Titre: Title:	Development of a Flexible Multiobjective Layout Design Tool for the Design of Mechatronic Systems with a Space Miniature Greenhouse Case Study
Auteur: Author:	Yann-Seing Law-Kam Cio
Date:	2021
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Law-Kam Cio, Y.-S. (2021). Development of a Flexible Multiobjective Layout Design Tool for the Design of Mechatronic Systems with a Space Miniature Greenhouse Case Study [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/6335/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/6335/
Directeurs de recherche: Advisors:	Sofiane Achiche, Giovanni Beltrame, & Aurelian Vadean
Programme: Program:	PhD.

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

DEVELOPMENT OF A FLEXIBLE MULTIOBJECTIVE LAYOUT DESIGN TOOL FOR THE DESIGN OF MECHATRONIC SYSTEMS WITH A SPACE MINIATURE GREENHOUSE CASE STUDY

YANN-SEING LAW-KAM CIO

Département de génie mécanique

Thèse présentée en vue de l'obtention du diplôme de *Philosophiae Doctor*

Génie mécanique

Mai 2021

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Cette thèse intitulée:

DEVELOPMENT OF A FLEXIBLE MULTIOBJECTIVE LAYOUT DESIGN TOOL FOR THE DESIGN OF MECHATRONIC SYSTEMS WITH A SPACE MINIATURE GREENHOUSE CASE STUDY

présenté par **Yann-Seing LAW-KAM CIO**

en vue de l'obtention du diplôme de *Philosophiae Doctor*

a été dûment acceptée par le jury d'examen constitué de :

Maxime RAISON, président

Sofiane ACHICHE, membre et directeur de recherche

Giovanni BELTRAME, membre et codirecteur de recherche

Aurelian VADEAN, membre et codirecteur de recherche

Abolfazl MOHEBBI, membre

Yong ZENG, membre externe

DEDICATION

I dedicate this thesis work to my mother, my father, and my brother.

ACKNOWLEDGEMENTS

I would like to express my gratitude towards my supervisors for their support and trust in my capabilities of realizing the work related to my research project.

For 6 years, my supervisor, Prof. Sofiane Achiche, has been accompanying and guiding me through the journey of post-graduate studies from Master’s to Ph.D. Not only did he teach me how to be a researcher, but he also came up with plenty of ideas and suggestions to give a direction to my career. I am happy to work with Prof. Sofiane Achiche and look forward to keeping this great complicity going in the next steps of my career.

My co-supervisor, Prof. Giovanni Beltrame, made my integration into the SpaceBean project with the technical society PolyOrbite smoother. He also made sure I had everything that I needed either access to the laboratory, equipment for prototyping, or financial support to accomplish my work.

My co-supervisor, Prof. Aurelian Vadean, enriched my work and knowledge with his expertise through his advice and feedback. He also trusted me enough to be part of another research project in collaboration with the technical society Poly-Grames and Professor Jean-Jacques Laurin.

I would like to thank Prof. Jean-Jacques Laurin and Mario Jolicoeur for giving me the chance to be part of the research work.

I would like to express my appreciation to the FRQNT and the Canadian Space Agency for their financial support through scholarships. I would also like to thank Polytechnique de Montréal for their financial support as well as for their workspaces.

I would like to thank my colleagues from the different laboratories and research groups. First, my colleagues from the Laboratoire de Conception de Systèmes Intelligents et Mécatroniques (CoSIM) of Prof. Sofiane Achiche made the work environment so entertaining and joyful. Second, all my colleagues from PolyOrbite and Poly-Grames welcomed me to be part of their work. I would also like to make a special mention to my colleagues Yuanchao Ma and Christophe Marcel Trouillefou for the countless hours of hard work and joyful moments we shared.

Last but none least, I would like to thank my mother, my father, my brother, and my friends for their moral support and advice throughout this adventure.

RÉSUMÉ

Les systèmes mécatroniques sont fondamentalement complexes à concevoir étant donné le besoin d'intégration de composantes mécaniques, électroniques ainsi que logiciel. Les dépendances entre ces composantes sont à la fois difficiles à modéliser et à prendre en compte tôt dans le processus de conception.

L'objectif de cette thèse est de développer un outil d'aide à la conception assistée par ordinateur afin de supporter les concepteurs à positionner et choisir les composantes pour un produit mécatronique. En d'autres mots, le but est de réaliser le schéma de configuration des composantes d'un produit tôt dans le processus de conception.

Le schéma de configuration suit des lignes directrices définies par l'étude et l'analyse des dépendances entre composantes. Une dépendance est définie comme l'impact/influence d'une composante sur les autres. Une dépendance positive implique qu'une composante aide une autre composante à accomplir des requis fonctionnels du produit. Tandis qu'une dépendance négative implique qu'une composante empêche une autre composante à accomplir pleinement des requis fonctionnels du produit. En basant l'outil d'aide à la conception sur l'étude et l'analyse des dépendances, il est possible de supporter l'utilisateur à énoncer son problème de positionnement de composantes.

De plus, l'outil permet de traduire le problème de positionnement en un problème d'optimisation en termes d'objectifs et de contraintes. Cette optimisation est donc un exercice combinatoire considérant toutes les positions et tous les choix de composantes possibles. Afin de résoudre cette optimisation, les méthodes d'approximation, plus précisément, les algorithmes évolutifs sont utilisés et adaptés. Afin de tester l'outil d'aide à la conception développé dans cette thèse, le schéma de configuration d'une serre autonome a été utilisé comme étude de cas.

Dans un premier temps, une nouvelle formulation du problème de configuration d'une serre autonome a été développée et traduite en une optimisation à 15 objectifs. Par la suite, ce problème a été résolu en utilisant un algorithme génétique ayant qu'une seule fonction objective en faisant la somme pondérée des 15 fonctions objectives. Ceci a donc permis de démontrer que l'outil proposé permettait de formuler un problème de configuration d'un produit mécatronique adéquatement.

Par la suite, le problème d'optimisation de la configuration de la serre autonome a été adapté afin de permettre l'utilisation d'un algorithme évolutif multiobjectif. En réalisant cette adaptation, nous avons aussi développé une nouvelle méthode de réduction d'objectifs en faisant la somme pondérée d'objectifs par sous-systèmes identifiés à l'aide la méthode de conception axiomatique. Ceci a permis de réduire le nombre d'objectifs de 15 à 5. Ce problème a, par la suite, été résolu grâce à l'algorithme « non-dominated sorting genetic algorithm II » (NSGA-II). Après 15 à 20 minutes de calcul, le processus d'optimisation a proposé 50 configurations possibles d'une serre autonome.

Finalement, le processus de modularisation a été intégré dans l'outil d'aide au design. En effet, durant l'optimisation, les composantes peuvent se combiner, permettant ainsi de former des modules et de réduire le volume occupé par les composantes. La formation du module est faite grâce à la gestion des dépendances entre composantes ainsi qu'à la représentation par matrices utilisant des nombres complexes. Ceci combiné au NSGA-III a permis de réduire le nombre de modules d'une serre autonome de 9 à un minimum de 4.

ABSTRACT

Mechatronic systems are inherently complex to design due to the integration of software, electronic and mechanical components. The dependencies between these components are both complex to model and difficult to consider early in the design phase.

The objective of this Ph.D. is to develop a multi-objective layout design optimization algorithm to synthesize a mechatronic system, by placing its components based on the device's purpose. This Ph.D. mainly focuses on design support during the conceptual and preliminary design phases.

The layout design of a mechatronic design is achieved by following guidelines defined by the management of the dependencies between components. A dependency is defined as the impact of one component on another one. A positive dependency implies that one component helps another to accomplish the functional requirements of the product. While a negative dependency implies that one component hinders another component to fully accomplish functional requirements of the product. Using the management of the dependencies as the core of the design support tool, it is possible to support the user during the problem statement of the layout design.

Furthermore, the tool also translates the problem statement into an optimization problem in terms of objectives and constraints. This optimization is a combinatorial exercise considering all the possible positions and choices of the components. To resolve this optimization, approximation methods, particularly evolutionary algorithms are used and adapted to solve this problem. To test the developed design support tool in this thesis, the layout design of an autonomous greenhouse has been used as a case study.

Firstly, a novel problem statement of the layout design of an autonomous greenhouse has been developed and translated into an optimization problem of 15 objectives. Then, this problem was solved using a single-objective genetic algorithm by doing a weighted sum of the 15 objectives. This demonstrated that the proposed tool is able to formulate the layout design of a mechatronic problem adequately.

Then, the optimization problem of the layout design of an autonomous greenhouse has been adapted to allow the use of multi-objective evolutionary algorithms. During this adaptation, we were able to develop a novel objective reduction method by doing the weighted sum of objectives

by subsystems identified with the aid of the axiomatic design method. This allows reducing the number of objectives from 15 to 5. Then, this problem was solved with the non-dominated sorting genetic algorithm II (NSGA-II). After a 15 to 20 min optimization time, the result yielded 50 possible layout designs of an autonomous greenhouse.

Finally, the modularization process has been integrated into the computer-aided design tool. Indeed, during the optimization phase, the components can be combined to form a module and reduce the volume occupied by the components. The formation of modules is done using the product-related dependencies between components as well as design structure matrices using complex numbers. This along with the NSGA-III allowed reducing the number of modules of an autonomous greenhouse from 9 to a minimum of 4.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

AD	Axiomatic Design
CA	Customer Attributes
CAD	computer-aided design
CAED	computer-aided engineering design
CAM	computer-aided manufacturing
DSM	design structure matrices
DP	Design Parameter
FOV	Field Of View
FR	Functional Requirement
Fu	Function
GA	Genetic Algorithm
LED	Light-Emitting Diode
M	Mean
MaOP	Many-objective Optimization Problem
MDO	Multidisciplinary Design Optimization
NSGA	Non-dominated Sorting Genetic Algorithm
PCB	Printed Circuit Board
Pr	Property
PRDM	Product-Related Dependencies Management
PV	Process Variable

CHAPTER 1 INTRODUCTION

1.1 Context

Product design and development methodologies have been developed by engineers and designers to improve the product design process for enterprises in terms of five criteria: product quality, product cost, development time, development cost, and development capability as mentioned by Ulrich and Eppinger [1]. These methodologies [1, 2] present a workflow of the design process through guidelines and directives. For example, a summary of the methodology proposed by Ulrich and Eppinger in Product Design and Development [1] is presented in Figure 1.1.

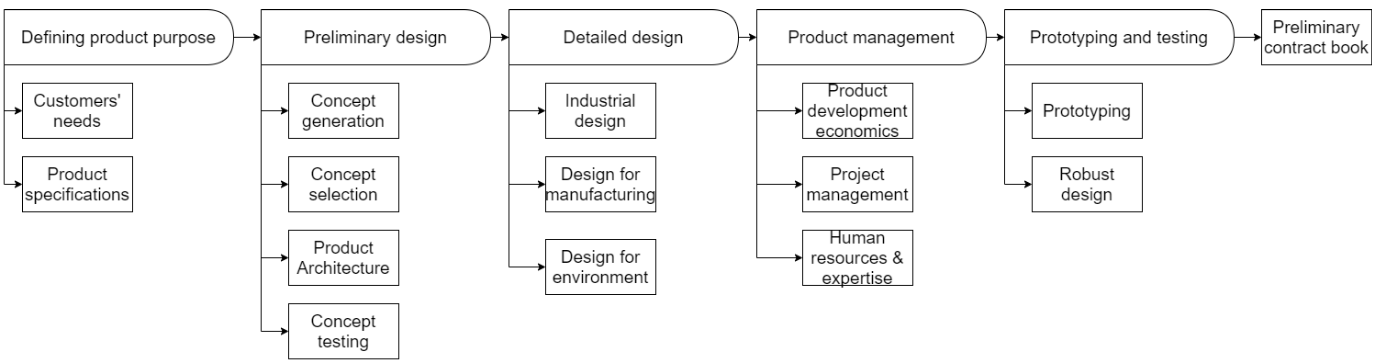


Figure 1.1. Product design and development methodology from Ulrich and Eppinger

In the literature, one could find many generic product design and development methodologies that split the design process in different ways. However, most of them can be sum up by four major steps as reported in Figure 1.2. The first step would be the conceptual design phase where the product specifications are defined based on the customers' needs. Then, concepts of the product are generated and evaluated to keep those that satisfy design requirements and constraints. The second step is the detailed design phase. In this phase, the components and subsystems of the promising concepts defined during the conceptual design phase are being described in terms of their exact geometry and specifications. Furthermore, these concepts are also being tested using simulation tools. The third step is project management where the product feasibility is evaluated in terms of the cost and available resources. The product development timeline is also established in this phase. The last step is to test and build the prototypes and evaluate their performances in terms of the product specifications and customers' needs defined in the first phase. The outcome of this phase will be the final product to be mass-produced and sell.

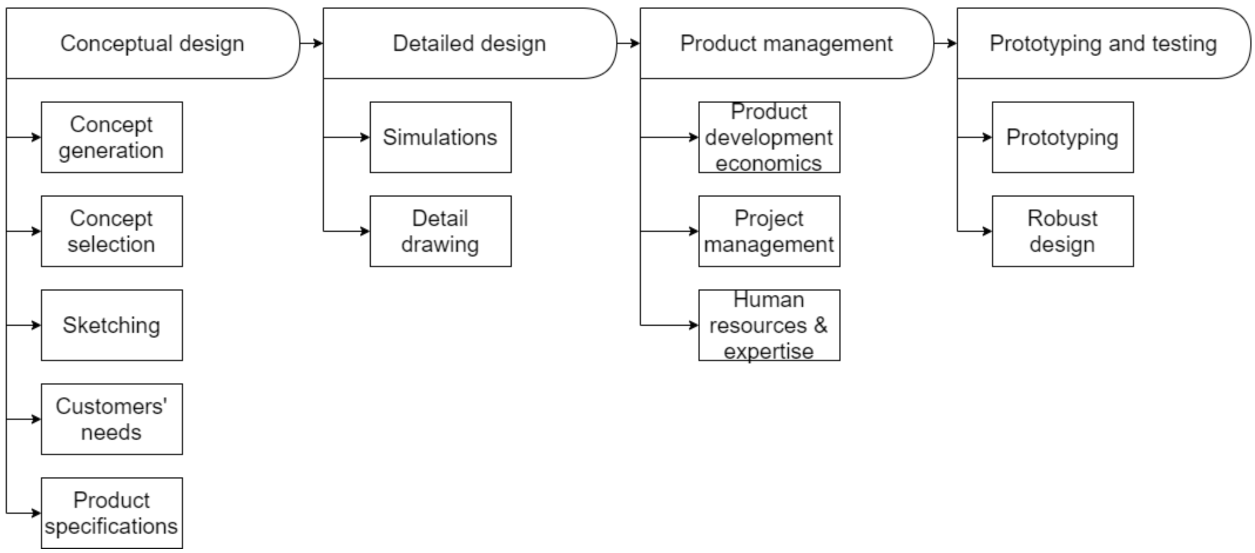


Figure 1.2. General workflow of a product design and development

For every step shown in Figure 1.2, it is possible to identify computer-aided tools. Indeed, the most popular tools are probably computer-aided design (CAD) and computer-aided manufacturing (CAM) such as Catia [3]. These tools are often used for the detailed design phase where the exact geometry and placement of the components are defined by the user. The tool can also automatically optimize the shape of the components for manufacturing in terms of production time and the quantity of raw material needed. Furthermore, in the detailed design phase, there are many tools to run specific simulation tests on the design such as a thermal analysis using a specialized finite element modeling software [4]. The project management phase also has tools to track the general development of a product. For example, Microsoft Project [5] can be used to plan the to-do task and defined a project timeline, thus, supporting the design team through the design process. The prototyping and testing phase can also benefit from statistical tools. For example, JMP [6] offers a software solution to do the design of experiments which can be used to define how robust are the developed prototypes as well as the final product. Finally, the conceptual design phase can take advantage of tools such as SysML[7], HyperStudy [8], or Engineering Drawing Software [9]. However, many engineers perceive these tools as inadequate or incomplete for the conceptual design for many reasons that will briefly be explained in the rest of this chapter.

In 2000, Lipson and Shpitalni [10] reported the importance of conceptual design in which they qualified as the most critical phase of the design process. The conceptual design is a stage where the engineers and designers explore different basic concepts based on function requirements with little information on how to fully achieve the product. They also report that most engineers and designers prefer to do the conceptual design phase through sketches of the components' geometry and functions done by hand instead of using the available CAD/CAM tools. The main reason is the lack of flexibility from the user interface of these tools to sketch concepts when the information on the product is incomplete. This hinders the ability to quickly sketch concepts by starting from scratch or by combining existing ones. 18 years later, Vuletic et al. [11] published a review on the challenges in computer-aided engineering design (CAED) tools for conceptual design. Some of these identified challenges still report that the current CAED tools are ill-adapted for the engineers and the designers. Their review considers most of the works published between 2000 and 2017 that aimed at overcoming the challenges of developing CAED tools for conceptual design. More precisely, this review reports four main issues of the current CAED tools. The first one is as mentioned above, the human-computer interface that is ill-adapted for the conceptual design, especially for quick sketches. The second issue is the difficulty to obtain feedback from customers on the product concepts. Indeed, often, the customer is involved in the product development and the current CAED tools do not allow the customer to indicate the desired modification. The third issue is that by using the current CAED tools, the user is more prone to converge prematurely towards a concept, hence, cutting the conceptual design phase short. The last issue is the learning curve associated with the use of the CAED tools. Users tend to spend an important amount of time learning how to use the CAED tool instead of using the CAED tool for their product development. This review also suggested what should be included in a CAED tool to improve the conceptual design of products, these suggestions are split into three main categories and are reported in Table 1.1.

