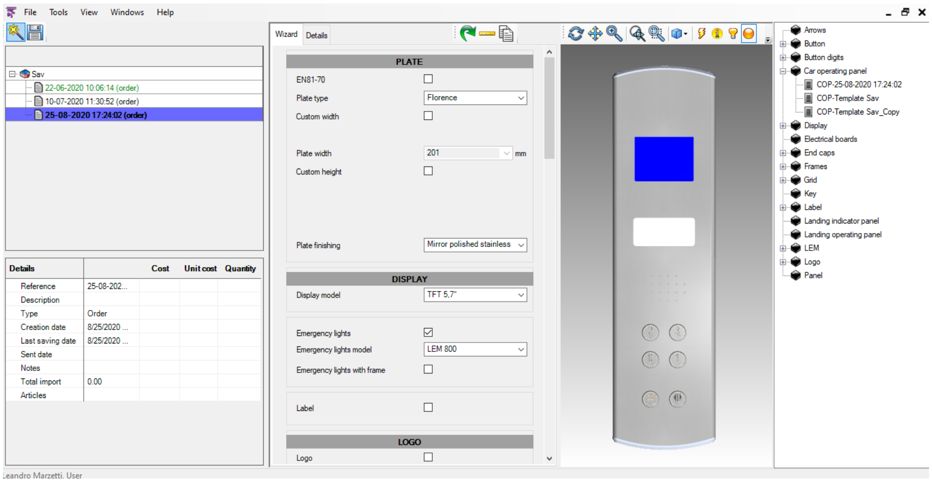


FIGURE 5.29: Product duplicated and saved

5.4 Results

In this Chapter have been presented a framework to support the configuration design from the configurations itself of an ETO product to the management of orders and reports.

The case study proposed to validate the method and the framework concentrates on the configuration of lifts' panels.

The engine of the configuration phase is a solver based on the CSP problem of Chapter 4 combined with a preliminary Design Structured Matrix analysis. This decision has been made to split into sub-problems the original problem to make the configuration phase agile and rapid. The test case of one single lift panel shown in the previous section comprehends a set of about 900 rules related to about 300 parameters. The time to solve this case is around 5 seconds, using this approach. When a parameter is changed in the wizard tab, the refresh of the parameters related is on the order of milliseconds.

Moreover the framework includes the possibility of human intervention of the administrator directly on parameter and rules let this framework to be open and improved directly by the user without further work of the programmer.

Chapter 6

Discussions

The resulting advantages of using knowledge-based methodologies and product platform adoption for rapid configuration of new product variants are well known. Research efforts in this topics are continuously progressing concentrating on the development of theories and methodologies to achieve these goals. Different approaches are presented in literature, such as knowledge-based engineering, configuration, computational design synthesis and modularity. However, the use of this methodologies within the industrial world is often limited, probably due to the high costs that the creation of specialized software entails. In this context, the task to completely automate the procedure for product configuring is disproportionate but also very interesting. However, with the development of improved design technologies that can automatically manage the design information at different levels of detail, important progresses can be made.

The methodology developed in this thesis begins with the company knowledge formalization and representation from the available data within the company, which can be provided by adding different micro-contributes from technicians. The retrieving of the information, which is supported by a strong integration with the company tools, leads to the definition of a product family meta-model and to the building of a product family database. This preparatory phase, which is also the most difficult, is necessary for the implementation of the system within the company and to proceed with the conceptual design and the cost estimation of new products. The results have demonstrated that such a methodology allows a better management of the company data and it makes them reusable for the future jobs.

In order to give a support to the user in the definition of a new product, the knowledge formalization and representation is collected in a database of parameters and rules. The results have shown the benefits of the method in managing complex design problem. In particular, it has been seen that the method and the tools presented above allow the users to face subproblems of reduced size and help them in taking a priori choices. The aim of this methodology is to support the user during the conceptual design stage towards a solution of the design problem.

The approach proposed highlights how the Design Structured Matrix combined with Constraints Satisfaction Problems analysis can support the designer during the conceptual design phase in two aspects of his work: in optimization analysis, to reduce the design space of solutions to be investigated and subsequently optimized and in design configuration, in order to obtain a configured ETO product in an easy and fast way. The calculation velocity of this combination can be used also to develop automatic procedures for the design phase.

The methods for cost estimation are then applied to the solution that has been previously found. The cost estimation methods proposed in this thesis allow a rapid parameter-based estimation, descending from the attributes that have been calculated in the conceptual design stage. The application of the methods has shown a good estimation of the actual cost both for purchased parts and for parts produced within the company.

Finally, the methodology supporting framework is a step toward conceptual design support.

This thesis has presented the result in defining two configuration frameworks for the development of effective configuration tools. The objective of both frameworks is to obtain product variants in a very short time, performing a rendering view of the model from the modular configuration giving a final cost of the configured product, they differs on their declination: one is more concentrated on the design optimization and the other on the design configuration.

Benefits in term of cost savings and developing time reduction can be obtained to gain company competitiveness.

The frameworks are two because the first has been a prototypical software that concentrates on the optimization aspects, developing and improving the solver engine based on the backtracking method; the second software concentrates on configuration aspects and has instead been developed to completely support the design of ETO products, including the solver engine of the first software and managing more industrial scenarios: from the orders and pricelists to the reports.

The final framework is based on a multi-level structure able to handle multiple views of a modular product. It makes reference to functional information transformed in rules, which are then mapped into modules and structured in product architectures. It supports the configuration process providing the information with an appropriate degree of abstraction or detail, on the basis of the specific conceptual design phase.

The frameworks demonstrate the effective strength of multi-level product platform to visually represent and manage complex systems for product variants configuration. This work introduces an innovative method to manage product configuration and costs in a fast and easy way in order to gain a better understanding of change and potential propagation paths. In addition it can represent a database of the product information and then designers can take advantage of this information that normally is time consuming to retrieve because implicit.