Table 1.1 Main suggestions to achieve a CAED tool for the conceptual design of a product

Categories	Objective	Main suggestions
#1 Ideation process	Handle the lack of information and make the user interface more intuitive	• Reduce the time it takes to learn how to use the CAED tool • Improve visualization of ideas • Stimulate creativity • Tolerate ambiguity in the sketch
#2 Digitizing of design/ Translation of design	Digitalize sketches and ease the transition from sketches to designing based on expert knowledge.	• Detect redundant task for sketches • Focus more on the sketches and design and less on understanding the interface • Use a system that advises the user during the design process. • Allow for multiple levels of abstraction • Collaborative design
#3 Design review and evaluation	Support the designers and engineers to generate concepts. These concepts need to be transferable to other CAED tools of the following design phases	• Compatible with other CAED for the following design phases • Version control of the designs • Comparison between designs • Mixing and combining designs • Proper representation of the design

		• Exploration of the design space • Support for design evaluation

The rest of the review mainly focuses on comparing the different human-computer interfaces developed for conceptual design for computer-aided sketching. Hence, most of the reviewed papers cover issues related to the first and second categories.

Now that the general portrait of the challenges regarding the conceptual design has been painted, this thesis will give particular attention to parts of the second and third categories. The first category will not be treated since it focuses on manually sketch concepts. Developing sketching tools is not within the scope of this thesis. This research project leans more toward generating and evaluating many concepts in a reasonable amount of time.

Furthermore, it is important to mention that this research project focuses on optimizing mechatronic products. Mechatronic products are inherently complex systems due to the integration of components and subsystems from multiple disciplines such as mechanics, electronics, and computer science. This integration comes with several issues identified by researchers [12, 13] such as managing dependencies between components [14, 15], concepts evaluation of the mechatronic products [16], etc. Additionally, the integration issues for mechatronic products are related to the suggestions in Table 1.1, hence, developing a CAED tool for the conceptual design stage would improve the mechatronic product design process.

For this thesis, the design of an autonomous greenhouse will be used as a case study. An autonomous greenhouse is considered a mechatronic design due to the presence of multiple engineering domains involved, namely: mechanical, electrical, software and control. Indeed, a greenhouse needs to use mechanical engineering methods such as thermal analysis to maintain a uniform temperature. Then, one needs electrical components such as sensors (e.g., heat sensor) and actuators (e.g., water pump) to fulfil the needs of the plants. To adequately integrate these components, a control system must be developed at a low level such as the control of the water pump, or at a high level such as a computer vision system to monitor and ensure the survival and growth of the plant.

In brief, this thesis will focus on obtaining a set of tools to advise and support the designer during the conceptual design phase, to explore the design space, and to compare mechatronic product concepts, and recombine parts of different concepts to generate new ones.

1.2 Thesis organization

In Chapter 2, a relevant review of literature will be carried out to position the research work of this thesis with respect to recent similar works that focus on developing CAED tools for conceptual design. Moreover, a brief comparison of the optimization tools available for the exploration of the design space will be laid out.

In Chapter 3, the main research question and objective of this thesis will be defined.

Then, in Chapter 4, the CAED tools specifications will be dressed. Furthermore, an overview of the theory at the core of the proposed CAED tool will be done to ease the comprehension of the developed tool.

Chapter 5, Chapter 6 & Chapter 7 are composed of published/submitted articles that detail the proposed tool. Chapter 5 starts by reporting and analyzing research works on autonomous greenhouses. Then, the rest of the chapter mainly focuses on the problem statement based on the product-related dependencies modeling as well as the translation of these dependencies into objectives and constraints to formulate an optimization problem. The optimization is done with a single objective genetic algorithm. The objective is an aggregation of all the objectives using the weighted approach. Based on Chapter 5, Chapter 6 aims at solving the optimization problem with a multi-objective optimization algorithm. To achieve this, engineering design tools are used to mitigate the challenges faced when solving a many-objective optimization problem [17]. Hence, an objective reduction approach based on the identification of the sub-systems of the product is used with a multi-objective optimization algorithm. Finally, Chapter 7 adds another layer by allowing modularization during the optimization process. Indeed, modules containing components are formed based on product-related dependencies management between components as well as product performances represented by multiple objective functions. Figure 1.3 summarized the content of Chapter 5, Chapter 6 & Chapter 7.

In Chapter 8, a general discussion about the achievements and limitations of this thesis will be presented. Finally, Chapter 9 will conclude this thesis.

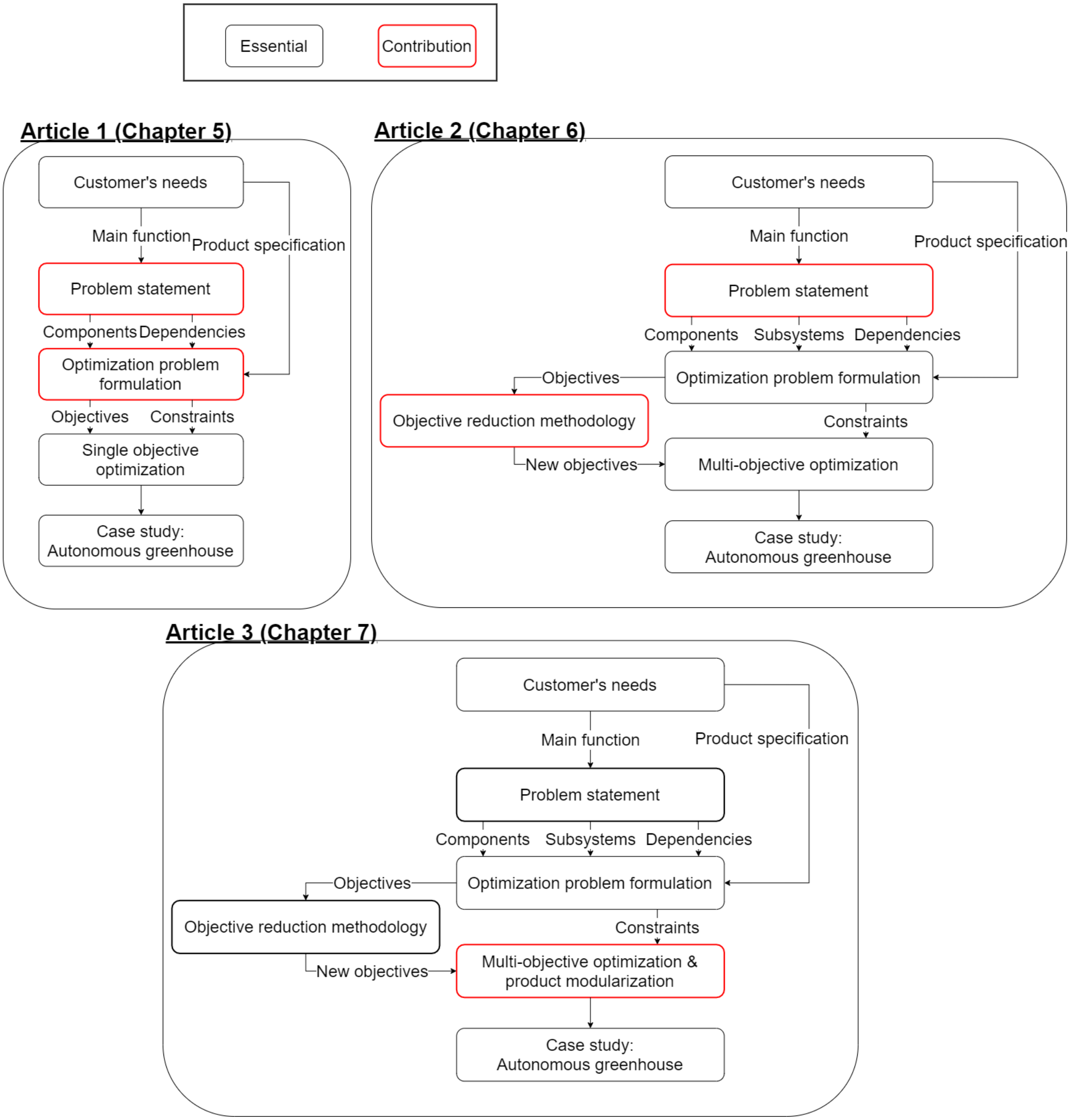


Figure 1.3 Summary of the work and contributions presented in Chapter 5, Chapter 6, and Chapter 7

CHAPTER 2 LITERATURE REVIEW

2.1 Conceptual design support tools

A relevant literature review on the trend in the conceptual design related to the design review and evaluation category of Table 1.1 during 2017 and 2020 will be done. From this starting point, it will be possible to contextualize and defined the research project of this thesis in Chapter 3. Two main approaches will be covered: product-specific methodology and generalized methodology.

The first approach is product-specific which means that the design methodology is developed for a specific product or product family. Chen et al. [18] developed a methodology to optimize the energy consumption of an autonomous underwater vehicle. The energy consumption is mainly affected by the drag which is related to the hydrodynamics as well as the power management and mass distribution of the vehicle. This multidisciplinary design problem was solved using single objective optimization algorithms. Bidoki et al. [19] also developed a methodology to achieve the conceptual design of an autonomous underwater vehicle. To achieve this, a multidisciplinary design optimization (MDO) architecture was combined with an evolutionary algorithm. The chosen MDO architecture was the multidisciplinary feasible since it guarantees a feasible concept. For the optimization of the MDO problem, the particle swarm optimization was chosen. The objective functions were summarized as the target detection probability and the target detection time. The target detection needed to be maximized, and the target detection time needed to be minimized. These objectives functions were aggregated into a single objective optimization using the weighted sum approach. Guo et al. [20] translated the conceptual design of a commercial aero-engine system problem into an optimization problem where the goal was to ensure the performance of the engine as well as environmental friendliness. Then, a comparison was done between two optimization approaches. The first one was a single objective optimization using the weighted sum approach to aggregate the objectives, a set of constraints, and the adaptive simulated annealing algorithm. The second approach was a multi-objective optimization where all the constraints were turned into objectives that were aggregated and then treated as one objective. The algorithm for multi-objective optimization was the non-dominated sorting genetic algorithm II (NSGA-II). Sun et al. [21] developed a two steps MDO method for the preliminary design of integral solid propellant ramjet supersonic cruise vehicles. The optimization was done in terms of the following disciplines needed to design the vehicles: geometry, aerodynamics, radar cross-section, propulsion, mass, and trajectory. The first step was defined as the conceptual design and was done through optimization using a GA with a low level of fidelity. The low level of fidelity means that the evaluation of the disciplines is computed with poor accuracy and low computational time. As opposed to a high level of fidelity which evaluates the disciplines with high accuracy and high computational time. The second step used a compromise between a low and high level of fidelity which is called a medium level of fidelity MDO. This medium level of fidelity, a GA as well as the concepts found during the conceptual design were used to achieve the preliminary design of vehicles. Sgueglia et al. [22] used an adapted version of the FAST algorithm to model hybrid-electric aircraft. The FAST algorithm is an aircraft design tool made for the conceptual design as well as the performance computation of aircraft. Once a model had been done using FAST, it was interfaced with the MDO tool OpenMDAO where the optimization was held. The objectives of the hybrid-electric aircraft conceptual design optimization were energy consumption and weight. The optimization was done using single objective optimization algorithms such as SNOPT as well as multi-objective algorithms such as NSGA-II. Werner et al. [23] developed a Python-based MDO methodology for the conceptual design of a generic b-pillar in terms of shape and size. First, the Latin hypercube sampling was used to create geometries, then the feasibility assessment was evaluated in terms of constraints of the product and its packaging. If the geometry was not feasible, it was penalized. If it was feasible, then, the load case analysis on the generic b-pillar was computed with the aid of the SFE CONCEPT. The evaluated geometry was used as the starting population for the NSGA-II algorithm which optimized four objectives related to the mass of the geometry.

In this thesis, the conceptual design phase of mechatronics is covered. Considering the wide variety of mechatronic products, the developed tool cannot consider only one product. Indeed, the tool must be able to aid the conceptual design of most mechatronic products. However, the review of these product-specific methodologies and tools highlights the commonly used tools to generate and evaluate concepts. Indeed, it is possible to see that many of these tools use search and optimization algorithms, more precisely, evolutionary algorithms.

The second approach offers a generalized methodology to design complex systems such as mechatronics.

Lynch et al. [24] developed an ontology-based methodology to reduce the conceptual design space of cyber-physical systems. Based on the semantic sensor network, they integrated the QUDT, Owl-Time, WGS84, and SWEET ontologies to describe the system at different levels of abstraction. Furthermore, the TOPSIS tool was used to evaluate the performance of the concepts and to find the near-optimal concept. However, the methodology does not offer an algorithm to support the design. El Amine et al. [25] developed a methodology for the conceptual design of a product drive by the ability to satisfy the requirements and performance of the concept while considering the uncertainty of the preliminary design phase. The metrics representing the ability to satisfy the requirements, performance, and uncertainty were aggregated with the weighted sum approach using the analytical hierarchy process. For the uncertainty aspect of the design process, the fuzzy logic theory was used to quantify its impact on the design. The methodology has been tested with the design of a concentrating solar power. These two works present a methodology but does not proposed to automatize part of their methodology. Hence the designer must go through the whole methodology by hand.

Chen and Xie [26] offered a methodology to design multidisciplinary products. First, based on the customer's needs, the functional requirements were defined. Then, to achieve the functional requirements, functional units were used and modeled as differential equations. These differential equations were then transformed into a complex-number domain using the Laplace transform. Considering that Laplace transform is used for linear systems if the system was non-linear, it was linearized around an equilibrium point first. Then, the stability of the system was evaluated using the Routh-Hurwitz method. Finally, the connected functional units were found and regrouped into a functional unit chain. By finding these functional unit chains, it was possible to describe functional requirements and predict its output based on a specific input. The workflow of the methodology starts with the user transforming the customer needs into functional requirements, then, the above methodology algorithm [27] will automatically output the optimal objective functional unit chain. Yi et al. [28] proposed to use the mode pursuing sampling (MPS) method to solve MDO. The MPS is popular for the global optimization of the black-box problem. The MPS started with approximating a model based on the objective function space using experimental points sampled uniformly throughout the design variable space. Then, a guiding mechanism was implemented to converge toward an optimum. Hence, iteratively the MPS allowed finding a near-global optimum. This method was used to optimize a bulk carrier conceptual design by aggregating the objectives into a single objective optimization using the weighted sum approach. Hong et al. [29] offered a conceptual framework mainly by combining different design tools. The methodology is based on a combination of the main concepts of axiomatic design, function-behaviour structure, theory of inventive problem solving (TRIZ) and innovative design thinking. The first phase was the concept generation which was done by the designer. Then, the computer evaluated these concepts. The final phase of one iteration was the concept improvement that was done by the designer with the aid of a human-computer interaction. This method has been used to do the conceptual design of an ultra-precision grinding machine. Jelev and Keane [30] made a framework based on the Blackboard model for the MDO in the early stage of a product design. The Blackboard model started with an initial design then sets a range of values for every design parameter. From that point, engineers could then explore the design space by modifying the values of the design parameter and find the preferred design for their respective domains. During this exploration phase, the engineers had access to a database that contained the information related to the preferred designs. This was done iteratively up to a degree of satisfaction based on the design team. They improved the Blackboard model by including a novel local search algorithm called multidisciplinary pattern search to support the engineers during this process. The method has been used on the conceptual design of a UAV wing. Hu et al. [31] proposed a methodology for the product conceptual design phase while focusing on the improvement of the innovative aspect of this phase. The first step was the interpretation of the customer needs which was done by using the two first steps of the quality-function deployment. These two first steps were used to find the design requirements as well as the components characteristics. Based on the design requirements, the design team used the case-based decision theory (CBDT) model and/or the CBDT combine with the TRIZ model to make decisions. The CBDT model was usually used to make general decisions and the CBDT-TRIZ model was used to generate innovative solutions quickly. Furthermore, the image scale method was used as a visualization tool to support the comparison of concepts. This methodology was used to design a garden hand-tool product. Kontogiannis and Savill [32] developed a framework for MDO during the conceptual design of complex systems based on surrogate modeling and multi-fidelity analysis. The surrogate model was done with a modified Kriging model using sampled data from low-fidelity and high-fidelity analysis. This model approximated the system model. During the whole optimization process, a sub-optimization was running in terms of multiple expected improvements which were estimators of the possible improvement of objective functions. This sub-optimization returned a set of Pareto points that were then evaluated by the error model. The sub-optimization routine was realized by their implementation of a multi-objective particle swarm optimization. The authors tested their framework on 1D and 2D test cases, on Rosenbrock function, on Sellar MDO test cases as well as on a typical transonic airfoil design problem. The framework was also tested in terms of single-objective optimization and multi-objective optimization.

These works achieve an optimization process mainly to fulfill the functional requirements. However, as mentioned before in Table 1.1, one of the main issues of the conceptual design is to generate, evaluate and visualize different concepts which are usually done through sketching. These concepts must fulfill the functional requirement of the system, but they also need to consider the dependencies between components and/or subsystems. Even though product-related dependencies are known to greatly influence the overall performance of a mechatronic system, most of the methods to synthesize a mechatronic system do not consider them at the early stage of the design. Indeed, these dependencies have a great impact on the success of the system as mentioned by [14, 15, 33]. A dependency can be positive which means the integration of components will help fulfill the functional requirements. As opposed to a negative dependency which will hinder the achievement of the functional requirements. The source of the negative dependencies can be associated with thermal effect, electromagnetic effect, etc [14, 15]. The dependencies of interest for this thesis are related to the spatial placement of the components considering that one of the most important aspects of the conceptual design is sketching. Consequently, **the main subject of this Ph.D. project will be the layout design of a mechatronic system. The layout design is defined as the choice of components and their placements**. The computer-aided tools developed will also offer 3D visual support for visualizing the layout concepts. For the rest of this thesis, the term layout design synthesis will summarize the description above.

2.2 Layout design optimization

Layout design optimization is a combinatorial optimization problem (COP) which means that there is a limited number of viable solutions. COPs have been sorted in the nondeterministic polynomial time complete (NP-C) problem. This means that the computational time needed to solve the problem increases exponentially with the size of the problem [34, 35].

Research on NP-C problem in mechatronics and robotics as well as other fields such as facility layout problems and container loading problems allowed to pinpoint the popular optimization methods to solve COPs. A summary of the main methods is presented in the following paragraphs.

Exact methods: exact methods use two main strategies to solve COPs: Branch & bound and cutting plane [34-36]. The principle of the Branch & bound is to split the main problem into multiple subproblems and to solve the subproblems individually. At each iteration, the unimproved and unrealistic are not going to be considered. For example, [37] presents a model predictive control method to define the configuration of multiple vehicles while using a collision avoidance algorithm based on branch &bound. First, an optimal configuration is calculated. Then, if a predicted collision is detected a subproblem is generated to avoid the collision. In [38], a branch & bound algorithm is used to minimize the energy consumption in a permutation flow shop scheduling problem. Each machine can have two states: idle or operational. The goal is to reduce the number of idled machines since they are consuming energy while doing nothing. The branch & bound is used to distribute and synchronize jobs to the machine considering the overall energy consumption.

For the cutting plane strategy, the idea is to set several constraints to reduce the solution space. For example, [39] treats the cutting and packing problems more precisely as the one-dimensional cutting stock problem. This specific type of problem consists of cutting multiple specific pieces with a defined size from a minimum number of stocks (materials). Using the cutting plane algorithm, a set of constraints represented by inequalities were added to the standard cutting stock problem formulation to minimize the number of stocks. In [40], the capacitated arc routing problem was optimized using a cutting plane algorithm. In this paper, the capacitated arc routing problem is defined as “finding a set of minimum cost routes that service all the positive-demand edges of a given graph, subject to capacity restrictions”. An example of a capacitated arc routing problem could be mail delivery. This work proposes a new set of inequalities to solve the capacitated arc routing problem. The conclusion on these methods are reported in Table 2.1

Table 2.1. Advantages and inconveniences of exact methods

Advantages	Disadvantages
Convergence towards global optimum	Works for small size problems
An optimal solution may be obtained	Computation time tend to be “infinite” when the size of the problem gets too large
Consider the whole solution space	Viable if the main problem can be split into subproblems

Intelligent methods: There are three main approaches: expert system, fuzzy system, and artificial neural networks [34-36]. Expert systems are artificial intelligence systems taught to solve a specific problem with the aid of a human expertise database. Fuzzy systems are often used to quantify a qualitative specification to choose between different alternatives. The artificial neural network uses many examples of a system to create a model of a problem. Once this model is done, a solution will be given for a set of inputs. In [41], the layout design of a space station was done with a new evolutionary algorithm called expert-guided evolutionary algorithm with tree-like structure decomposition. The algorithm starts by dividing the volume into subspaces with a tree-like structure. Then the layout is optimized by the evolutionary algorithm. Also, to avoid the local minimum an interface was developed by the author to allow a human expert to intervene. In [42], a layout design optimization algorithm was developed using the fuzzy constraint theory. The fuzzy constraint theory is used to evaluate the performance of layouts and define the best layout. The conclusion on these methods is reported in Table 2.2.

Table 2.2. Advantages and inconveniences of intelligent methods

Advantages	Disadvantages
After the training phase, the optimization time is low.	Need for expert knowledge

Uses years of expert knowledge	Need for training
	Unpredictable reaction when faced with a situation unseen during the training phase

Approximation methods: The most recent approximation methods are often based on evolutionary algorithms which in general start with a randomized initial population or a known good initial solution and try to converge toward the global optimal solution using different strategies. The strategies adopted can be a global search such as a genetic algorithm or a local search such as a Tabu search[34-36]. A few applications of the approximation methods are given in the next paragraphs.

In [43], a multi-objective genetic algorithm was implemented to design a micro-electro-mechanical system. The algorithm had to define the configuration of one center of mass and four serpentine springs while considering a set of constraints. In [44], a driller control console layout was designed to improve the human-machine interface for the manipulators who controlled the driller. The objective function optimized by a genetic algorithm is based on the human-machine interface theory and mathematical model. In [45], a generic mechatronic design algorithm was developed. A bond graph representation is used to characterize the mechatronic system and two genetic algorithm loops are used to optimize it. The first loop act on minor modifications of the design while the second loop optimise the overall design of the system. In [46], a combination of the genetic algorithm and the finite element analysis is applied to the layout optimization of a 2D and a 3D workpiece from an aircraft body panel. In [47], a layout design of human-machine interaction interface of cabin methodology has been developed using the theory of cognitive psychology combined with Genetic and Ant Colony algorithms. This methodology has then successfully optimized the layout design of a driller control room on a drilling rig which is a human-machine interaction interface of 16 manipulators. The conclusion on these methods is reported in Table 2.3

Table 2.3. Advantages and inconveniences of approximation methods

Advantages	**Disadvantages**
Obtains excellent solution	Can use a lot of resources
Reasonable computational time	Some methods need an initial solution
Can solve highly non-linear objective functions	Might be vulnerable to premature convergence (local extremum)

The literature review done in this sub-section presented the most used algorithms used for layout optimization. The optimization of a layout of a mechatronic system is a problem with numerous combinations since there is a lot of different viable sensors and configurations that an engineer can use. The exact methods would not be an adequate choice because they cannot handle high-dimensionality problems (more than 25 elements [35]) very well. Furthermore, exact methods tend to perform poorly when the space is not smooth which might be the case here.

Methods that need an initial solution would also be impractical since an initial solution for a specific mechatronic system might not exist. Considering an improvised initial solution combined with highly non-linear objective functions, there is a high possibility that the final solution would be a local optimum which can be far from the global optima. For this reason, any methods needing an initial solution will not be considered as a sole optimization algorithm.

The intelligent methods were not considered as a possible optimization algorithm, as they would not be practical because of the lack of information/data about a specific mechatronic system during the conceptual design phase to obtain an adequately trained intelligent model.

Approximation methods will therefore be considered in this Ph.D. work to optimize the choice of sensors as well as the layout design of a mechatronic system based on the component's dependencies since they can optimize highly non-linear objective functions and have the potential to output excellent solutions within a reasonable computational time.

CHAPTER 3 RESEARCH QUESTION, OBJECTIVES, AND HYPOTHESIS

3.1 Research question

Based on the conclusion of the literature review carried-out in Section 2.1 of Chapter 2, the research question is:

How to provide support to designers for the layout synthesis of a mechatronic system during the conceptual design phase?

Figure 3.1 summarizes the goal and the implication of the main research question of the project as it shows the main questions that need to be answered at different stages in order to answer the main research question.

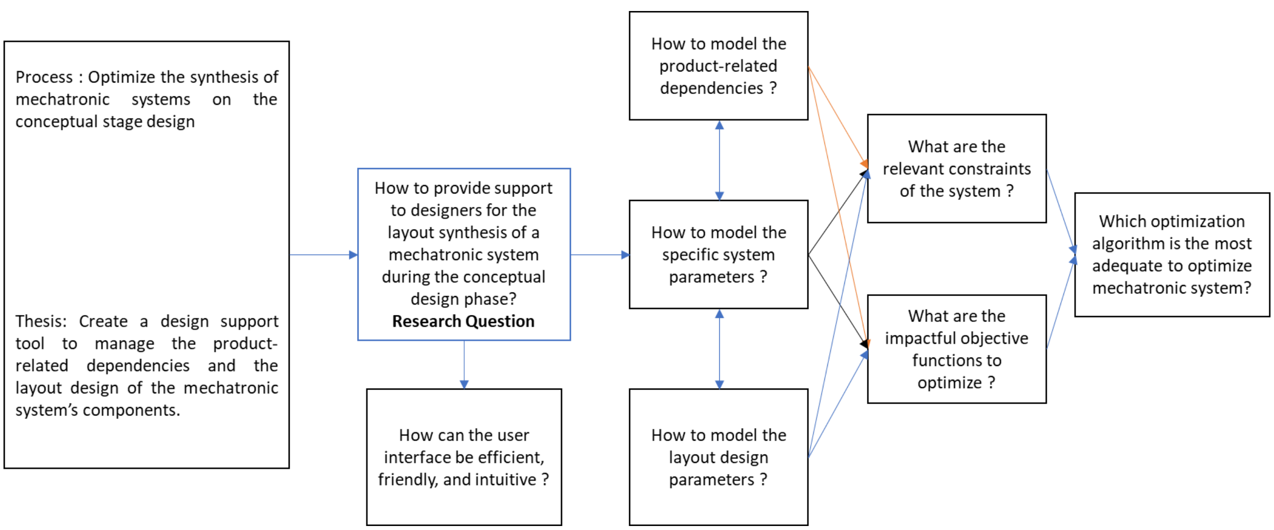


Figure 3.1. Summary of the research project

3.2 Objectives

The main objective of this project is as follows:

Develop a multi-objective design support tool based on evolutionary computing to synthesize the layout design of a mechatronic system.

Sub-Objective 1. Develop the mathematical formulation for layout constraints as well as the functional restrictions and objective functions.

Sub-Objective 2. Optimize the layout design of mechatronic systems using evolutionary computing algorithms.

Sub-Objective 3. Test and validate the proposed algorithm on a mechatronic system case study.

Figure 3.2 presents a diagram to summarize the main modules of the CAED tool developed in this thesis that will fulfill the stated objectives in this sub-section. It also highlights the areas of significant contributions of this thesis.

3.3 Hypothesis

Based on what was covered in the literature review and the research question at hand, the hypothesis which will be confirmed or infirmed in this project is:

By using a multi-objective layout design optimization algorithm based on product-related dependencies for the synthesis of mechatronic systems, it is possible to improve:

1. The exploration of the layout design space of the mechatronic system.
2. The performance of the mechatronic system.

To validate these hypotheses, a case study will be used. Indeed, the case study is the layout design of an autonomous greenhouse as part of a Canadian Space Agency funded research project led by Professor G. Beltrame (Polytechnique Montréal). The goal is to integrate a miniature autonomous greenhouse into a nanosatellite. This case study is suitable for the validation of the developed tools for two main reasons. First, considering the volume restriction of a nanosatellite, choosing an optimal layout for the design is crucial as it would allow space minimization while keeping the full functionality of the device. Second, the closeness of mechanical and electrical components and the limited free space available can greatly affect the growth of the plant. Indeed, the closeness of electrical components can yield inaccurate measurements that would misguide the control system. This case study is the subject of two published scientific papers reported in Chapter 5 and Chapter 6.

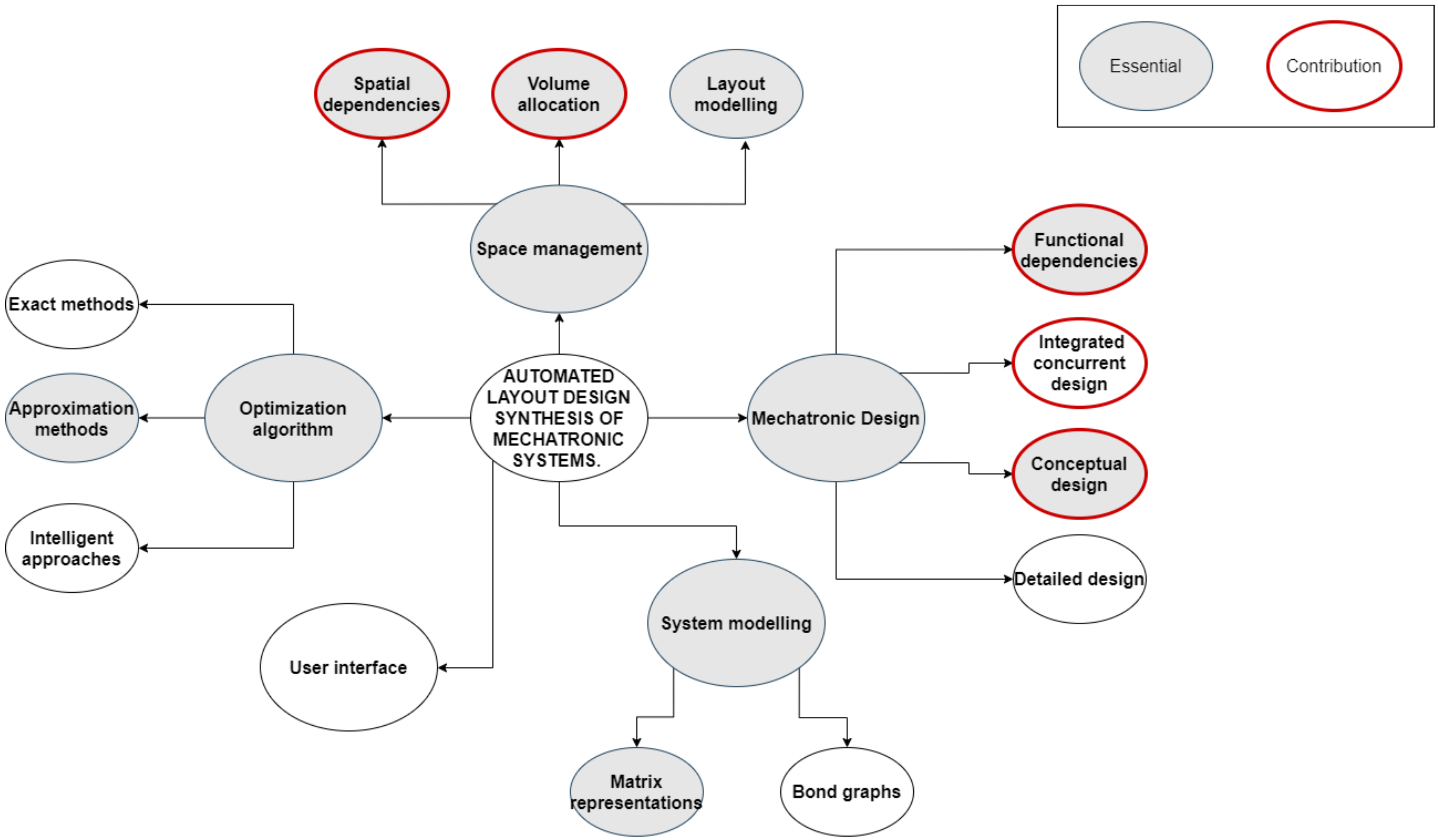


Figure 3.2. Area of relevance and contributions diagram of the research project

CHAPTER 4 TOOL SPECIFICATIONS & PRODUCT-RELATED DEPENDENCIES

4.1 Computer-aided tool specifications

The proposed computer-aided tools should allow the improvement of mechatronic systems' design by optimizing the position of the components based on the components' dependencies and the system's functionalities. It should also greatly assist the synthesis of mechatronic systems since the components' dependencies bring issues early in the design process that needs to be managed. It will be possible to see this phenomenon in the case study considered in this project. First, the essential and optional needs will be described.

4.1.1 Essential needs

The essential needs are defined as the minimum functionalities that need to be included in the developed computer-aided tool to obtain a layout design. Hence, the proposed tool must include the following list of requirements:

- Able to acquire user's input
- Able to generate and evaluate layout designs
- Able to present a layout design to the user

4.1.2 Optional needs

The optional needs are defined as the desired functionalities that could be included in the developed computer-aided tool to improve user comfort and product design process. Hence, the proposed tool could benefit from the following list of requirements:

- Offers an intuitive interface
- Interfaces the proposed tool to other tools
- Able to output more than one layout design

4.1.3 Constraints

The main constraint concerns the learning curve associated with the use of the computer-aided tool. Hence, the proposed tool should follow these guidelines:

- The user should not spend more time learning how to use the proposed computer-aided tool than on designing systems.
- The workflow of the tool should be similar to the workflow usually used by designers.

4.2 Product-related dependencies modeling

In [13], a comprehensive study was carried out on challenges faced by engineers in mechatronic systems design, and a comprehensive classification of product-related dependencies was proposed. This classification will be the basis for the dependencies modeling for this thesis.

The challenges presented in [13] have been classified into six main categories related to mechatronic systems design: activity, mindset, competence, organizational, product, and other. The activity, mindset, competence, organizational and other categories contain challenges that concern the education of mechatronic engineers or how the company organizes itself for integrating the different engineering disciplines involved. For example, a company must adapt its communication systems to allow a proper knowledge transfer from one domain to another. This could also lead to a change in the facility's layout to encourage the collaboration of engineers from different domains to obtain concurrent engineering. These challenges will not be considered in this research project. Indeed, the issues covered by this Ph.D. thesis are only related to the design of the mechatronic product itself. It is important to note that product-related challenges remain the most complex to tackle and are known for having the most impact on the performances of the final product [12, 13]. These product-related dependencies stem generally from the following challenges [13]:

1. Lack of common understanding of the overall system
2. Difficulty in assessing the consequences of choosing between two alternatives
3. Lack of a common language to represent a concept
4. Modeling and controlling multiple relations in the product concept
5. Being in control of the multiple functional states of the product
6. Transfer of models and information between domains.