Chapter 7

Conclusions

In an engineer-to-order context, configuration and offer stage have a very critical role because the cost estimation must be performed on products that are not already designed. Therefore, the research work concentrates on applying knowledge-based tools to the context above. Through an extensive review of the state of art and an analysis of the industrial context, an approach and two frameworks that support the offer stage for new products have been developed. The approach and the frameworks have been thought to be user friendly at industry level.

Two examples, i.e. gas & oil structures and lift's panels, are presented in the thesis to illustrate how the approach could be used in practice. Several architectures have been implemented using the software tool. The final developed system is able also to manage orders, to organize them in a report, to configure a product and to solve design constraints.

The preliminary results show that the methodology is a valid support to the offer stage of engineer to order ETO products and it can help engineers to choose the correct product configuration among different design alternatives. The solver tools presented for variable assignment within the framework that allow to find design solutions in an automatic manner is fast, valid and gives reliable results. Moreover, the use of computational design synthesis tools supports the exploration of new product architectures.

On the other hand, research and development work still remains to improve the system. Future research can be oriented to generalize the framework as much as possible, in particular for the generation of the Graphical User Interface, to further reduce the development time of the software customization which, as was well specified in the Chapter 5, has been considerably reduced but still requires a considerable effort. Moreover the experimentation of the framework should be extended to other product families, in order to validate it in different case studies and eventually to discover different company needs.

Bibliography

[1] AISC. *Specification for Structural Steel Buildings (ANSI/AISC 360-10)*. 2010.

[2] Albert Albers, Markus Spadinger, Manuel Serf, Stefan Reichert, Steffen Heldmaier, Micha Schulz, and Nikola Bursac. "Coupling of Computer-Aided Methods: Supporting Product Developer During Embodiment Synthesis". In: *Advances in Structural and Multidisciplinary Optimization*. Springer International Publishing, 2017, pp. 536–548. DOI: 10.1007/978-3-319-67988-4_41. URL: https://doi.org/10.1007%2F978-3-319-67988-4_41.

[3] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[4] Jafar Alzubi, Anand Nayyar, and Akshi Kumar. "Machine learning from theory to algorithms: an overview". In: *Journal of physics: conference series*. Vol. 1142. 1. IOP Publishing. 2018, p. 012012.

[5] Samuel André and Fredrik Elgh. "Modeling of transdisciplinary engineering assets using the design platform approach for improved customization ability". In: *Advanced Engineering Informatics* 38 (2018), pp. 277–290. DOI: 10.1016/j.aei.2018.07.006. URL: https://doi.org/10.1016%2Fj.aei.2018.07.006.

[6] Samuel André, Fredrik Elgh, Joel Johansson, and Roland Stolt. "The design platform – a coherent platform description of heterogeneous design assets for suppliers of highly customised systems". In: *Journal of Engineering Design* 28.10-12 (2017), pp. 599–626. DOI: 10.1080/09544828.2017.1376244. URL: https://doi.org/10.1080%2F09544828.2017.1376244.

[7] *Artificial Intelligence Techniques*. 2016. DOI: 10.1002/9781119175391.part3. URL: https://doi.org/10.1002%2F9781119175391.part3.

[8] ASCE. *Minimum Design Loads for Buildings and Other Structures (ASCE Standard 7-05)*. Jan. 2010. DOI: 10.1061/9780784408094.

[9] Callie Berliner and James A. Brimsom. *Cost Management for Today's Advanced Manufacturing: The CAM-I Conceptual Design*. Harvard Business School Press, Boston, 1988.

[10] Haseeb Bhatti. *Future of Jobs-World Economic Forum-Review*. Jan. 2018.

[11] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[12] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[13] Stefano Bistarelli, Boi Faltings, and Nicoleta Neagu. "Interchangeability with thresholds and degradation factors for Soft CSPs". In: *Annals of Mathematics and Artificial Intelligence* 67.2 (2013), pp. 123–163. DOI: 10.1007/s10472-013-9348-8. URL: https://doi.org/10.1007%2Fs10472-013-9348-8.

[14] Geoffrey Boothroyd. "Design for Manufacture and Assembly: The Boothroyd-Dewhurst Experience". In: *Design for X*. Springer Netherlands, 1996, pp. 19–40. DOI: 10.1007/978-94-011-3985-4_2. URL: https://doi.org/10.1007%2F978-94-011-3985-4_2.

[15] Antoine Brière-Côté, Louis Rivest, and Alain Desrochers. "Adaptive generic product structure modelling for design reuse in engineer-to-order products". In: *Computers in Industry* 61.1 (2010), pp. 53–65. DOI: 10.1016/j.compind.2009.07.005. URL: https://doi.org/10.1016%2Fj.compind.2009.07.005.

[16] Tyson R. Browning. "Applying the design structure matrix to system decomposition and integration problems: a review and new directions". In: *IEEE Transactions on Engineering Management* 48.3 (2001), pp. 292–306. DOI: 10.1109/17.946528. URL: https://doi.org/10.1109%2F17.946528.

[17] Andrei Bulatov, Martin Dyer, Leslie Ann Goldberg, Markus Jalsenius, and David Richerby. "The complexity of weighted Boolean #CSP with mixed signs". In: *Theoretical Computer Science* 410.38-40 (2009), pp. 3949–3961. DOI: 10.1016/j.tcs.2009.06.003. URL: https://doi.org/10.1016%2Fj.tcs.2009.06.003.

[18] Jonathan Cagan, Matthew I. Campbell, Susan Finger, and Tetsuo Tomiyama. "A Framework for Computational Design Synthesis: Model and Applications". In: *Journal of Computing and Information Science in Engineering* 5.3 (2005), pp. 171–181. DOI: 10.1115/1.2013289. URL: https://doi.org/10.1115%2F1.2013289.

[19] Jorge D Camba and Manuel Contero. "Improved Representation of Dependencies in Feature-based Parametric CAD Models using Acyclic Digraphs." In: *GRAPP*. 2015, pp. 16–25.

[20] Matthew I. Campbell and Kristina Shea. "Computational design synthesis". In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 28.3 (2014), pp. 207–208. DOI: 10.1017/s0890060414000171. URL: https://doi.org/10.1017%2Fs0890060414000171.

[21] Catarina Carvalho, Víctor Dalmau, and Andrei Krokhin. "CSP duality and trees of bounded pathwidth". In: *Theoretical Computer Science* 411.34-36 (2010), pp. 3188–3208. DOI: 10.1016/j.tcs.2010.05.016. URL: https://doi.org/10.1016%2Fj.tcs.2010.05.016.

[22] Amaresh Chakrabarti, Kristina Shea, Robert Stone, Jonathan Cagan, Matthew I. Campbell, Noe Vargas Hernandez, and Kristin L. Wood. "Computer-Based Design Synthesis Research: An Overview". In: *Journal of Computing and Information Science in Engineering* 11.2

(2011). DOI: 10.1115/1.3593409. URL: https://doi.org/10.1115%2F1.3593409.

[23] Senthil K. Chandrasegaran, Karthik Ramani, Ram D. Sriram, Imré Horváth, Alain Bernard, Ramy F. Harik, and Wei Gao. "The evolution, challenges, and future of knowledge representation in product design systems". In: *Computer-Aided Design* 45.2 (2013), pp. 204–228. DOI: 10.1016/j.cad.2012.08.006. URL: https://doi.org/10.1016%2Fj.cad.2012.08.006.

[24] Siva R. Chavali, Chiradeep Sen, Gregory M. Mocko, and Joshua D. Summers. "Using Rule Based Design in Engineer to Order Industry: An SME Case Study". In: *Computer-Aided Design and Applications* 5.1-4 (2008), pp. 178–193. DOI: 10.3722/cadaps.2008.178-193. URL: https://doi.org/10.3722%2Fcadaps.2008.178-193.

[25] Peng Chen, Long Wen, Ran Li, and Xinyu Li. "A hybrid backtracking search algorithm for permutation flow-shop scheduling problem minimizing makespan and energy consumption". In: *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE, 2017. DOI: 10.1109/ieem.2017.8290165. URL: https://doi.org/10.1109%2Fieem.2017.8290165.

[26] Paolo Cicconi, Vincenzo Castorani, Michele Germani, Marco Mandolini, and Alessio Vita. "A multi-objective sequential method for manufacturing cost and structural optimization of modular steel towers". In: *Engineering with Computers* (2019). DOI: 10.1007/s00366-019-00709-0. URL: https://doi.org/10.1007%2Fs00366-019-00709-0.

[27] Paolo Cicconi, Michele Germani, Sergio Bondi, Alessandro Zuliani, and Emanuele Cagnacci. "A Design Methodology to Support the Optimization of Steel Structures". In: *Procedia CIRP* 50 (2016), pp. 58–64. DOI: 10.1016/j.procir.2016.05.030. URL: https://doi.org/10.1016%2Fj.procir.2016.05.030.

[28] Paolo Cicconi, Miriam Nardelli, Roberto Raffaeli, and Michele Germani. "Integrating a constraint-based optimization approach into the design of oil & gas structures". In: *Advanced Engineering Informatics* 45 (2020), p. 101129. ISSN: 1474-0346. DOI: https://doi.org/10.1016/j.aei.2020.101129. URL: http://www.sciencedirect.com/science/article/pii/S1474034620301002.

[29] Paolo Cicconi, Roberto Raffaeli, and Michele Germani. "An Approach to Support Model Based Definition by PMI Annotations". In: *Proceedings of CAD16*. CAD Solutions LLC, 2016. DOI: 10.14733/cadconfp.2016.152-156. URL: https://doi.org/10.14733%2Fcadconfp.2016.152-156.

[30] Paolo Cicconi, Roberto Raffaeli, Matteo Marchionne, and Michele Germani. "A Model-Based Simulation Approach to Support the Product Configuration and Optimization of Gas Turbine Ducts". In: *Proceedings of CAD17*. CAD Solutions LLC, 2017. DOI: 10.14733/

cadconfp.2017.258-262. URL: https://doi.org/10.14733%2Fcadconfp.2017.258-262.

[31] Guillaume Ducellier, Pierre-Alain Yvars, and Benoît Eynard. "Managing design change order in a PLM platform using a CSP approach". In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 8.3 (2014), pp. 151–158. DOI: 10.1007/s12008-014-0213-8. URL: https://doi.org/10.1007%2Fs12008-014-0213-8.

[32] P. Duverlie and J. M. Castelain. "Cost Estimation During Design Step: Parametric Method versus Case Based Reasoning Method". In: *The International Journal of Advanced Manufacturing Technology* 15.12 (1999), pp. 895–906. DOI: 10.1007/s001700050147. URL: https://doi.org/10.1007%2Fs001700050147.

[33] Klaus Ehrlenspiel, Alfons Kiewert, and Udo Lindemann. "Factors That Influence Manufacturing Costs and Procedures for Cost Reduction". In: *Cost-Efficient Design*. ASME, 2007, pp. 143–384. DOI: 10.1115/1.802507.ch7. URL: https://doi.org/10.1115%2F1.802507.ch7.

[34] Fredrik Elgh. "Decision support in the quotation process of engineered-to-order products". In: *Advanced Engineering Informatics* 26.1 (2012), pp. 66–79. DOI: 10.1016/j.aei.2011.07.001. URL: https://doi.org/10.1016%2Fj.aei.2011.07.001.

[35] Anna Ericsson, Gunnar Erixon, Society of Manufacturing Engineers, and American Society of Mechanical Engineers. *Controlling Design Variants: Modular Product Platforms*. Society of Manufacturing Engineers, 1999. ISBN: 9780791801505. URL: https://books.google.it/books?id=6ByeNgAACAAJ.