The work presented in [15] deepens the classification of the product-related challenges and developed a system to classify the product-related dependencies. This classification was developed as a methodology that can be used by design teams to identify and influence the dependencies within a mechatronic product. These dependencies have been described as appearing between product attributes: function, property, and structure. Based on the reviewed literature [48-53], it was proposed to use the terms functions, properties, and means to describe the product across the engineering disciplines, where:

- Function (Fu): the task of a product. A function is defined as the task that a product must be able to complete.
- Property (Pr): the property is defined as a characteristic of the product caused by the chosen means
- Mean (M): the mean is defined as a sub-system or method used to accomplish a function.

As in [15, 33], in this thesis, these terms are also used to define product-related dependencies. It is worth noting that a product-related dependency can exist between two or more attributes of the product. These dependencies are abstract and can be defined between the product attributes in terms of Fu-Fu, Fu-M, Pr-M, M-M, Pr-Fu, Pr-Pr, as illustrated in Figure 4.1.

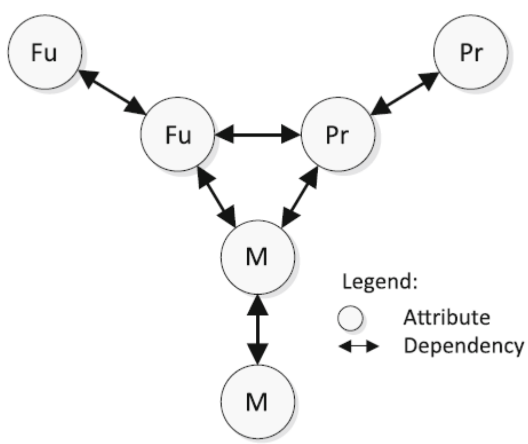


Figure 4.1. Representations of the product attribute dependencies

Six dependencies can be created (shown in Figure 4.1) as a result of a purely combinatorial exercise:

1. Fu–Fu: A dependency between two functions is described by the link that is created when a function reacts to a stimulus created by another function.
2. M–M: A dependency between two means in the product.
3. Fu–M: A function is realized by a mean. This mean can then help further decompose the function into sub-functions, which creates the dependency between functions and means.
4. Pr–M: Properties are the result of choosing means, thereby creating dependencies between means and properties.
5. Pr–Fu: There is no direct relation between a function and a property. Both are related to means according to the Theory of Domains [48, 49]. A link can be established by combining the two relations Fu–M and the Pr–M. Therefore, the Fu-Pr relation will not be described as a separate relation.
6. Pr–Pr: There is no direct relation between a property and a property, and the argumentation is the same as the Pr-Fu relation.

In this thesis, Fu-Fu dependencies will not be considered since these dependencies are defined by the client or very early by the designer of the mechatronic system. Pr-Fu and Pr-Pr will not be managed either as there is no direct relation between function and property or property and property since they must use means to influence each other as explained above. Therefore, the considered dependencies are the following (as defined in [15]):

1. Fu-M in terms of 1) *Fu-M disposition* where functions transformed into means and then into sub-functions; 2) *Cumulative Fu-M* where several means can contribute to realizing a single function and 3) *Adverse effect* that occurs when finding means to realize a function, also may create solutions with adverse effects.

2. Pr-M in terms of 1) *Property scheme* as in how the means contribute to the realization of a property.

3. M-M in terms of 1) *Volume allocation* which is an aspect that plays a significant role when looking at the relations between means; and 2) *Physical interface* between two components (the link between components).

This thesis focuses on automatically integrating the six product-related dependencies presented above when designing the layout of a mechatronic system.

CHAPTER 5 ARTICLE 1: EVOLUTIONARY LAYOUT DESIGN SYNTHESIS OF AN AUTONOMOUS GREENHOUSE USING PRODUCT-RELATED DEPENDENCIES

The article presented by Yann-Seing Law, Yuanchao Ma, Aurelian Vadean, Giovanni Beltrame, and Sofiane Achiche in this chapter is published in the journal *Artificial intelligence for Engineering Design, Analysis and Manufacturing.*

Law-Kam Cio, Y., Ma, Y., Vadean, A., Beltrame, G., & Achiche, S. (2021). Evolutionary layout design synthesis of an autonomous greenhouse using product-related dependencies. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, 35(1), 49-64. doi:10.1017/S0890060420000384

5.1 Abstract

The development of autonomous greenhouses has caught the interest of many researchers and industrial considering their potential of offering an optimal environment for the growth of high-quality crops with minimum resources. Since an autonomous greenhouse is a mechatronic system, the consideration of its sub-systems (e.g. heating systems) and components (e.g. actuators, sensors) interactions early in the design phase can ease the product development process. Indeed, this consideration could shorten the design process, reduce the number of redesign loops, and improve the performance of the overall mechatronic system. In the case of a greenhouse, it would lead to a higher quality of the crops and better management of resources. In this work, the layout design of a general autonomous greenhouse is translated into an optimization problem statement while considering product-related dependencies. Then, a genetic algorithm is used to carry out the optimization of the layout design. The methodology is applied to the design of a fully autonomous greenhouse (45 cm X 30 cm X 30 cm) for the growth of plants in space. Although some objectives are conflictual, the developed algorithm proposes a compromise to obtain a near-optimal feasible layout design. The algorithm was also able to optimize the volume of components (occupied space) while considering the energy consumption and the overall mass. Their respective summed values are 2844.32 cm3, which represents 7% of the total volume, 5.86 W, and 655.8g.

5.2 Introduction

Autonomous greenhouses have been developed as a means to grow plants in an optimal environment defined in terms of humidity, temperature, lighting, and gases concentration (Sabri, Aljunid, et al. 2011, Vera, Osorio-Comparán, et al. 2017). Such technology has many advantages such as better management of resources, off-season growth of high-quality crops, etc. (Rabago, de Santago, et al. 2013, Abas and Dahlui 2015). However, the design of such a greenhouse is not an easy task. One major issue is its layout design. In general, the layout design is the spatial management of different elements in a given space based on domain-specific objectives. In architecture and urban design, the ease of access from rooms to hallways could be a domain-specific objective (Koenig and Schneider 2012). Another example closer to the design of a greenhouse is the design of a satellite where the domain-specific objective is to ensure that every element can carry out its functions (Taura and Nagasaka 1999). Furthermore, layout and domain-specific objectives are usually conflictual. Indeed, the layout design of an autonomous greenhouse needs to manage two important conflictual objectives, which are to maximize the size of the pack-soil and to minimize the amount of resources needed. On top of those objectives, the greenhouse must be functional and must allow an optimal environment for the growth of specific plants. Since an autonomous greenhouse is a mechatronic system, it is essential to consider interactions among the components during the design phase (Mohebbi, Baron, et al. 2014, Torry-Smith, Mortensen, et al. 2014) to increase its efficiency. To overcome this challenge, the layout design of a greenhouse consists of concurrently solving three main issues.

The first issue is to define the size and location of the pack-soil as well as the storage of resources (e.g. water tank). The size of both the pack soil and storage of resources is defined by the plants chosen to be grown in the greenhouse. Indeed, plant seeds need a space between each other to properly grow, which means the size of the pack-soil depends on the number of seeds. As for the storage of resources, each plant needs to consume a certain amount to properly grow as well. This means that prior knowledge about the ideal environment of the chosen plant is needed.

Since the greenhouse has to be autonomous, a set of sensors to acquire data of the current environment is required. Hence, the second issue is the need to define the size and performance of sensors (e.g. humidity) and actuators (e.g. water pumps), that control the greenhouse environment.

A set of actuators must be present to modify the environment. Depending on the model of sensors and actuators, the size and the performance of the system can change in terms of the volume allocated for both the plant growth and the assessment of the environment.

As for the locations of these components, they depend on the third and last main issue, which is the interaction between all the greenhouse components. Indeed, the sensors, actuators, pack soil, and storage of resources must be carefully positioned to avoid the malfunction of the greenhouse. A malfunction can occur when one component has adverse /negative effects on other components as reported by Chouinard et al. (Chouinard, Achiche et al. 2017, Chouinard, Achiche, et al. 2019). For example, the temperature sensor cannot be placed close to the heater to avoid an erroneous reading of the temperature, which could lead to freezing or drying the pack-soil. It is worth mentioning that in this paper we will consider the same categories in terms of heat, electromagnetic effects (EMFs) and vibrations when present.

The main contribution of this paper is the development of a methodology to formulate an optimization problem for the automated layout design of a greenhouse considering the three issues mentioned above in terms of 1) size and location of pack soil and storage of resources, 2) size and performance of sensors and actuators and 3) location of components considering their adverse effects.

First, the problem statement of the layout design of a greenhouse is carried out and translated into an optimization problem. Using evolutionary computing (a genetic algorithm), the size, location, and parameters of every component of the greenhouse are optimized.

5.3 Background and Literature Review

We focus on studies of autonomous greenhouse design and greenhouse layout design. The research trends concerning autonomous greenhouses are generally targeting climate control, wireless networks, and integrated design.

5.3.1 Autonomous Greenhouse

This part of the literature review is carried out to identify the product specifications needed to design an autonomous greenhouse. It is destined to systematically formulate the problem statement presented in the next section.

Researchers are interested in improving the assessment of the environment of a greenhouse in different situations. Castañeda-Miranda and Castaño (Castañeda-Miranda and Castaño 2017) made a comparison between an autoregressive algorithm (ARX) and an artificial neural network to predict the internal temperature of a greenhouse based on both external and internal parameters, such as the outside temperature, and the humidity. This study found that the artificial neural network provides a better prediction than ARX. The method is tested with data from a greenhouse and a weather station. Romantchik et al. (Romantchik, Ríos, et al. 2017) designed a cooling control system to prevent the temperature of the greenhouse from exceeding 25°C using fan-pad systems. Based on a ventilation system, an algorithm is developed to support the design of a photovoltaic solution, which would supply the necessary energy. Vera et al. (Vera, Osorio-Comparán, et al. 2017) built a greenhouse with its environment controlled by monitoring the temperature, humidity, carbon dioxide, and illumination levels. In this greenhouse, a heat sensor and a heater controlled the temperature. A humidity sensor for soil controlled the humidity of the soil and a solenoid valve fed the water. A relative humidity sensor and a micro sprayer controlled the humidity of the air. A carbon dioxide sensor and a fan regulated carbon dioxide levels. Finally, a combination of a luminosity sensor, luminosity source, and a timer controlled the illumination. In their research works, Abas et al. (Abas and Dahlui 2015, Abas, Salman, et al. 2016) used the temperature, humidity, and irradiance acquired by sensors to control the temperature, the humidity, and the interval of time between the activation of an irrigation system. An intruder detector activates the intruder repellent, using electric fences and ultrasonic sound. The whole system is powered by a solar panel. Matos et al. (Matos, Gonçalves, et al. 2015) automated fodder production part of a hydroponic system (growing plants without soil). The system automatically placed seeds in the trays, managed the nutrient solution preparation and the water distribution. Paraforos and Griepentrog (Paraforos and Griepentrog 2013) used a multivariable control of a greenhouse in terms of carbon dioxide quantity, temperature, and humidity. A non-linear steady-state model is used to develop an input/output linear decoupled controller by linearizing and discretizing the model for given operation points. Pala et al. (Pala, Mizenko, et al. 2014) proposed a control strategy for aeroponic systems. In their work, first, a network of sensors and actuators is implemented to monitor the environment through a user interface to relay the information to the user and to allow the user to manually control the system if needed. Second, the aeroponics system design called Aero Pot is presented. The Aero Pot is a nutrient distribution system composed of two nozzles, where the nutrients are given to a plant and a motor to move the nozzle from one plant to another. The optimization of the system is done using a genetic algorithm (GA). The user can define the number of lights and pumps and their power consumption. The GA first optimizes the power consumption, then provides the power schedule of components for one day. The preliminary results of this optimization are promising since the first experiments demonstrated that with reasonable power consumption, the plant was healthy. Rabago et al. (Rabago, de Santago, et al. 2013) designed a solar-powered automatic greenhouse. The system controlled the moisture, the temperature, and the irrigation schedule using information about the humidity, moisture, temperature, and soil mixture. The components were solar panels, batteries, valves, a relative humidity sensor for the air, a humidity sensor for the soil, a halogen lamp to heat, fans, and microcontrollers. In the work presented by Hahn (Hahn 2011) a fuzzy controller is developed to prevent tomatoes from cracking because of overheating. To control the temperature of the crop, a shading screen control, and irrigation system control are used. The solar radiation, the substrate temperature, and the canopy temperature were the inputs of the fuzzy controller while the output was the command sent to the irrigation system and the motor controlling the shading screen position. Schubert et al. (Schubert, Quantius, et al. 2011) proposed a greenhouse module design for extraterrestrial habitats. The system design started by suggesting multiple designs of the greenhouse module, which contained the growth system. The growth system is composed of the germination unit which starts the growth of the plants before transferring them in a grow pallet. The growth pallet is then placed in a growth channel, where the environment is controlled to satisfy the needs of a given plant for every stage of its growth. The growth channel unit is filled with multiple growth channels installed on a conveyor system. Finally, the greenhouse module is integrated with eight subsystems to control the environment of the plants. Xu and Li (Xu and Li 2008) developed a greenhouse control system using multiple agents. The intelligent control center is composed of a collecting artificial agent to gather data from the greenhouse, which is processed by the data processing one. The data transmitting agent then stores the information in a database. The intelligent control center also has an agent, which controls the greenhouse environment in terms of temperature, illumination, humidity, and carbon dioxide concentration. Herrero et al. (Herrero, Blasco, et al. 2007) implemented an elitist multi-objective evolutionary algorithm to identify the parameters of a greenhouse model. The greenhouse model used is composed of 15 parameters to estimate the internal temperature and humidity. Using the implemented evolutionary algorithm and a set of data obtained from an operating greenhouse, a Pareto optimal set of the greenhouse model was found. Then, the greenhouse model from the set of criteria closer to the ideal optimality criteria was validated using another set of data obtained from the same operating greenhouse.

Most of the time, researchers use wireless communication networks of components to monitor and control the climate of the greenhouse. Hence, the use of wireless communication adds complexity to the design of autonomous greenhouses. In Azaza et al. (Azaza, Tanougast, et al. 2016) a fuzzy logic-based controller combined with a wireless communication system based on the ZigBee platform controlled the climate of a greenhouse. The temperature, humidity, carbon dioxide, and illumination integrated a fuzzy set beside the external meteorological variables and the setpoints given by a user. Then, a decision scheme, which represented the observer design flow, was set up in terms of ventilation, heating, humidification, and dehumidification. Finally, the fuzzy logic controller is implemented using FPGA programming to assess the greenhouse environment in terms of temperature and humidity using the heating and ventilating system. In Krishna et al. (Krishna, Madhuri, et al. 2016) a wireless network based on ZigBee assessed the greenhouse environment in terms of humidity, moisture, and temperature. The sensors sent the data to an ARM7 microcontroller, which used a ZigBee transmitter. The data is then sent in real-time to a central unit combined with a Zigbee transmitter to monitor the data. Goumopoulos et al. (Goumopoulos 2012, Goumopoulos, O'Flynn, et al. 2014) developed an automatic irrigation control system using machine learning and wireless sensor network information formed of multiple nodes, where one node monitored a zone containing multiple plants. The control strategy had three main components. First, the ontology of the plant is used to define the rules for decision-making based on prior knowledge. Second, the Decision Support System acquired all the information from the data analysis of the greenhouse to make a decision for the well-being of the plant. Finally, machine learning is used to obtain new connections among the data acquired. Three types of sensors are used: soil moisture sensors, humidity sensors, and thermistors for air temperature. In Sabri et al. (Sabri, Aljunid, et al. 2011) a fuzzy logic approach and a wireless network based on ZigBee controlled the greenhouse. The difference in temperature and humidity of the greenhouse are monitored and used as inputs for the fuzzy controller. The fan, heater, and humidifier command are the output of the controller. Ferentinos et al. (Ferentinos, Katsoulas, et al. 2017) made a sensitivity analysis of a wireless sensor network (WSN) reading in function of solar exposition level. They made experiments to evaluate which readings were more stable between the three expositions level. The first exposition level was labeled “exposed nodes” and was a WSN directly exposed to solar radiation. The second one was labeled “boxed nodes” and was fully protected from solar radiation by a ventilated box. The last one was labeled “shaded nodes” and used a metallic surface to protect the nodes from direct sunlight. The analysis showed that the most stable reading was from the “shaded nodes”. Hence, they used “shaded nodes” in a commercial cucumber greenhouse. Using this system, they studied plant conditions such as the transpiration of the crops, the leaf temperatures, etc.

The climate control of a greenhouse adapted and integrated into infrastructure or uncommon environments is also an area of research. Nadal et al. (Nadal, Llorach-Massana, et al. 2017) proposed an integrated rooftop greenhouse (iRTG) at the Autonomous University of Barcelona campus. To grow crops successfully, the iRTG recycled many resources from the building to control the airflow and temperatures of a greenhouse. Indeed, the whole building adopted a mode of operation depending on the season to control the ventilation system of the building. For example, when the temperature was too high, the windows are open to cool down the greenhouse. With the iRTG, tomatoes and lettuce crops are produced. Poulet and Doule (Poulet and Doule 2014) made a preliminary design of a greenhouse for food and for a Zen garden (for crew emotional state) called GreenHab. The GreenHab purpose is to eventually be used as a greenhouse on Mars. At the moment, the growth of different lettuces in GreenHab is being studied and tested at the Mars Desert Research Station of the Mars Society of Utah. The study of GreenHab is carried out in terms of temperature, illumination, and humidity. The system is only partially automated since the crew can also modify the environment of the greenhouse. Giroux et al. (Giroux, Berinstain, et al. 2006) designed a greenhouse for Mars’s environment. The greenhouse is equipped with sensors to monitor humidity, temperature, and radiation. The actuators used were the heaters, the fan, and the exhaust fan, which are controlled based on the temperature of the greenhouse environment. The greenhouse can operate all year long by changing its operation mode based on the external environment, such as the outdoor temperature. Furthermore, the water distribution is done manually. An analysis of a greenhouse environment and power consumption over a year is also carried out.