[36] Ronald Farrell and Timothy Simpson. "Product Platform Design to Improve Commonality in Custom Products". In: *Journal of Intelligent Manufacturing* 14 (Dec. 2003), pp. 541–556. DOI: 10.1023/A:1027306704980.

[37] Alexander Felfernig, Lothar Hotz, Claire Bagley, and Juha Tiihonen. *Knowledge-based configuration: From research to business cases*. Newnes, 2014.

[38] Xavier Fischer. "Conception Inversée Intégrée : prédimensionnement de produits par satisfaction de contraintesInverted Integrated Design: constraint satisfaction for design support system". In: *Mécanique & Industries* 3.6 (2002), pp. 593–605. DOI: 10.1016/s1296-2139(02)01204-6. URL: https://doi.org/10.1016%2Fs1296-2139%2802%2901204-6.

[39] Xavier Fischer, Christophe Merlo, Jérémy Legardeur, Laurent Zimmer, and Alexis Anglada. "Knowledge Management and Support Environment in Early Phases of Design Process". In: *Volume 4: 24th Computers and Information in Engineering Conference*. ASME, 2004. DOI: 10.1115/detc2004-57791. URL: https://doi.org/10.1115%2Fdetc2004-57791.

[40] Gerald Frank, Doris Entner, Thorsten Prante, Vaheh Khachatouri, Martin Schwarz, et al. "Towards a generic framework of engineering design automation for creating complex CAD models". In: *International Journal on Advances in Systems and Measurements* 7.1 (2014), pp. 179–192.

[41] Klaas Gadeyne, Gregory Pinte, and Kristof Berx. "Describing the design space of mechanical computational design synthesis problems". In: *Advanced Engineering Informatics* 28.3 (2014), pp. 198–207. DOI: `10.1016/j.aei.2014.03.004`. URL: `https://doi.org/10.1016%2Fj.aei.2014.03.004`.

[42] Shuming Gao, Shuting Zhang, Xiang Chen, and Youdong Yang. "A framework for collaborative top-down assembly design". In: *Computers in Industry* 64.8 (2013), pp. 967–983. DOI: `10.1016/j.compind.2013.05.007`. URL: `https://doi.org/10.1016%2Fj.compind.2013.05.007`.

[43] Saeed Gholizadeh and Amir Baghchevan. "Multi-objective seismic design optimization of steel frames by a chaotic meta-heuristic algorithm". In: *Engineering with Computers* 33.4 (2017), pp. 1045–1060. DOI: `10.1007/s00366-017-0515-0`. URL: `https://doi.org/10.1007%2Fs00366-017-0515-0`.

[44] Alistair G. F. Gibb. "Standardization and pre-assembly- distinguishing myth from reality using case study research". In: *Construction Management and Economics* 19.3 (2001), pp. 307–315. DOI: `10.1080/01446190010020435`. URL: `https://doi.org/10.1080%2F01446190010020435`.

[45] Lewis Gordon. "Product Design for Manufacturing and Assembly." In: *Mechanical Engineers' Handbook [Internet]*. John Wiley and Sons, 2015, pp. 1–19. DOI: `10.1002/9781118985960.meh202`. URL: `http://dx.doi.org/10.1002/9781118985960.meh202`.

[46] Rafael Gouriveau and Daniel Noyes. "Risk management – dependability tools and case-based reasoning integration using the object formalism". In: *Computers in Industry* 55.3 (2004), pp. 255–267. DOI: `10.1016/j.compind.2004.08.003`. URL: `https://doi.org/10.1016%2Fj.compind.2004.08.003`.

[47] Tias Guns, Anton Dries, Guido Tack, Siegfried Nijssen, and Luc De Raedt. "The MiningZinc Framework for Constraint-Based Itemset Mining". In: *2013 IEEE 13th International Conference on Data Mining Workshops*. IEEE, 2013. DOI: `10.1109/icdmw.2013.38`. URL: `https://doi.org/10.1109%2Ficdmw.2013.38`.

[48] Anders Haug, Sara Shafiee, and Lars Hvam. "The costs and benefits of product configuration projects in engineer-to-order companies". In: *Computers in Industry* 105 (2019), pp. 133–142. DOI: `10.1016/j.compind.2018.11.005`. URL: `https://doi.org/10.1016%2Fj.compind.2018.11.005`.

[49] Bergen Helms, Kristina Shea, and Frank Hoisl. "A Framework for Computational Design Synthesis Based on Graph-Grammars and Function-Behavior-Structure". In: *Volume 8: 14th Design for Manufacturing and the Life Cycle Conference 6th Symposium on International Design and Design Education 21st International Conference on Design Theory and Methodology, Parts A and B*. ASME, 2009. DOI: 10.1115/detc2009-86851. URL: https://doi.org/10.1115%2Fdetc2009-86851.

[50] Katja Hölttä-Otto and Kevin Otto. "Platform Concept Evaluation". In: *Product Platform and Product Family Design*. Springer US, 2006, pp. 49–72. DOI: 10.1007/0-387-29197-0_4. URL: https://doi.org/10.1007%2F0-387-29197-0_4.

[51] Katja M. M. Holtta and Mikko P. Salonen. "Comparing Three Different Modularity Methods". In: *Volume 3b: 15th International Conference on Design Theory and Methodology*. ASME, 2003. DOI: 10.1115/detc2003/dtm-48649. URL: https://doi.org/10.1115%2Fdetc2003%2Fdtm-48649.

[52] Joel Johansson, Manuel Contero, Pedro Company, and Fredrik Elgh. "Supporting connectivism in knowledge based engineering with graph theory, filtering techniques and model quality assurance". In: *Advanced Engineering Informatics* 38 (2018), pp. 252–263. DOI: 10.1016/j.aei.2018.07.005. URL: https://doi.org/10.1016%2Fj.aei.2018.07.005.

[53] Peter Jonsson and Johan Thapper. "Tractability conditions for numeric CSPs". In: *Theoretical Computer Science* 715 (2018), pp. 21–34. DOI: 10.1016/j.tcs.2018.01.013. URL: https://doi.org/10.1016%2Fj.tcs.2018.01.013.