From this literature review, one can conclude that the main function of an autonomous greenhouse is to ensure the growth of plants by controlling the climate of the greenhouse. Based on this main function, a list of product specifications will be listed in the next section for the layout design of an autonomous greenhouse.

5.3.2 Greenhouse Layout Design

Komasilovs et al. (Komasilovs, Stalidzans, et al. 2013) used a GA called GAMBot-Eva to optimize the design of a robotic system traveling through the greenhouse layout, evaluating health conditions of plants and spraying pesticides on them if needed. The optimization problem took into account the tasks of the robot, the price, and the energy consumption of the robot components. However, the greenhouse layout is fixed, and the optimization is done with the parameters of the robotic system. Eben-Chaime et al. (Eben-Chaime, Bechar, et al. 2011) optimized the overall cost of a greenhouse layout based on different expenses, such as seedlings and labor costs, for different greenhouse layouts. The layout can be changed in many ways to reduce the overall cost of the greenhouse. Four different layouts are presented, and the overall cost is calculated. Hence, the performance of the greenhouse is not taken into account to choose the layout. Ferentinos et al. (Ferentinos, Tsiligiridis, et al. 2005, Ferentinos and Tsiligiridis 2007) optimized the topology of the wireless sensor network for precise agriculture applications in terms of energy consumption and sensor sensitivity characteristics. The problem is also subjected to connectivity and spatial density constraints. This multi-objective optimization is turned into a single objective optimization using a weight sum approach. The optimization problem is solved with a GA using binary representation. A dynamic optimal design algorithm is also included in the GA for sensors with battery capacities. Ferentinos et al. (Ferentinos and Albright 2005) also used a GA using binary representation for the placement of the lighting system for a greenhouse. The optimization considered different lighting characteristics, such as light uniformity, as well as economical aspects, such as the investment costs. They also used a penalization function to avoid shading effects that can happen when designing a lighting system. The problem is rewritten as a single objective optimization problem using a weight sum approach.

5.4 Main objective and Contributions

Our previous section shows that, until now, only a few works are covering the greenhouse layout design. Moreover, these works do not fully consider the layout in the design of the autonomous greenhouse. This might be caused by an incomplete problem statement of the layout design. Hence, the problem statement of the layout design needs to be improved to rigorously define the components needed and their interaction.

To the best of our knowledge, the evolutionary layout design of greenhouses considering the placement of components and their interaction (dependencies) has never been done. The main contribution of this paper is a novel methodology to formulate a more comprehensive problem statement for layout design as shown in Figure 5.1. Furthermore, the methodology also considers the translation of a problem statement from an engineering design perspective to the formulation of an optimization problem. First, a problem statement is developed by identifying the components of an autonomous greenhouse and their interactions. The problem statement is then translated into an optimization problem which is solved using a GA.

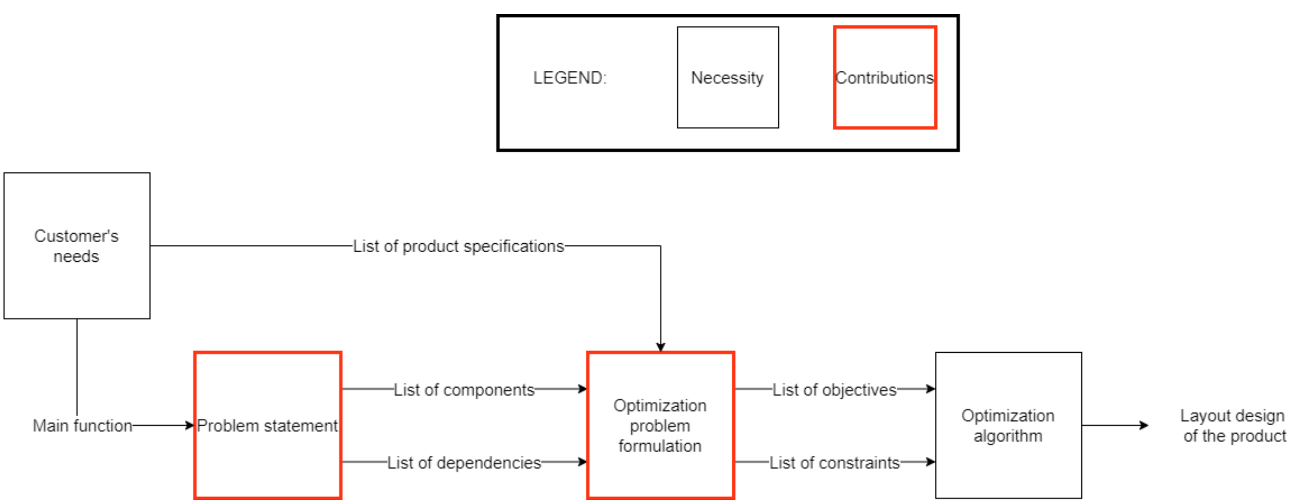


Figure 5.1 Highlights of contributions in the general methodology

The rest of the paper will be structured as follows: Section 5.55.5 defines the problem statement of the layout design of a greenhouse. Section 5.6 reviews similar problems solved using algorithmic approaches and describes the implementation of the GA. Section 5.7 reports the results and analysis of the design of an autonomous greenhouse obtained by the GA used in this paper. Based on these results, the main limitations and future research avenues are identified. Finally, section 5.8 concludes this paper.

5.5 Problem Statement

5.5.1 Identification of Autonomous Greenhouse Components and Interactions

Using the six product-related dependencies presented by Torry-Smith et al. (Torry-Smith, Mortensen, et al. 2014), we identify the components needed for the development of an autonomous greenhouse. The product-related dependencies are defined in terms of function (Fu), property (Pr), and mean (M) of the product and their interactions. The function is defined as the tasks that systems and/or subsystems must be able to complete. Here, property refers to a property of the system such as the mass. Often the property is affected by the choice of means. Finally, the mean is what is used to accomplish a function. The approach developed here is framed by the product-related dependencies methodology which offers a general framework that is generally used in multi-domain systems design. This can be easily generalized to other complex systems design activities, such as for mechatronics, where designers can have a more thought-out starting point early in the design process.

Fu-M can be explained by the following: a function is realized by a mean, which can be further decomposed into sub-functions and so on. Using this definition, some components can be identified:

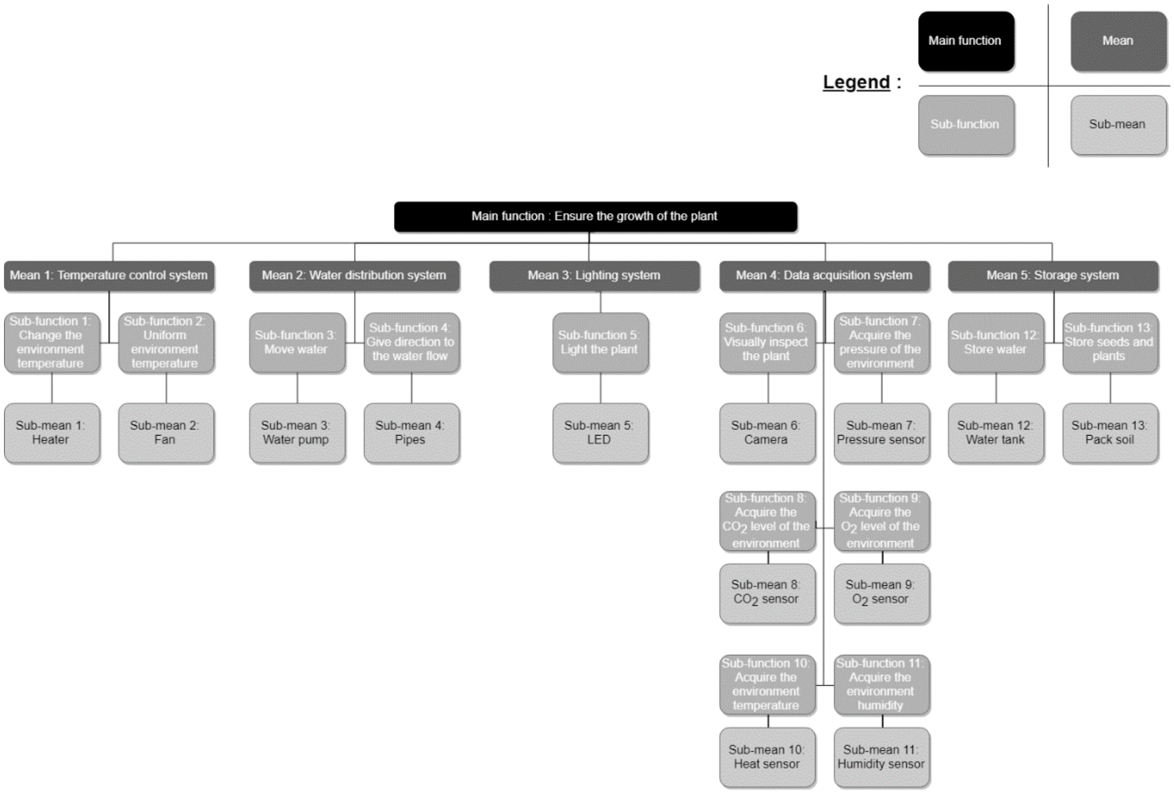


Figure 5.2. Organigram representing the Fu-M decomposition

1) Fu-M disposition & Cumulative Fu-M: Figure 5.2 presents the results of the Fu-M decomposition to identify the components.

2) Adverse effect: as mentioned above, the adverse effects considered in this paper are categorized in terms of heat, vibration, and EMFs. We also add another category which is the obstruction of a component field of view. This will be represented by OBS in Table 5.1. Although it is difficult to evaluate this fourth category considering that the dimensions of each component are still unknown, it is possible to estimate an order of magnitude for each of the components. For example, generally, a water tank is bigger than a heat sensor, hence, the water tank has more chance to obstruct the field of view of a component such as a camera. A table such as Table 1 is used by Chouinard et al. (Chouinard, Achiche et al. 2017, Chouinard, Achiche, et al. 2019) to identify components generating adverse effect and those who are affected by these adverse effects. The first column is the list of components. The second column is a qualitative evaluation of the intensity of adverse effects generated by

the component. The last column is a qualitative evaluation of the negative impact of the component affected by adverse effects.

Table 5.1 Relation between components and adverse effects

Component	Affecting	Affected
Heater	Heat: High OBS: Low	-
Water tank	OBS: High	-
Pack soil	OBS: High	-
Heat sensor	EMF: Low OBS: Low	Heat: High EMF: Low
Camera	EMF: Low OBS: Medium	EMF: Low OBS: High
Fan	EMF: Low OBS: Low	OBS: Low
LED	Heat: High OBS: Low	OBS: High
Water pump	Vibration: High EMF: Medium OBS: Medium	Vibration: Medium
Pipes	OBS: Low	-

CO_2 sensor	EMF: Low OBS: Low	Heat: Low EMF: Low
O_2 sensor	EMF: Low OBS: Low	Heat: Low EMF: Low
Humidity sensor	EMF: Low OBS: Low	Heat: Low EMF: Low
Pressure sensor	EMF: Low OBS: Low	Heat: Low EMF: Low

Considering Table 5.1, it is possible to identify the most detrimental combinations of components and to formulate the following guidelines to avoid these adverse effects:

i. The heater cannot be close to heat sensor
ii. LED cannot be close to heat sensor
iii. The components cannot prevent the LED from lighting the pack-soil
iv. The components cannot prevent the camera from filming the pack-soil.

Pr-M can be explained by the following: a property is affected by the chosen means. Using this definition, the rest of the components can be identified:

Property scheme:

Property 1: Mass of the system. The mass is calculated by taking the sum of all the mass of the means. For autonomous greenhouse, a low mass can be an indicator of low consumption of resources and is generally favorable.

Property 2: Electrical current of the system. We approximate the electrical current by summing all the electrical current of the means. Generally, a low electrical current consumption is also favorable.

Property 3: Electrical voltage of the system. The mean with the highest voltage will be the reference point of this property. Generally, a low electrical voltage is favorable.

M-M can be explained as a dependency between two means:

1) Volume allocation:
   i. The space between the LED and the pack-soil is the volume allocated to the plant since plants grow towards the light.
   ii. Enough space must be allocated to store water and the amount of water depends on the plant.
   iii. The camera must be able to film the pack-soil as much as possible.

2) Physical interface:
   i. The water tank, water pump, and pack soil must be linked by pipes to ensure the water distribution system.
   ii. The LED must light as much as possible the pack-soil.
   iii. The fan must be close to the heater for better heat convection.

It is possible to see that by identifying the Fu-M disposition and Cumulative Fu-M, the list of components needed to synthesize an autonomous greenhouse is found:

C1. Heater: to heat the greenhouse

C2. Water tank: to contain the water for the plants

C3. Pack soil: to contain the seeds of the plant

C4. Heat sensor: to acquire temperature data from the environment

C5. Camera: to follow the growth of the plants

C6. Fan: to ensure air circulation to stabilize the room temperature

C7. LED: to provide the light necessary for photosynthesis

C8. Water pump: to allow water distribution from the water tank to the pack-soil.

C9. PCB: to monitor pressure, humidity, O_2, and CO_2 (sensors)

C10. Pipes: to link the water tank, the water pump, and the pack-soil for water distribution.

Furthermore, the adverse effects as well as the Pr-M and M-M dependencies give guidelines to handle the placement of components within the system. To assign the strength of a dependency, the design structure matrices (DSMs) (Pimmler and Eppinger 1994, Browning 2016) are used in this work. DSMs have been used for modeling interactions between components and/or subsystems in many fields such as mechatronic design. The DSM representation and the scale value represented by Pimmler and Eppinger (Pimmler and Eppinger 1994) are adapted to model the layout component interactions. In this work, a layout component interaction is composed of three characteristics: Closeness, Field of View, and Physical Connection. These three matrices can model the layout design of most complex systems including mechatronic systems.

1) Closeness of two components: In this matrix, each value follows the scale shown in Table 5.2 to evaluate the closeness of two components.

Table 5.2 Closeness strength scale

Closeness	Really far	Far	Unaffected	Close	Really close
Value	-2	-1	0	1	2

And so, the closeness matrix for the first nine components mentioned above (the pipes are excluded from the DSMs) is:

	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	0	0	0	-2	0	2	0	0	0
C2	0	0	1	0	0	0	0	1	0
C3	0	1	0	0	0	0	0	1	0
C4	-2	0	0	0	0	0	-2	0	0
C5	0	0	0	0	0	0	0	0	0
C6	2	0	0	0	0	0	0	0	0
C7	0	0	0	-2	0	0	0	0	0
C8	0	1	1	0	0	0	0	0	0
C9	0	0	0	0	0	0	0	0	0

The selection of weights for the closeness matrix is justified as follows: the assigning of the weight is based on the interaction of the components and the impact on the main function requirement which is to ensure the growth of the plant. Hence, the most detrimental interaction is related to temperature regulation. The heat sensor (C4) absolutely needs to be as far as possible from any heat sources. In our case, the heater (C1) and the LED (C7) are the main heat sources. Therefore, the (C1, C4) and (C7, C4) cells have the value -2. Still considering the temperature aspect of the greenhouse, the environment temperature must be uniform. A local hot spot or cool spot on the pack-soil could prevent the growth of plants. For this reason, the heater (C1) and the fan (C6) need to be as close as possible to assure a proper heat flow. This explains why the value of (C1, C6) cell is 2. Finally, the water distribution is composed of a water pump (C8), a water tank (C2), and the pack-soil (C3) which are connected by the tubes. The length of the tube could be reduced if all three components are closed to one another. Advantages are coming along with the reduction of tubes such as decreasing the active time of the pump which leads to a decrease in energy consumption. Hence, the (C2, C3), (C2, C8), and (C3, C8) cells have a value of 1.

2) Interaction between the field of view (FOV) of two components: In this matrix, each value follows the scale shown in Table 5.3 to estimate the importance of the interaction between the two components FOV.

Table 5.3 Field of view strength scale

FOV	Detrimental	Undesired	Unaffected	Desired	Required
Value	-2	-1	0	1	2

And so, the FOV matrix is given by:

	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	0	0	0	0	0	0	0	0	0
C2	0	0	0	0	0	0	0	0	0
C3	0	0	0	0	2	0	2	0	0
C4	0	0	0	0	0	0	0	0	0
C5	0	0	2	0	0	0	0	0	0
C6	0	0	0	0	0	0	0	0	0
C7	0	0	2	0	0	0	0	0	0
C8	0	0	0	0	0	0	0	0	0
C9	0	0	0	0	0	0	0	0	0

This matrix refers to two components which are the LED (C7) and the camera (C5) which need to illuminate and capture the pack-soil (C3), respectively. The values of cells (C3, C5) and (C3, C7) are 2 because the fulfillment of their functional requirements is more sensitive to the relative position of the pack-soil. Indeed, the LED must uniformly illuminate the pack-soil as much as it can to allow every seed a chance to grow. As for the camera, it must capture most of the pack-soil to help the operator identify a malfunction or to visually assess the health of the plants.

3) Physical connection of two components: In this matrix, each value indicates the number of links (wire and/or pipe) that two components need. And so, the physical connection matrix is:

	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	0	0	0	0	0	0	0	0	0
C2	0	0	0	0	0	0	0	1	0
C3	0	0	0	0	0	0	0	1	0
C4	0	0	0	0	0	0	0	0	0
C5	0	0	0	0	0	0	0	0	0
C6	0	0	0	0	0	0	0	0	0
C7	0	0	0	0	0	0	0	0	0
C8	0	1	1	0	0	0	0	0	0
C9	0	0	0	0	0	0	0	0	0

5.5.2 Formulation of the optimization problem

Based on the problem statement for the synthesis of an autonomous greenhouse mentioned in the last section. The translation of the problem statement to an optimization problem will be done in this section.

First, two types of objectives included in the overall objective of the solution are developed. The first type is the component-specific objectives (CSOs). CSOs concern only the design of the component itself. The second type is the layout design objectives (LDOs). LDOs cover the layout design based on the placement of components of the greenhouse.

The CSOs are:

1. Minimizing the volume (except for the pack soil and water tank volume, which need to be maximized)
2. Minimizing the mass.
3. Minimizing the energy consumption.

The LDOs are:

1. Minimize the distances between the pack-soil, the water tank, and the water pump.
2. Minimize the distance between the heater and the fan.
3. Minimize the distance between connection points linking two components.
4. Maximize the distance between the LED and the heat sensor.
5. Maximize the distance between the heater and the heat sensor.
6. Maximize the lighting of the pack-soil by the LED.
7. Maximize the view of the pack-soil captured by the camera.

For LDOs, a weight is assigned to every one of them based on the DSMs mentioned before. Moreover, all distances and lengths use the 3D Euclidean distance. As for the 6th and 7th LDOs, the objectives use the 3D Euclidean distance between the line of sight (LOS) of two components as shown in Figure 5.3. The ideal LOS for the second component (C2) is the desired LOS presented in Figure 5.3. Hence, the distance between the desired LOS and C2 needs to be minimized.

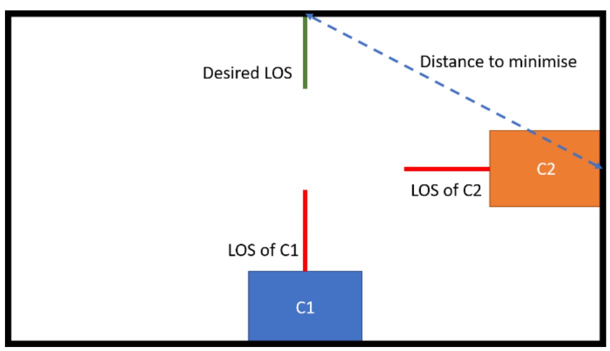


Figure 5.3. Objective based on the line of sight of components.

Second, the design of an autonomous greenhouse is necessarily subjected to a set of constraints coming from multiple sources. All of these constraints need to be respected to obtain a feasible solution. In this work, we consider three sources of constraints. The first source of constraints is based on the limited space allowed and the physical boundary of every component. The second source originates from the interaction of the components. The third and last source is the specifications of the greenhouse which would be given by a customer. In this work, the following constraints are considered:

1. The overall volume occupied by the components must be lower than the greenhouse volume.

2. The boundaries of a component must fit within the boundaries of the greenhouse.
3. The boundaries of a component cannot overlap another component's boundaries.
4. The field of view of the camera must be wide enough to capture the pack-soil.
5. The field of view of the LED must be wide enough to light the pack-soil.
6. The overall mass must be lower than a defined mass.
7. The overall energy consumption must be lower than a defined quantity of energy.
8. The voltage and current of each component cannot exceed the given thresholds.

The first three constraints come from the limited volume allowed. The fourth and fifth constraints are related to the interaction between components. The last three constraints are the product specifications that stem from customers' needs.

Finally, the optimization problem can be summarized as a single objective optimization problem

$$\begin{cases} Minimize \sum_{i=1}^{n} w_i O(\vec{x})_i \\ subjected\ to \\ CT_j\ for\ j = 1, \ldots, m \end{cases} \qquad (1)$$

where

$\vec{x}$ is the decision variable vector. It contains the parameters of every component as shown in Figure 4,

O ($\vec{x}_i$) is the ith objective of all the objectives (CSOs + LDOs),

W_i is the weight associated with the ith objectives of all the objectives and,

CT_j is the jth constraints.

The weight is 1 for all CSOs because they need to be optimized to overcome the constraints and to respect the product specification, but they are not the most important aspect of the optimization to accomplish the main function which is ensuring the growth of the plant. For LDOs, the weight can be found in the DSMs. For our formulation, n is the total number of objectives and m is the total number of constraints. In this work, n is equal to 10 and m is equal to 8.

5.6 Genetic Algorithm Implementation

Considering the optimization problem given in Section 5.5.2, the design of an autonomous greenhouse can be seen as a many-objective optimization problem (MaOP) like most real-life applications in engineering (Fleming, Purshouse, et al. 2005). Even though algorithms have been developed since 2003 (Hisao, Noritaka, et al. 2008, Li, Li et al. 2015, Huang, Zhang, et al. 2019) to solve MaOPs, we will use a single objective optimization to solve the layout design of an autonomous greenhouse for two main reasons. First, we are not searching for a Pareto front of designs. The main objective of this research is to thoroughly express a problem statement and its optimization formulation for autonomous greenhouse. Hence, we attempt to find one near-optimal design to validate the formulations based on the a priori preferences of the designers. Second, selecting an adequate algorithm for a given problem is not a trivial task as shown by the comparative study of Panerati and Beltrame (Panerati and Beltrame 2014). The study consists of evaluating the performance of 15 multi-objective design-space exploration algorithms on the optimization of three real-life applications using three performance metrics such as the average distance from the reference set. The comparison showed that no algorithm outperforms the others, however, general guidelines on the simulation time and size of the design space were found. Another example, (Saldanha, Soares, et al. 2017) defines the best algorithm between non-dominated sorting genetic algorithm-II, predator-prey and multi-objective particle swarm optimization. To achieve this, they needed to evaluate the results of each algorithm with performance metrics and a decision-making method called PROMETHEE. They found that the multi-objective particle swarm optimization was the best algorithm to design a shell-and-tube heat exchanger because of its robustness. Furthermore, the no-free lunch theorem (Wolpert and Macready 1997) informs us that a priori no algorithm outperforms another one in all optimization problems. Consequently, selecting an inadequate algorithm could output a poor set of Pareto front solutions which could erroneously lead us to believe that the formulation is inadequate. Hence, we decide to aggregate all the objectives using scalarizing functions (Marler and Arora 2004, Kaim, Cord, et al. 2018) into a single-objective optimization. Scalarizing functions can be used to articulate the preferences of the designers to find one solution from the Pareto front. Using this, we will better assess if our formulation can find a feasible and near-optimal solution.

Evolutionary algorithms are also known to be effective when it comes to solving single objective combinatorial optimization problems as explained in surveys concerning facility layout problems (Drira, Pierreval, et al. 2007, Moslemipour, Lee, et al. 2012, Ahmadi, Pishvaee, et al. 2017). Other domains also use evolutionary algorithms. Yu et al. (Yu, Yang, et al. 2007) used a parallel genetic implementation to optimize shopping routes of shoppers in terms of the shortest car-based route. The parallel genetic implementation is used to reduce the computational time by dividing the resources of the computer to execute the genetic algorithm operators. The implementation is tested on a case study in Dalian City, China. Zhao et al. (Zhao, Hsu, et al. 2016) implemented a GA to minimize the mental workload of human operators in a mixed-model assembly line based on many factors, such as the assembly complexity and operator experience. The motivation of this work is to reduce the errors resulting from human mental fatigue and to improve the efficiency of the assembly line. Ribas et al. (Ribas, Yamamoto, et al. 2013) combined hybrid micro-GA and mixed-integer linear programming to schedule and plan an oil pipeline network. The scheduling considered the management of the production and the operation, the inventory management, and the transportation of oil to name a few. The use of micro-GAs was to lower the computational time and resources while keeping good solutions. The developed algorithm is tested on Brazilian pipeline networks. Zhang and Zhang (Zhang and Zhang 2007) developed a GA to design a network based on the network partition problem. This optimization problem goal was to reduce the inter-network communication while managing the traffic distribution over sub-networks. The implemented GA considered the traffic matrix, the devices needed in a network as well as the current devices used in the industry. To validate the effectiveness of the GA, a simulation is carried out. Király and Abonyi (Király and Abonyi 2015) made a GA implementation to solve a multi-Traveling Salesmen Problem (mTSP). This work was greatly inspired by an industrial case study, where an electric and gas energy supplier needed to transfer materials from different sources to a specific location. This issue is an mTSP, which is a combinatorial optimization problem. Cheng et al. (Cheng, Gupta, et al. 2017) combined particle swarm optimization with a multitasking coevolution mechanism. The novelty resides in the multitasking coevolution mechanism, where two or more tasks have their own objective functions to optimize for an overall problem. The tasks were usually influencing each other, which led to a concurrent optimization problem. To validate the developed algorithm, the optimization of the productivity of a composite manufacturing is simulated. The two tasks to be optimized were the resin transfer molding and the injection/compression-liquid composite molding because these processes shared a part of the design space. Saleh and Chelouah (Saleh and Chelouah 2004) used an algorithm based on the GA to locate the position of an unknown point on Earth using satellite equipment. Such problems are called GPS surveying network, which is a variation of the problem from the classical survey network problem. The goal is to optimize the robustness of a GPS network based on the resources available to the network (e.g. cost, personnel, satellite, etc.). Guzmán-Cruz et al. (Guzmán-Cruz, Castañeda-Miranda, et al. 2009) compared five optimization algorithms to define which algorithm is better for the calibration of a specific greenhouse model. The five algorithms are genetic algorithms, evolutionary programming, evolutionary programming, least squares, and sequential quadratic programming. The greenhouse model is composed of 16 parameters and is used to estimate the internal humidity and temperature. For the calibration of the greenhouse model, evolutionary programming was the best choice. Elferchichi et al. (Elferchichi, Gharsallah, et al. 2009) used a weighted sum genetic algorithm to define the optimal inflow hydrograph to provide water to different farmers without emptying the reservoirs available. The objective function, as well as the constraints, were formulated using the water level of every reservoir. The weight associated with an objective function was defined with a sensitivity analysis. Based on the on-demand water quantity of the farmers, the GA was able to find an inflow hydrograph to supply the farmers without emptying the reservoirs for a specific period of time. Ushada and Murase (Ushada and Murase 2009) made the design of a customizable greening material combining three main tools. The first tool was the swarm modeling to set the design attributes. Then, the second tool was the desirability model to define the importance of design attributes based on the consumer mentality constraints. Finally, the third tool was the particle swarm optimization (PSO) to optimize the design of a greening material. The developed methodology was tested in a case study designing Sungoke moss. Utamina et al. (Utamima, Reiners, et al. 2019) developed a novel evolutionary algorithm called evolutionary hybrid neighborhood search (EHNS) which combines mutation-based neighborhood search and Tabu search algorithms. The first step of the EHNS loop was the mutation-based neighborhood search algorithm which uses the roulette wheel selection to pick individuals within the population. Then, mutation operators were applied to the chosen individuals. The second step was the replacement of the current individuals by the mutated individuals. If the best new individual is not better than the previous best individual, the Tabu search is chosen to make a local search around the best new individual. The last step is the setup of the next generation using the elitism and scramble principle. The EHNS was used to solve many agricultural problems from the literature. Moreover, the results of the algorithm were compared to other algorithms such as ant colony optimization, GA, etc.

Since the performance of different types of evolutionary algorithms greatly depends on the problem at hand as mentioned in (Youssef, Sait, et al. 2001, Ma, Simon, et al. 2013). We choose a GA considering it has a low complexity of its basic implementation. The global search effectiveness of the GA is also adequate for our problem since we do not consider an initial solution. There are seven main components in a GA: encoding/decoding, initial population, parent selection, crossover, mutation, survivor selection, and termination condition.

5.6.1 Encoding/Decoding & Initial Population

For the encoding, an individual is considered as a solution, which is represented by a vector of components. The vector is represented in Figure 5.4, where the CX are the components for X = 1, 2, 3…

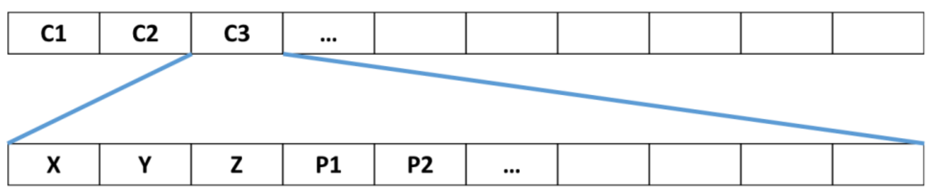


Figure 5.4 Decision variable vector: vector representation of components within a solution and the parameters within a component

Each component has a set of parameters represented by a vector as well. Two types of parameters can be given to a component. The first type is the common parameters that every component has. For example, the xyz position of the component within the greenhouse. The second type of parameter is specific to the component. For example, the energy consumption in terms of the voltage (V) and the current (A). The vector is represented in Figure 5.4, where XYZ is an example of common parameters and the PX are the specific parameters for X = 1,2,3…

The common parameters of components are:

1. The XYZ position of the component within the greenhouse.
2. The XYZ dimensions of the component (length, width, and height).
3. The mass of the component

The specific parameters considered are:

1. Energy consumption in terms of voltage and current for all the components except for the pack-soil and water tank.
2. Field of view of the component only for the camera and LED.
3. Location of the connection of the pipes on a given component only for the water tank, water pump, and pack-soil.

The initial population is randomly generated but must respect the set of constraints defined above. For each parameter, a random number is generated within a defined range of values. It is important to note that there is no code implemented to check if a solution appears more than once in the initial population. Since there are numerous combinations, it is less likely to find the same solution twice.

5.6.2 Parent Selection & Crossover

The parent selection implementation is based on the roulette wheel selection [44]. The principle of the roulette wheel selection is to divide a wheel considering the number of solutions and their overall objective function within a population. A fixed point is then randomly chosen on the circumference of the wheel. After spinning the wheel, the solution that stops in front of the fixed point is chosen as a parent. In other words, the parent selection is randomized as well. After using this method to choose two parents, the crossover function creates two children based on the parents' solution vector. A random crossover point is selected within the parents' solution vector. All components of the first parent after the crossover point are replaced by the components of the second parent and vice versa (see Figure 5.5). The two vectors created are considered as the children.

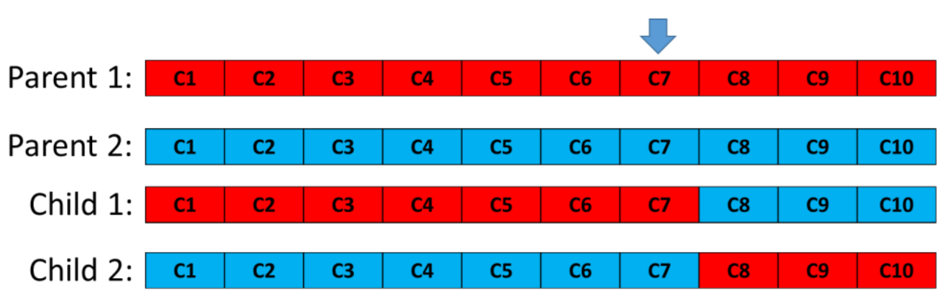


Figure 5.5 Crossover operation

5.6.3 Mutation

The mutation process for this implementation has two steps (Figure 5.6). The first step is a random selection of a component within a solution. The second step is a random selection of a parameter from the selected solution and a replacement of the parameter value by a random value within the range of values of this parameter.

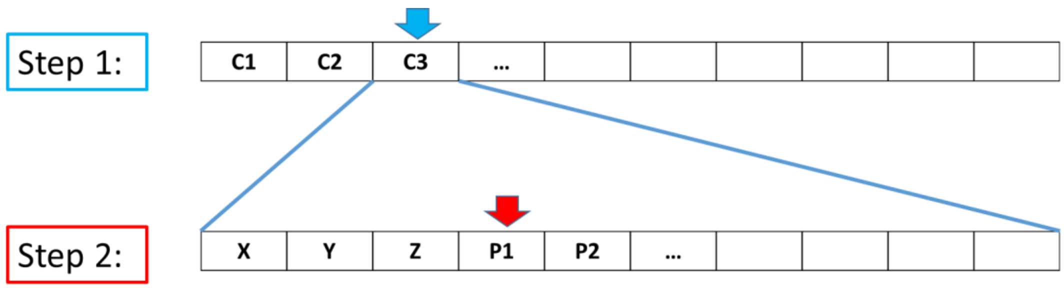


Figure 5.6 Mutation operation

5.6.4 Survivor Selection

The selection of individuals for the next generation only considers a single objective function which is the summation of the weighted CSOs and LDOs. Indeed, the maximum population size is fixed. This means that a new child, mutated or not, is compared to the rest of the population in terms of the single objective function when the population reaches its maximum. If the single objective function is better than the worst solution of the population, the child replaces this solution. Otherwise, the child is discarded. If the child is an infeasible solution, it is also discarded.