[54] Henning Kagermann, Wolf-Dieter Lukas, and Wolfgang Wahlster. "Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution". In: *VDI nachrichten* 13.1 (2011).

[55] Ron Kohavi and Foster Provost. "Glossary of Terms". In: *Machine Learning* 2 (Jan. 1998), pp. 271–274. DOI: 10.1023/A:1017181826899.

[56] Janet Kolodner. "What Is Case-Based Reasoning?" In: *Case-Based Reasoning*. Elsevier, 1993, pp. 3–31. DOI: 10.1016/b978-1-55860-237-3.50005-4. URL: https://doi.org/10.1016%2Fb978-1-55860-237-3.50005-4.

[57] Yohanes Kristianto, Petri Helo, and Roger Jianxin Jiao. "A system level product configurator for engineer-to-order supply chains". In: *Computers in Industry* 72 (2015), pp. 82–91. DOI: 10.1016/j.compind.2015.04.004. URL: https://doi.org/10.1016%2Fj.compind.2015.04.004.

[58] Vipin Kumar. "Algorithms for constraint-satisfaction problems: A survey". In: *AI magazine* 13.1 (1992), pp. 32–32.

[59] Vipin Kumar and Yow-Jian Lin. "A framework for intelligent backtracking in logic programs". In: *Lecture Notes in Computer Science*.

Springer Berlin Heidelberg, 1986, pp. 108–123. DOI: 10.1007/3-540-17179-7_7. URL: https://doi.org/10.1007%2F3-540-17179-7_7.

[60] Hai Quoc Le, Somjit Arch-int, Huy Xuan Nguyen, and Ngamnij Arch-int. "Association rule hiding in risk management for retail supply chain collaboration". In: *Computers in Industry* 64.7 (2013), pp. 776–784. DOI: 10.1016/j.compind.2013.04.011. URL: https://doi.org/10.1016%2Fj.compind.2013.04.011.

[61] Olivier Lhomme. "Arc-Consistency Filtering Algorithms for Logical Combinations of Constraints". In: *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Springer Berlin Heidelberg, 2004, pp. 209–224. DOI: 10.1007/978-3-540-24664-0_15. URL: https://doi.org/10.1007%2F978-3-540-24664-0_15.

[62] Bingbing Li, Li Ding, Mark Rajai, Di Hu, and Shengzi Zheng. "Backtracking Algorithm-based Disassembly Sequence Planning". In: *Procedia CIRP* 69 (2018), pp. 932–937. DOI: 10.1016/j.procir.2017.12.007. URL: https://doi.org/10.1016%2Fj.procir.2017.12.007.

[63] Lei Lin, Ivana Rasovska, Roland De Guio, and Sébastien Dubois. "Optimization Methods for Inventive Design". In: *TRIZ – The Theory of Inventive Problem Solving*. Springer International Publishing, 2017, pp. 151–185. DOI: 10.1007/978-3-319-56593-4_7. URL: https://doi.org/10.1007%2F978-3-319-56593-4_7.

[64] John McDermott. "R1: A rule-based configurer of computer systems". In: *Artificial Intelligence* 19.1 (1982), pp. 39–88. DOI: 10.1016/0004-3702(82)90021-2. URL: https://doi.org/10.1016%2F0004-3702%2882%2990021-2.

[65] Chris McMahon, Alistair Lowe, and Steve Culley. "Knowledge management in engineering design: personalization and codification". In: *Journal of Engineering Design* 15.4 (2004), pp. 307–325. DOI: 10.1080/09544820410001697154. URL: https://doi.org/10.1080%2F09544820410001697154.

[66] Clemens Münzer. "Constraint-Based Methods for Automated Computational Design Synthesis of Solution Spaces". PhD thesis. Technische Universität München, 2015.

[67] Jesper Møller, Henrik Reif Andersen, and Henrik Hulgaard. "Product configuration over the internet". In: *Proceedings of the 6th INFORMS* (2001).

[68] Douglas C Montgomery. *Design and analysis of experiments*. John wiley & sons, 2017.

[69] David R. Morrison, Sheldon H. Jacobson, Jason J. Sauppe, and Edward C. Sewell. "Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning". In: *Discrete Optimization* 19 (2016), pp. 79–102. DOI: 10.1016/j.disopt.2016.

01.005. URL: https://doi.org/10.1016%2Fj.disopt.2016.01.005.

[70] Jochen Mueller, Kristina Shea, and Chiara Daraio. "Mechanical properties of parts fabricated with inkjet 3D printing through efficient experimental design". In: *Materials & Design* 86 (2015), pp. 902–912. DOI: 10.1016/j.matdes.2015.07.129. URL: https://doi.org/10.1016%2Fj.matdes.2015.07.129.

[71] Miriam Nardelli, Paolo Cicconi, Roberto Raffaeli, and Michele Germani. "Supporting design tasks through constraint satisfaction tools". In: *Proceedings of the DESIGN 2018 15th International Design Conference*. Faculty of Mechanical Engineering and Naval Architecture University of Zagreb Croatia The Design Society Glasgow UK, 2018. DOI: 10.21278/idc.2018.0499. URL: https://doi.org/10.21278%2Fidc.2018.0499.

[72] Michael Negnevitsky. *Artificial Intelligence: A Guide to Intelligent Systems*. Addison-Wesley, 2005. ISBN: 9780321204660. URL: https://books.google.it/books?id=1BxYQnrfv9MC.

[73] Gregory Norris. "Integrating life cycle cost analysis and LCA". In: *International Journal of Life Cycle Assessment* 6 (Mar. 2001), pp. 118–120. DOI: 10.1007/BF02977849.

[74] P Russel Norvig. *A modern approach*. Prentice Hall Upper Saddle River, NJ, USA: 2002.

[75] Eliab Z. Opiyo. "A feature-based approach to conceptualisation, upfront modelling, and planning for the future of complex systems". In: *International Journal of Information Technology and Management* 16.1 (2017), p. 91. DOI: 10.1504/ijitm.2017.10001055. URL: https://doi.org/10.1504%2Fijitm.2017.10001055.

[76] Klas Orsvärn and Morten H. Bennick. "Tacton: Use of Tacton Configurator at FLSmidth". In: *Knowledge-Based Configuration*. Elsevier, 2014, pp. 211–218. DOI: 10.1016/b978-0-12-415817-7.00017-7. URL: https://doi.org/10.1016%2Fb978-0-12-415817-7.00017-7.

[77] Kevin Otto, Katja Hölttä-Otto, Timothy W. Simpson, Dieter Krause, Sebastian Ripperda, and Seung Ki Moon. "Global Views on Modular Design Research: Linking Alternative Methods to Support Modular Product Family Concept Development". In: *Journal of Mechanical Design* 138.7 (2016). DOI: 10.1115/1.4033654. URL: https://doi.org/10.1115%2F1.4033654.

[78] Roderick Owen and Imre Horvath. "Towards product-related knowledge asset warehousing in enterprises". In: *Proceedings of the Fourth International Symposium on Tools and Methods of Competitive Engineering* (May 2002).

[79] Gerhard Pahl, Wolfgang Beitz, Jörg Feldhusen, and Karl-Heinrich Grote. *Engineering Design*. Springer London, 2007. DOI: 10.1007/

978-1-84628-319-2. URL: https://doi.org/10.1007%2F978-1-84628-319-2.

[80] Doru Panescu and Carlos Pascal. “Holonic coordination obtained by joining the contract net protocol with constraint satisfaction”. In: *Computers in Industry* 81 (2016), pp. 36–46. DOI: 10.1016/j.compind.2015.08.010. URL: https://doi.org/10.1016%2Fj.compind.2015.08.010.

[81] Margherita Peruzzini, Mauro Pistacchi, Andrea Savoretti, Marco Mandolini, and Roberto Raffaeli. “A dependencies satisfaction tool to support the cost oriented configuration of industrial products in the conceptual design stage”. In: *Computer-Aided Design and Applications* 15.6 (2018), pp. 819–830. DOI: 10.1080/16864360.2018.1462565. URL: https://doi.org/10.1080%2F16864360.2018.1462565.

[82] Yvars Pierre-Alain. “Pareto bi-criterion optimization for system sizing : A deterministic and constraint based approach”. In: *ICED 11 - 18th International Conference on Engineering Design - Impacting Society Through Engineering Design* 4 (Jan. 2011), pp. 420–429.

[83] Thomas U. Pimmler and Steven D. Eppinger. “Integration analysis of product decompositions”. In: *ASME Conference on Design Theory and Methodology. Minneapolis, MN*. 1994, pp. 343–351.

[84] Paul Pitiot, Michel Aldanondo, and Elise Vareilles. “Concurrent product configuration and process planning: Some optimization experimental results”. In: *Computers in Industry* 65.4 (2014), pp. 610–621. DOI: 10.1016/j.compind.2014.01.012. URL: https://doi.org/10.1016%2Fj.compind.2014.01.012.

[85] Charles Prud’homme, Jean-Guillaume Fages, and Xavier Lorca. *Choco Documentation*. TASC - LS2N CNRS UMR 6241, COSLING S.A.S. 2017. URL: http://www.choco-solver.org.

[86] Roberto Raffaeli, Paolo Cicconi, Maura Mengoni, and Michele Germani. “Modular Product Configuration: An Automatic Tool for Eliciting Design Knowledge From Parametric CAD Models”. In: *Volume 1: 36th Design Automation Conference, Parts A and B*. ASME, 2010. DOI: 10.1115/detc2010-28242. URL: https://doi.org/10.1115%2Fdetc2010-28242.

[87] Roberto Raffaeli, Maura Mengoni, and Michele Germani. “Improving the link between computer-assisted design and configuration tools for the design of mechanical products”. In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 27.1 (2013), pp. 51–64. DOI: 10.1017/s0890060412000388. URL: https://doi.org/10.1017%2Fs0890060412000388.

[88] Roberto Raffaeli, Andrea Savoretti, and Michele Germani. “Design knowledge formalization to shorten the time to generate offers for Engineer To Order products”. In: *Lecture Notes in Mechanical Engineering*. Springer International Publishing, 2016, pp. 1107–1114.

DOI: 10.1007/978-3-319-45781-9_110. URL: https://doi.org/10.1007%2F978-3-319-45781-9_110.

[89] Eugen Rigger and Thomas Vosgien. "Design automation state of practice-potential and opportunities". In: *Proceedings of the DESIGN 2018 15th International Design Conference*. Faculty of Mechanical Engineering and Naval Architecture University of Zagreb Croatia The Design Society Glasgow UK, 2018. DOI: 10.21278/idc.2018.0537. URL: https://doi.org/10.21278%2Fidc.2018.0537.

[90] Sebastian Ripperda and Dieter Krause. "Cost Effects of Modular Product Family Structures: Methods and Quantification of Impacts to Support Decision Making". In: *Journal of Mechanical Design* 139.2 (2017). DOI: 10.1115/1.4035430. URL: https://doi.org/10.1115%2F1.4035430.

[91] Gianfranco La Rocca. "Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design". In: *Advanced Engineering Informatics* 26.2 (2012), pp. 159–179. DOI: 10.1016/j.aei.2012.02.002. URL: https://doi.org/10.1016%2Fj.aei.2012.02.002.

[92] Gianfranco La Rocca, Lars Krakers, and Michel van Tooren. "Development of an ICAD Generative Model for Blended Wing-Body Aircraft Design". In: *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. American Institute of Aeronautics and Astronautics, 2002. DOI: 10.2514/6.2002-5447. URL: https://doi.org/10.2514%2F6.2002-5447.