5.6.5 Termination Conditions

The terminal condition for this implementation of the GA is based on improvement through generations. If there is no improvement in the population after several generations, the GA is then stopped. However, the counter is reset to zero every time there is an improvement. The number of generations is defined through trials and errors.

Figure 5.7 shows the convergence graph of the implemented GA. Figure 5.7 was generated with the average of the overall objective of the best solution by generation of 10 GA runs with a terminal condition of 500 generations without improvement in the population. Furthermore, the 95% confidence interval by generation was computed to evaluate de uncertainty of the average. After

68224 generations, the 95% confidence interval drops down to 0 because the terminal condition does not set a finite number of generations. In other words, one GA run can have more generations than another one. Knowing this, the average values after 68224 generations are the values of the longest GA run. The different amount of generations per run can also explain the increase in the values of the 95% confidence interval between 10 000 and 45 000 generations. In this interval, some GA runs had already been terminated with a low objective value while other runs were still optimizing and had higher objective value.

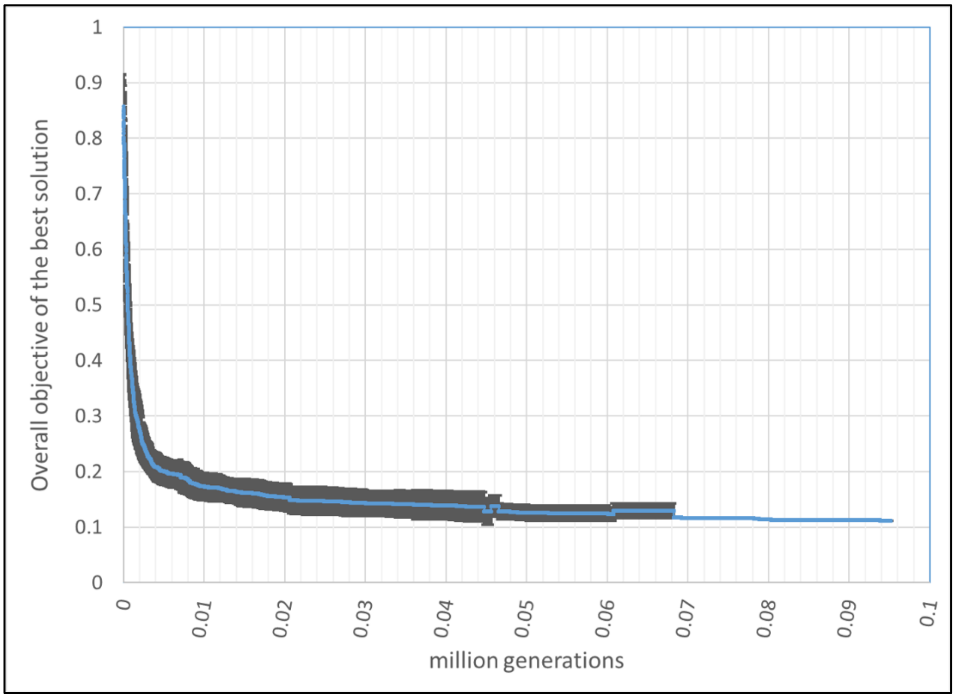


Figure 5.7 Convergence graph of the genetic algorithm

5.7 Results & Discussion

The simulation parameters have been chosen by the authors to design a general autonomous greenhouse. The parameters related to the GA operators are fine-tuned by trials and errors of the algorithm. For the parameters of components, the maximum and minimum values of each component come from technical datasheets from different manufacturers. For example, the range of values for the parameters of the water pumps was chosen based on different models of the component in the market (Alibaba 2019, Amazon 2019, Enabler 2019, good 2019, Systems 2019).

The components parameters and the size of the autonomous greenhouse were chosen based on different work related to the field of space biology such as Kibo, a small experiment module sent to the ISS to conduct experiments with the Arabidopsis plant (Yano, Kasahara, et al. 2013). Another example can be found in (Fu, Liu, et al. 2013) where a horn-type producer is designed as a life-support system. The study of plants in the space environment is getting a lot of attention for different reasons such as providing food and managing the gas cycle for astronauts (Häuplik-Meusburger, Peldszus, et al. 2011, Haeuplik-Meusburger, Paterson, et al. 2014).

The parameters of the simulation and the components are respectively given in Table 5.4 and Table 5.5.

Table 5.4 Simulation parameters (The parameters in a dark gray shading are GA parameters and those in light gray shading are product specifications)

Parameters	**Values**
Maximum population size	100
Number of unimproved generations to terminate the algorithm	500
Number of crossovers (if there are crossovers)	20
Probability of crossover	80%
Probability of mutation	5%
The probability of a random solution generated	80%
Maximum voltage for one component in a solution	9 V
Maximum current for one component in a solution	1000 mA
Max mass of a solution	1500 g
Maximum energy consumption of an individual	15 W
Greenhouse dimension	450 x 300 x 300 mm^3

Table 5.5 Parameters of components

Pack soil Dimensions range: 250 x 175 x 8 to 450 x 300 x 20 mm^3 Mass range: 300 to 425 g	Water tank Dimensions range: 50 x 50 x 50 to 100 x 100 x 100 mm^3 Mass range:150 to 1200 g
Heater Dimensions range: 30 x 30 x 5 to 80 x 80 x 10 mm^3 Mass range: 20 to 50 g Voltage range: 3.3 to 12 V	Heat sensor Dimensions range: 12 x 12 x 5 to 25 x 25 x 10 mm^3 Mass range: 0.1 to 1 g Voltage range: 1.7 to 3.6 V

Current range: 400 to 7000 mA	Current range: 0.01 to 0.02 mA
Camera Dimensions range: 10 x10 x 2.5 to 22 x 26 x 11 mm^3 Mass range: 0.1 to 6.4 g Voltage range: 1.7 to 5 V Current range: 50 to 160 mA Field of view: 60 to 90 °	LED Dimensions range: 40 x 40 x 1.84 to 100 x 100 x 2 mm^3 Mass range: 10 to 35 g Voltage range: 2.9 to 3.7 V Current range: 700 to 1400 mA Field of view: 60 to 90 °
Fan Dimensions range: 40 x40 x 10 to 80 x 80 x 25 mm^3 Mass range: 18.6 to 62.6 g Voltage range: 2 to 5.5 V Current range: 66 to 170 mA	Water Pump Dimensions range: 32 x 32 x 23 to 54 x54 x 46 mm^3 Mass range: 80 to 150 g Voltage range: 3 to 12 V Current range: 200 to 500 mA
PCB Dimensions range: 30 x 30 x 1 to 50 x 50 x 4 mm^3 Mass range: 5 to 10 g Voltage range: 3.3 to 6 V Current range: 5 to 50 mA	

It is important to explain the red, blue, and green lines in Figure 5.8 & Figure 5.9. The red lines are the line of sight of the component starting from the middle of the component. The blue and green lines make a cone, which represents the field of view of the component.

Figure 5.8 and Figure 5.9 show that the algorithm applied the guidelines given by the identification of the components and their interactions presented in Section 5.5. It is possible to see that the placement of components was mainly affected by the following dependencies: adverse effect and physical interface.

First, the GA avoids adverse effects by placing the heat sensor on the opposite side of the heater to avoid erroneous readings of the temperature. Erroneous readings of the temperature are also avoided because the LEDs are far from the heat sensor. Furthermore, every component does not prevent the LEDs and the camera to light and capture the pack-soil respectively except for the water tank. Indeed, the water tank is blocking a part of the field of view of the LEDs and the camera. However, most of the pack-soil is within these fields of view. Considering that one of the goals of the GA is also to maximize the size of the water tank and the pack-soil, it is more likely that a small portion of the field of views will be blocked by the water tank as can be seen in Figure 5.9.

Finally, the GA algorithm followed the guidelines from the physical interface by positioning the fan and heater side by side to improve the heat convection. The LEDs and the camera are positioned above the pack-soil, where they can maximize the lighting and the capturing of the pack-soil respectively. The pack-soil, water tank, and water pump are close to each other even if the pack-soil and water tank have important volumes. Hence, the algorithm minimizes the length of the pipe by minimizing the distance between these components.

Table 5.6 presents the values of the position and parameters of all components found with the GA. PX, PY, and PZ are the xyz position; DX, DY, and DZ are the xyz dimension; M, V, and C are respectively the mass, voltage, and current; the FOV is the field of view. Table 5.7 presents all connection points of the pipes on a given component.

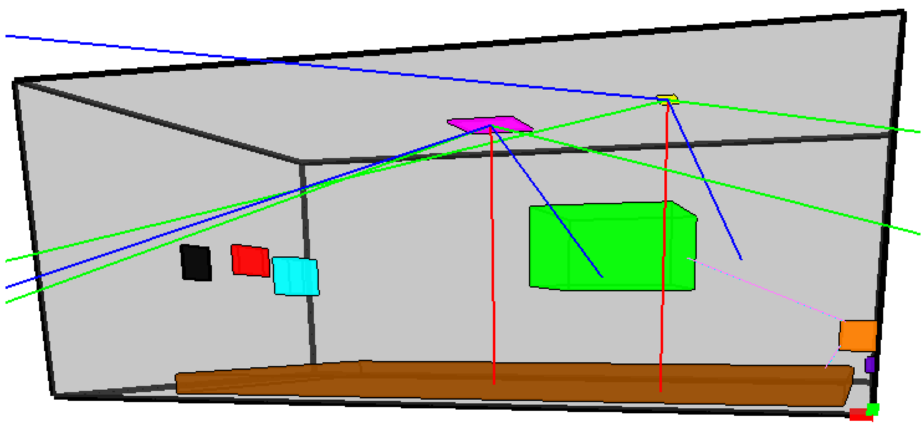


Figure 5.8 Layout optimization of the greenhouse (Side view)

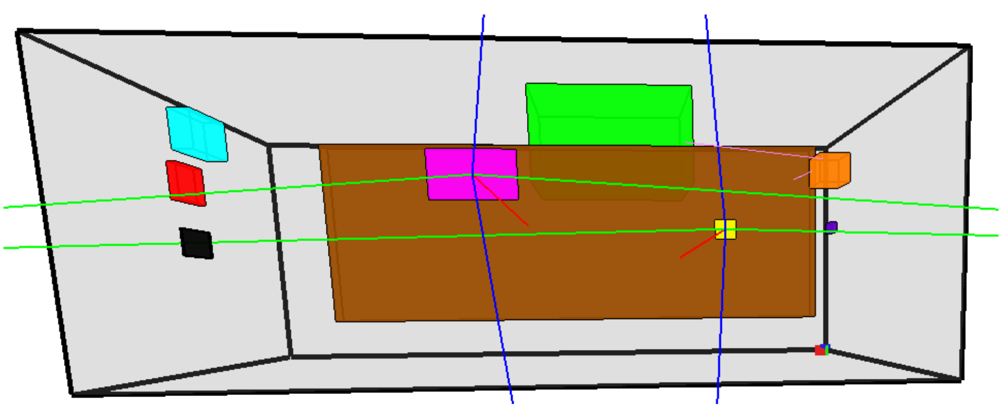


Figure 5.9 Layout optimization of the greenhouse (Top view)

Table 5.6 Numerical results of the simulation

Component	Colour	PX	PY	PZ	DX	DY	DZ	M	V	C	FOV
Heater	Red	450	138	232	35	40	5	21	3.6	783	-
Water tank	Green	132	119	300	92	96	98	207	-	-	-
Pack soil	Brown	405	0	52	391	244	20	307	-	-	-
Heat sensor	Purple	0	22	187	15	13	5	0.1	1.8	0.01	-
Camera	Yellow	120	300	133	10	16	2.6	0.4	1.7	50	73
Fan	Cyan	450	164	291	49	44	13	19.3	2.2	73	-
LED	Magenta	271	300	209	43	42	1.88	11	2.9	716	64
Water pump	Orange	0	43	272	34	38	23	85	3	233	-
PCB	Black	450	166	162	39	30	1	4.5	4.5	5	-

Table 5.7 Connection points of pipes on components

Pipes connection points	PX	PY	PZ
Water tank	132	156	260
Pack soil	32	20	250
Water pump #1	21	77	264
Water pump #2	23	48	255

It is possible to see that most of the sensors are close to their minimum values (see Table 5.5) in terms of volume, mass, voltage, and current. The only ones that are close to their maximum volume are the pack-soil and water tank, which are the ones that we wanted to maximize. The occupied volume is 2 844 316, 28 mm3 which is only 7% of the total volume of the autonomous greenhouse. The weight and energy consumption constraints are also respected since the obtained values are respectively 655.8 g and 5.86 W. The presented problem statement of the layout design of a greenhouse is flexible enough to take into consideration physics analysis. For example, gravity can be included in the placement of components to favor the placement of heavy items on the base of the greenhouse, which would make the system more stable. The minimization of the gravitational potential energy could be added as an objective function. Another interesting physical analysis would be a heat analysis, which could help place components and identify missing components in the heating system. For example, the heat analysis based on external and internal parameters could inform the designer if a cooling system is needed. The optimization problem could also include and optimize complexity metrics, which would help define the simplest robust solution.

One of the major issues in the presented algorithm is the parameters of the components are given in terms of the range of values hence it is, therefore, possible to get a solution for which components are not available in the market. Although it remains interesting to keep the parameters of components as a value range, a version of this algorithm could be developed to include a database containing different models of the same component. Furthermore, it would also be possible to change the shape of components from the rectangle approximation that is currently used, to a realistic shape of the component based on the dimensions given by the database. It would also give a more precise position of the line of sight of components similar to a camera. Another interesting future improvement would be to consider and allow combinations of certain components to minimize the space occupation of components. For example, instead of using a heater and a fan, a compact coaxial fan/heater could be used.

If the simulation parameters presented in Table 5.4 were to be changed, the output solution is most likely to change. However, the degree of impact on the solution has not been thoroughly studied. Hence, the GA parameters should be optimized to find the ideal simulation parameters using tuning algorithms (Eiben and Smit 2012, Montero, Riff, et al. 2018). For example, Ooi et al. (Ooi, Lim, et al. 2019) proposed a self-tune linear adaptive genetic algorithm that modifies the mutation probability rate and the population size based on the diversity of the population. Moreover, the product specifications should undergo a sensitivity analysis to evaluate the robustness of the design.

The optimization problem statement presents many objectives, which can be conflicting. For example, the optimization algorithm needs to minimize the overall volume and maximize the volume of the pack-soil. Currently, these conflicting objectives are implicitly considered by a weighted sum approach in the single-objective optimization. In future work, it is possible to explicitly consider them with a multi-objective optimization without assigning weights to them (Kalyanmoy 2001, Simon 2013). By using multi-objective optimization, a wider range of solutions would be available to the designer to choose from. As mentioned before, the layout design of an autonomous greenhouse can also be considered as a many-objective optimization problem that requires a more sophisticated algorithm to be solved (Hisao, Noritaka, et al. 2008, Li, Li et al. 2015, Huang, Zhang, et al. 2019). However, the selection process of the ideal algorithm to find the best range of solutions needs to be done carefully. Furthermore, the designer would still need to choose one solution among a Pareto-optimal set of solutions given by the multi-objective optimization algorithm using a posteriori approaches (Wang, Olhofer, et al. 2017, Yu, Jin, et al. 2019) which is not a trivial task as reported by (Torry-Smith, Achiche, et al. 2011, Mørkeberg Torry-Smith, Qamar, et al. 2012). Also, the presented algorithm rejects automatically a solution that does not respect constraints. Although it is one way to deal with constraints, other methods based on penalties or the dominating concept could also be used and might yield better results (Kalyanmoy 2001).

Finally, the output design of the presented algorithm should be validated by prototyping an autonomous greenhouse. By doing so, new phenomenon or interactions between components could emerge or the importance of one interaction over another could be identified. The algorithm could be then improved, and a new design might be output.

5.8 Conclusion

This paper presented how a general problem statement of the layout design of an autonomous greenhouse based on the placement of components is defined and translated into an optimization problem. A GA is then used to solve the optimization problem composed of multiple functional and spatial objective functions and constraints aggregated into a single overall objective using a weight sum approach. As mentioned before, the problem statement is flexible enough to include physical analysis, such as heat analysis if the designer wants to consider them. Although the proposed methodology presents weaknesses for the modeling of real components, we deemed that the approximations used in this paper are adequate to give a general idea of the layout design of an autonomous greenhouse and therefore a starting point for a designer. Indeed, we were able to make the design of an autonomous greenhouse for space biology applications where the volume of all the components is 2844, 32 cm3, which is 7% of the total volume. The greenhouse also consumes 5.86 W and weighs 655.8g, which respects the constraints of the problem statement. Furthermore, the GA can output a promising solution by compromising several spatial guidelines, such as keeping the heat sensor far from the LED and the heater. The GA is also able to find and evaluate an enormous amount of design variations in a reasonable time based on guidelines from the designer. Moreover, the GA can converge towards a near-optimal solution. The validation of the algorithm has yet to be done by prototyping the greenhouse and evaluate its capacity to ensure plant growth.

5.9 Acknowledgments

The authors would like to show their appreciation towards le Fonds de recherche du Québec – Nature et Technologies (FRQNT) and the Canadian Space Agency (CSA) for their financial support.