[93] Daniel Sabin and Rainer Weigel. "Product configuration frameworks-a survey". In: *IEEE Intelligent Systems* 13.4 (1998), pp. 42–49. DOI: 10.1109/5254.708432. URL: https://doi.org/10.1109%2F5254.708432.

[94] Arthur L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229. DOI: 10.1147/rd.33.0210.

[95] Peter Sanders, Kurt Mehlhorn, Martin Dietzfelbinger, and Roman Dementiev. *Sequential and Parallel Algorithms and Data Structures*. Springer International Publishing, 2019. DOI: 10.1007/978-3-030-25209-0. URL: https://doi.org/10.1007%2F978-3-030-25209-0.

[96] Andrea Savoretti, Marco Mandolini, Roberto Raffaeli, and Michele Germani. "Analysis of the Requirements of an Early Life-cycle Cost Estimation Tool: An Industrial Survey". In: *Procedia Manufacturing* 11 (2017). 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy, pp. 1675 –1683. ISSN: 2351-9789. DOI: https://doi.org/10.1016/j.promfg.2017.07.291. URL: http://www.sciencedirect.com/science/article/pii/S2351978917304997.

[97] Christian Schulte and Guido Tack. "Views and Iterators for Generic Constraint Implementations". In: *Principles and Practice of Constraint Programming - CP 2005*. Springer Berlin Heidelberg, 2005, pp. 817–821. DOI: 10.1007/11564751_71. URL: https://doi.org/10.1007%2F11564751_71.

[98] Gaetano Scirè. "Industria 4.0, la via italiana per la competitività delle PMI". PhD thesis. July 2017.

[99] Stefan Sievers, Tim Seifert, Marcel Franzen, Gerhard Schembecker, and Christian Bramsiepe. "Fixed capital investment estimation for modular production plants". In: *Chemical Engineering Science* 158 (2017), pp. 395–410. DOI: 10.1016/j.ces.2016.09.029. URL: https://doi.org/10.1016%2Fj.ces.2016.09.029.

[100] Elliot Soloway, Judy Bachant, and Keith Jensen. "Assessing the Maintainability of XCON-in-RIME: Coping with the Problems of a VERY Large Rule-Base". In: *Proceedings of the Sixth National Conference on Artificial Intelligence - Volume 2*. AAAI'87. Seattle, Washington: AAAI Press, 1987, 824–829. ISBN: 0934613427.

[101] Georg Steinbichler. "Methods and procedures for optimizing the development of parts for injection-molding production processes". PhD thesis. University Erlangen-Nuernberg, 2008.

[102] Melody Stokes. *Managing engineering knowledge: MOKA: methodology for knowledge based engineering applications*. Vol. 3. 3. Professional Engineering Publishing London, 2001.

[103] Robert B. Stone, Kristin L. Wood, and Richard H. Crawford. "A heuristic method for identifying modules for product architectures". In: *Design Studies* 21.1 (2000), pp. 5–31. DOI: 10.1016/s0142-694x(99)00003-4. URL: https://doi.org/10.1016%2Fs0142-694x%2899%2900003-4.

[104] Jo W. Strandhagen, Logan R. Vallandingham, Erlend Alfnes, and Jan Ola Strandhagen. "Operationalizing lean principles for lead time reduction in engineer-to-order (ETO) operations: A case study". In: *IFAC-PapersOnLine* 51.11 (2018), pp. 128–133. DOI: 10.1016/j.ifacol.2018.08.246. URL: https://doi.org/10.1016%2Fj.ifacol.2018.08.246.

[105] Joshua David Summers and David Rosen. "Mechanical engineering modelling language (MEML): Requirements for conceptual design". In: *DS 75-6: Proceedings of the 19th International Conference on Engineering Design (ICED13), Design for Harmonies, Vol. 6: Design Information and Knowledge, Seoul, Korea, 19-22.08. 2013*. 2013.

[106] Abdourahim Sylla, Delphine Guillon, Elise Vareilles, Michel Aldanondo, Thierry Coudert, and Laurent Geneste. "Configuration knowledge modeling: How to extend configuration from assemble/make to order towards engineer to order for the bidding process". In: *Computers in Industry* 99 (2018), pp. 29–41. DOI: 10.1016/

j.compind.2018.03.019. URL: https://doi.org/10.1016%2Fj.compind.2018.03.019.

[107] Masayuki Tanabe and Atsumi Miyake. "Safety design approach for onshore modularized LNG liquefaction plant". In: *Journal of Loss Prevention in the Process Industries* 23.4 (2010), pp. 507–514. DOI: 10.1016/j.jlp.2010.04.004. URL: https://doi.org/10.1016%2Fj.jlp.2010.04.004.

[108] Ronnie E. Thebeau. "Knowledge management of system interface and interactions for product development processes." PhD Thesis. Massachusetts Institute of Technology, 2001.

[109] Juha Tiihonen and Alexander Felfernig. "An introduction to personalization and mass customization". In: *Journal of Intelligent Information Systems* 49.1 (2017), pp. 1–7. DOI: 10.1007/s10844-017-0465-4. URL: https://doi.org/10.1007%2Fs10844-017-0465-4.

[110] Urli Tommaso, Ceschia Sara, Schaerf Andrea, and Di Gaspero Luca. In: *A General Local Search Solver for FlatZinc*. June 2015.

[111] Hassen Trabelsi, Pierre-Alain Yvars, Jamel Louati, and Mohamed Haddar. "Interval computation and constraint propagation for the optimal design of a compression spring for a linear vehicle suspension system". In: *Mechanism and Machine Theory* 84 (2015), pp. 67–89. DOI: 10.1016/j.mechmachtheory.2014.09.013. URL: https://doi.org/10.1016%2Fj.mechmachtheory.2014.09.013.

[112] Ryuhei Uehara. "Backtracking". In: *First Course in Algorithms Through Puzzles*. Springer Singapore, 2018, pp. 111–127. DOI: 10.1007/978-981-13-3188-6_5. URL: https://doi.org/10.1007%2F978-981-13-3188-6_5.

[113] Karl T. Ulrich, Karen Tung, and Sloan School of Management. *Fundamentals of Product Modularity*. Working paper (Sloan School of Management). Sloan School of Management, Massachusetts Institute of Technology, 1991. URL: https://books.google.it/books?id=Lp95YgEACAAJ.

[114] K.T. Ulrich and S.D. Eppinger. *Product Design and Development*. McGraw-Hill/Irwin, 2012. ISBN: 9780071086950. URL: https://books.google.it/books?id=-eH-ewEACAAJ.

[115] Wim J.C. Verhagen, Pablo Bermell-Garcia, Reinier E.C. van Dijk, and Richard Curran. "A critical review of Knowledge-Based Engineering: An identification of research challenges". In: *Advanced Engineering Informatics* 26.1 (2012), pp. 5–15. DOI: 10.1016/j.aei.2011.06.004. URL: https://doi.org/10.1016%2Fj.aei.2011.06.004.

[116] Xuan-Ha Vu, Djamila Sam-Haroud, and Boi Faltings. "Enhancing numerical constraint propagation using multiple inclusion representations". In: *Annals of Mathematics and Artificial Intelligence* 55.3-4 (2009), pp. 295–354. DOI: 10.1007/s10472-009-9129-6. URL: https://doi.org/10.1007%2Fs10472-009-9129-6.

[117] Lihui Wang, Weiming Shen, Helen Xie, Joseph Neelamkavil, and Ajit Pardasani. "Collaborative conceptual design—state of the art and future trends". In: *Computer-Aided Design* 34.13 (2002), pp. 981–996. DOI: 10.1016/s0010-4485(01)00157-9. URL: https://doi.org/10.1016%2Fs0010-4485%2801%2900157-9.

[118] Ru Wang, Anand Balu Nellippallil, Guoxin Wang, Yan Yan, Janet K. Allen, and Farrokh Mistree. "Systematic design space exploration using a template-based ontological method". In: *Advanced Engineering Informatics* 36 (2018), pp. 163–177. DOI: 10.1016/j.aei.2018.03.006. URL: https://doi.org/10.1016%2Fj.aei.2018.03.006.

[119] John N. Warfield. "Binary Matrices in System Modeling". In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-3.5 (1973), pp. 441–449. DOI: 10.1109/tsmc.1973.4309270. URL: https://doi.org/10.1109%2Ftsmc.1973.4309270.

[120] Leo S. Wierda. "Linking Design, Process Planning and Cost Information by Feature-based Modelling". In: *Journal of Engineering Design* 2.1 (1991), pp. 3–19. DOI: 10.1080/09544829108901667. URL: https://doi.org/10.1080%2F09544829108901667.

[121] John Wiley. "Modern Design and Optimization". In: *Multidisciplinary Design Optimization Supported by Knowledge Based Engineering*. John Wiley & Sons, Ltd, 2015, pp. 10–26. DOI: 10.1002/9781118897072.ch2. URL: https://doi.org/10.1002%2F9781118897072.ch2.

[122] Olga Willner, Jonathan Gosling, and Paul Schönsleben. "Establishing a maturity model for design automation in sales-delivery processes of ETO products". In: *Computers in Industry* 82 (2016), pp. 57–68. DOI: 10.1016/j.compind.2016.05.003. URL: https://doi.org/10.1016%2Fj.compind.2016.05.003.

[123] M. Windheim, E. Greve, and D. Krause. "Decisive economies and opportunity cost of modular product structure alternatives: An empirical case study". In: *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE, 2017. DOI: 10.1109/ieem.2017.8290170. URL: https://doi.org/10.1109%2Fieem.2017.8290170.

[124] Marc Windheim, Nicolas Gebhardt, and Dieter Krause. "Towards a Decision-Making Framework for Multi-Criteria Product Modularization in Cooperative Environments". In: *Procedia CIRP* 70 (2018), pp. 380–385. DOI: 10.1016/j.procir.2018.02.027. URL: https://doi.org/10.1016%2Fj.procir.2018.02.027.

[125] David F. Wyatt, David C. Wynn, Jerome P. Jarrett, and P. John Clarkson. "Supporting product architecture design using computational design synthesis with network structure constraints". In: *Research in Engineering Design* 23.1 (2011), pp. 17–52. DOI: 10.1007/s00163-011-0112-y. URL: https://doi.org/10.1007%2Fs00163-011-0112-y.

[126] Sisi Xuanyuan, Zhaoliang Jiang, Yan Li, and Zhaoqian Li. "Case reuse based product fuzzy configuration". In: *Advanced Engineering Informatics* 25.2 (2011), pp. 193–197. DOI: 10.1016/j.aei.2010.08.002. URL: https://doi.org/10.1016%2Fj.aei.2010.08.002.

[127] Dong Yang and Ming Dong. "A constraint satisfaction approach to resolving product configuration conflicts". In: *Advanced Engineering Informatics* 26.3 (2012), pp. 592–602. DOI: 10.1016/j.aei.2012.03.008. URL: https://doi.org/10.1016%2Fj.aei.2012.03.008.

[128] Jin Yi, Chao Lu, and Guomin Li. "A literature review on latest developments of Harmony Search and its applications to intelligent manufacturing". In: *Mathematical Biosciences and Engineering* 16.4 (2019), pp. 2086–2117. DOI: 10.3934/mbe.2019102. URL: https://doi.org/10.3934%2Fmbe.2019102.

[129] Zaoxu Zhu, Gianfranco La Rocca, and Michael J. L. van Tooren. "A methodology to enable automatic 3D routing of aircraft Electrical Wiring Interconnection System". In: *CEAS Aeronautical Journal* 8.2 (2017), pp. 287–302. DOI: 10.1007/s13272-017-0238-3. URL: https://doi.org/10.1007%2Fs13272-017-0238-3.