5.10 References

Abas, M. A. and M. Dahlui (2015). Development of greenhouse autonomous control system for Home Agriculture project. 2015 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA), IEEE.
Abas, M. A., M. Salman, M. Ridwan and M. Adzhar (2016). "Autonomous Irrigation Control through Rectifying Methods for Tropical Greenhouse Electronic System." International Journal of Simulation: Systems, Science and Technology 17(32).
Ahmadi, A., M. S. Pishvaee and M. R. Akbari Jokar (2017). "A survey on multi-floor facility layout problems." Computers & Industrial Engineering 107: 158-170.
Alibaba. (2019). "mini 6V DC Dosing pump Peristaltic Dosing Head For Aquarium Lab Analytical Water with Water Pipe Peristaltic pump." from https://www.alibaba.com/product-detail/mini-6V-DC-Dosing-pump-Peristaltic_60829374698.html?spm=a2700.7724857.normalList.199.1dca42bfUVp4XT.
Amazon. (2019). "Dicrey Micro Water Pump Mini Submersible Pump Fountain Pump Aquarium Water Pump Small DC Motor Water Pump 3V 4.5V 100L/H." from https://www.amazon.com/Dikley-Micro-Submersible-Fountain-Aquarium/dp/B0791FSR2S.
Azaza, M., C. Tanougast, E. Fabrizio and A. Mami (2016). "Smart greenhouse fuzzy logic based control system enhanced with wireless data monitoring." ISA transactions 61: 297-307.
Browning, T. R. (2016). "Design structure matrix extensions and innovations: a survey and new opportunities." IEEE Transactions on Engineering Management 63(1): 27-52.
Castañeda-Miranda, A. and V. M. Castaño (2017). "Smart frost control in greenhouses by neural networks models." Computers and Electronics in Agriculture 137: 102-114.
Cheng, M.-Y., A. Gupta, Y.-S. Ong and Z.-W. Ni (2017). "Coevolutionary multitasking for concurrent global optimization: With case studies in complex engineering design." Engineering Applications of Artificial Intelligence 64: 13-24.
Chouinard, U., S. Achiche and L. Baron (2019). "Integrating negative dependencies assessment during mechatronics conceptual design using fuzzy logic and quantitative graph theory." Mechatronics 59: 140-153.
Chouinard, U., S. Achiche, C. Leblond-Ménard and L. Baron (2017). Assessment of dependencies in mechatronics conceptual design of a quadcopter drone using linguistic fuzzy variables. DS 87-4 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 4: Design Methods and Tools, Vancouver, Canada, 21-25.08. 2017.

Drira, A., H. Pierreval and S. Hajri-Gabouj (2007). "Facility layout problems: A survey." Annual Reviews in Control 31(2): 255-267.
Eben-Chaime, M., A. Bechar and A. Baron (2011). "Economical evaluation of greenhouse layout design." International Journal of Production Economics 134(1): 246-254.
Eiben, A. E. and S. K. Smit (2012). Evolutionary Algorithm Parameters and Methods to Tune Them. Autonomous Search. Y. Hamadi, E. Monfroy and F. Saubion. Berlin, Heidelberg, Springer Berlin Heidelberg: 15-36.
Elferchichi, A., O. Gharsallah, I. Nouiri, F. Lebdi and N. Lamaddalena (2009). "The genetic algorithm approach for identifying the optimal operation of a multi-reservoirs on-demand irrigation system." Biosystems Engineering 102(3): 334-344.
Enabler, s. T. I. H. (2019). "6V Mini Water Pump." from https://www.seeedstudio.com/6V-Mini-Water-Pump-p-1945.html.
Ferentinos, K. P. and L. D. Albright (2005). "Optimal design of plant lighting system by genetic algorithms." Engineering Applications of Artificial Intelligence 18(4): 473-484.
Ferentinos, K. P., N. Katsoulas, A. Tzounis, T. Bartzanas and C. Kittas (2017). "Wireless sensor networks for greenhouse climate and plant condition assessment." Biosystems Engineering 153: 70-81.
Ferentinos, K. P. and T. A. Tsiligiridis (2007). "Adaptive design optimization of wireless sensor networks using genetic algorithms." Computer Networks 51(4): 1031-1051.
Ferentinos, K. P., T. A. Tsiligiridis and K. G. Arvanitis (2005). Energy optimization of wireless sensor networks for environmental measurements. CIMSA. 2005 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, 2005.
Fleming, P. J., R. C. Purshouse and R. J. Lygoe (2005). Many-Objective Optimization: An Engineering Design Perspective, Berlin, Heidelberg, Springer Berlin Heidelberg.
Fu, Y., H. Liu, L. Shao, M. Wang, Y. A. Berkovich, A. Erokhin and H. Liu (2013). "A high-performance ground-based prototype of horn-type sequential vegetable production facility for life support system in space." Advances in Space Research 52(1): 97-104.
Giroux, R., A. Berinstain, S. Braham, T. Graham, M. Bamsey, K. Boyd, M. Silver, A. Lussier-Desbiens, P. Lee, M. Boucher, K. Cowing and M. Dixon (2006). "Greenhouses in extreme environments: The Arthur Clarke Mars Greenhouse design and operation overview." Advances in Space Research 38(6): 1248-1259.
good, B. (2019). "12V DC Dosing Pump Peristaltic Pump For Aquarium Lab Analytical Water."
Goumopoulos, C. (2012). An Autonomous Wireless Sensor/Actuator Network for Precision Irrigation in Greenhouses. Smart Sensing Technology for Agriculture and Environmental Monitoring. S. C. Mukhopadhyay. Berlin, Heidelberg, Springer Berlin Heidelberg: 1-20.
Goumopoulos, C., B. O’Flynn and A. Kameas (2014). "Automated zone-specific irrigation with wireless sensor/actuator network and adaptable decision support." Computers and electronics in agriculture 105: 20-33.
Guzmán-Cruz, R., R. Castañeda-Miranda, J. J. García-Escalante, I. L. López-Cruz, A. Lara-Herrera and J. I. de la Rosa (2009). "Calibration of a greenhouse climate model using evolutionary algorithms." Biosystems Engineering 104(1): 135-142.
Haeuplik-Meusburger, S., C. Paterson, D. Schubert and P. Zabel (2014). "Greenhouses and their humanizing synergies." Acta Astronautica 96: 138-150.
Hahn, F. (2011). "Fuzzy controller decreases tomato cracking in greenhouses." Computers and Electronics in Agriculture 77(1): 21-27.

Häuplik-Meusburger, S., R. Peldszus and V. Holzgethan (2011). "Greenhouse design integration benefits for extended spaceflight." Acta Astronautica 68(1-2): 85-90.
Herrero, J. M., X. Blasco, M. Martínez, C. Ramos and J. Sanchis (2007). "Non-linear robust identification of a greenhouse model using multi-objective evolutionary algorithms." Biosystems Engineering 98(3): 335-346.
Hisao, I., T. Noritaka and N. Yusuke (2008). Evolutionary many-objective optimization: A short review. 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence).
Huang, W., Y. Zhang and L. Li (2019). "Survey on Multi-Objective Evolutionary Algorithms." Journal of Physics: Conference Series 1288: 012057.
Kaim, A., A. F. Cord and M. Volk (2018). "A review of multi-criteria optimization techniques for agricultural land use allocation." Environmental Modelling & Software 105: 79-93.
Kalyanmoy, D. (2001). Multi objective optimization using evolutionary algorithms, John Wiley and Sons.
Király, A. and J. Abonyi (2015). "Redesign of the supply of mobile mechanics based on a novel genetic optimization algorithm using Google Maps API." Engineering Applications of Artificial Intelligence 38: 122-130.
Koenig, R. and S. Schneider (2012). "Hierarchical structuring of layout problems in an interactive evolutionary layout system." Artificial Intelligence for Engineering Design, Analysis and Manufacturing 26(2): 129-142.
Komasilovs, V., E. Stalidzans, V. Osadcuks and M. Mednis (2013). Specification development of robotic system for pesticide spraying in greenhouse. 2013 IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI).
Krishna, K. L., J. Madhuri and K. Anuradha (2016). A ZigBee based energy efficient environmental monitoring alerting and controlling system. 2016 International Conference on Information Communication and Embedded Systems (ICICES), IEEE.
Li, B., J. Li, K. Tang and X. Yao (2015). "Many-objective evolutionary algorithms: A survey." ACM Computing Surveys (CSUR) 48(1): 1-35.
Ma, H., D. Simon, M. Fei and Z. Chen (2013). "On the equivalences and differences of evolutionary algorithms." Engineering Applications of Artificial Intelligence 26(10): 2397-2407.
Marler, R. T. and J. S. Arora (2004). "Survey of multi-objective optimization methods for engineering." Structural and Multidisciplinary Optimization 26(6): 369-395.
Matos, J., P. J. Gonçalves and P. M. Torres (2015). "An automatic mechanical system for hydroponics fodder production." Romanian Review Precision Mechanics, Optics & Mecatronics(47).
Mohebbi, A., L. Baron, S. Achiche and L. Birglen (2014). Trends in concurrent, multi-criteria and optimal design of mechatronic systems: A review. Innovative Design and Manufacturing (ICIDM), Proceedings of the 2014 International Conference on, IEEE.
Montero, E., M.-C. Riff and N. Rojas-Morales (2018). "Tuners review: How crucial are set-up values to find effective parameter values?" Engineering Applications of Artificial Intelligence 76: 108-118.
Mørkeberg Torry-Smith, J., A. Qamar, S. Achiche, J. Wikander, N. Henrik Mortensen and C. During (2012). "Challenges in Designing Mechatronic Systems." Journal of Mechanical Design 135(1).

Moslemipour, G., T. S. Lee and D. Rilling (2012). "A review of intelligent approaches for designing dynamic and robust layouts in flexible manufacturing systems." The International Journal of Advanced Manufacturing Technology 60(1-4): 11-27.
Nadal, A., P. Llorach-Massana, E. Cuerva, E. López-Capel, J. I. Montero, A. Josa, J. Rieradevall and M. Royapoor (2017). "Building-integrated rooftop greenhouses: An energy and environmental assessment in the mediterranean context." Applied energy 187: 338-351.
Ooi, C. S., M. H. Lim and M. S. Leong (2019). "Self-Tune Linear Adaptive-Genetic Algorithm for Feature Selection." IEEE Access 7: 138211-138232.
Pala, M., L. Mizenko, M. Mach and T. Reed (2014). Aeroponic greenhouse as an autonomous system using intelligent space for agriculture robotics. Robot Intelligence Technology and Applications 2, Springer: 83-93.
Panerati, J. and G. Beltrame (2014). "A comparative evaluation of multi-objective exploration algorithms for high-level design." ACM Trans. Des. Autom. Electron. Syst. 19(2): Article 15.
Paraforos, D. S. and H. W. Griepentrog (2013). "Multivariable greenhouse climate control using dynamic decoupling controllers." IFAC Proceedings Volumes 46(18): 305-310.
Pimmler, T. U. and S. D. Eppinger (1994). "Integration analysis of product decompositions."
Poulet, L. and O. Doule (2014). Greenhouse Automation, Illumination and Expansion Study for Mars Desert Research Station. 65th International Astronautical Congress, Toronto, Canada. International Astronautical Federation.
Rabago, F., E. de Santago and J. Moncada (2013). "Solar automated greenhouse." Advanced Materials Research 740: 198-202.
Ribas, P. C., L. Yamamoto, H. L. Polli, L. V. Arruda and F. Neves-Jr (2013). "A micro-genetic algorithm for multi-objective scheduling of a real world pipeline network." Engineering Applications of Artificial Intelligence 26(1): 302-313.
Romantchik, E., E. Ríos, E. Sánchez, I. López and J. R. Sánchez (2017). "Determination of energy to be supplied by photovoltaic systems for fan-pad systems in cooling process of greenhouses." Applied Thermal Engineering 114: 1161-1168.
Sabri, N., S. Aljunid, B. Ahmad, A. Yahya, R. Kamaruddin and M. Salim (2011). "Wireless sensor actor network based on fuzzy inference system for greenhouse climate control." Journal of Applied Sciences 11(17): 3104-3116.
Saldanha, W. H., G. L. Soares, T. M. Machado-Coelho, E. D. dos Santos and P. I. Ekel (2017). "Choosing the best evolutionary algorithm to optimize the multiobjective shell-and-tube heat exchanger design problem using PROMETHEE." Applied Thermal Engineering 127: 1049-1061.
Saleh, H. A. and R. Chelouah (2004). "The design of the global navigation satellite system surveying networks using genetic algorithms." Engineering Applications of Artificial Intelligence 17(1): 111-122.
Schubert, D., D. Quantius, J. Hauslage, L. Glasgow, F. Schroder and M. Dorn (2011). Advanced Greenhouse Modules for use within Planetary Habitats. 41st International Conference on Environmental Systems, American Institute of Aeronautics and Astronautics.
Simon, D. (2013). Evolutionary optimization algorithms, John Wiley & Sons.
Systems, T. F. (2019). "Miniature Peristaltic Pump." from https://www.takasago-fluidics.com/products/products_pump/peristaltic/.
Taura, T. and I. Nagasaka (1999). "Adaptive-growth-type 3D representation for configuration design." Artificial Intelligence for Engineering Design, Analysis and Manufacturing 13(3): 171-184.

Torry-Smith, J. M., N. H. Mortensen and S. Achiche (2014). "A proposal for a classification of product-related dependencies in development of mechatronic products." Research in Engineering Design 25(1): 53-74.
Torry-Smith, J. M. r., S. Achiche, N. H. Mortensen, A. Qamar, J. Wikander and C. During (2011). Mechatronic Design - Still a Considerable Challenge. ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.
Ushada, M. and H. Murase (2009). "Design of customisable greening material using swarm modelling." Biosystems Engineering 104(2): 169-183.
Utamima, A., T. Reiners and A. H. Ansaripoor (2019). "Optimisation of agricultural routing planning in field logistics with Evolutionary Hybrid Neighbourhood Search." Biosystems Engineering 184: 166-180.
Vera, M., R. Osorio-Comparán, A. Rienzo, M. Duarte-Mermoud and G. Lefranc (2017). Variables control of a modular greenhouse. 2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), IEEE.
Wang, H., M. Olhofer and Y. Jin (2017). "A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges." Complex & Intelligent Systems 3(4): 233-245.
Wolpert, D. H. and W. G. Macready (1997). "No free lunch theorems for optimization." IEEE Transactions on Evolutionary Computation 1(1): 67-82.
Xu, D. and H. Li (2008). Intelligent Greenhouse Control-System Based on Agent. 2008 International Conference on Intelligent Computation Technology and Automation (ICICTA).
Yano, S., H. Kasahara, D. Masuda, F. Tanigaki, T. Shimazu, H. Suzuki, I. Karahara, K. Soga, T. Hoson and I. Tayama (2013). "Improvements in and actual performance of the Plant Experiment Unit onboard Kibo, the Japanese experiment module on the international space station." Advances in Space Research 51(5): 780-788.
Youssef, H., S. M. Sait and H. Adiche (2001). "Evolutionary algorithms, simulated annealing and tabu search: a comparative study." Engineering Applications of Artificial Intelligence 14(2): 167-181.
Yu, B., Z. Yang and C. Cheng (2007). "Optimizing the distribution of shopping centers with parallel genetic algorithm." Engineering Applications of Artificial Intelligence 20(2): 215-223.
Yu, G., Y. Jin and M. Olhofer (2019). References or Preferences – Rethinking Many-objective Evolutionary Optimization. 2019 IEEE Congress on Evolutionary Computation (CEC).
Zhang, Q. and W. Zhang (2007). "Network partition for switched industrial Ethernet using genetic algorithm." Engineering Applications of Artificial Intelligence 20(1): 79-88.
Zhao, X., C.-Y. Hsu, P.-C. Chang and L. Li (2016). "A genetic algorithm for the multi-objective optimization of mixed-model assembly line based on the mental workload." Engineering Applications of Artificial Intelligence 47: 140-146.

CHAPTER 6 ARTICLE 2: OBJECTIVE REDUCTION USING AXIOMATIC DESIGN & PRODUCT-RELATED DEPENDENCIES: A LAYOUT SYNTHESIS OF AN AUTONOMOUS GREENHOUSE CASE STUDY

The article presented by Yann-Seing Law, Yuanchao Ma, Aurelian Vadean, Giovanni Beltrame, and Sofiane Achiche in this chapter is published in the conference *Proceedings of the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2020)*

Law-Kam Cio, Y, Ma, Y, Vadean, A, Beltrame, G, & Achiche, S. "Objective Reduction Using Axiomatic Design and Product-Related Dependencies: A Layout Synthesis of an Autonomous Greenhouse Case Study." *Proceedings of the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 11B: 46th Design Automation Conference (DAC)*. Virtual, Online. August 17–19, 2020. V11BT11A028. ASME. https://doi.org/10.1115/DETC2020-22397

6.1 Abstract

The many-objective optimization problem (MaOP) is defined as optimization with more than 3 objective functions. This high number of objectives makes comparing solutions more challenging. This holds true for design problems that are MaOPs by nature due to the inherent complexity and multifaceted nature of real-life applications. In the last decades, many strategies have attempted to overcome MaOPs such as removing objectives based on their impact on the optimization. However, from a design perspective, removing objectives could lead to an under optimal, unfeasible or unreliable design. Consequently, objective aggregation seems to be a better approach since objectives can be grouped based on design features controlled by the designers. The proposed methodology uses Axiomatic Design to decompose a system into subsystems or components, and Product-Related Dependencies Management to identify the dependencies between components and formulate the objectives. Then, these objectives are aggregated based on the subsystems found with the Axiomatic Design. The methodology, applied to the layout synthesis of an autonomous greenhouse, can trim down the number of objectives from 15 to 5. Then, using a modified non-

dominated sorting genetic algorithm-II (NSGA-II) combined with the objective aggregation, we were able to increase the number of "good" concepts found from 9 to 33 out of a total of 50 obtained designs.

6.2 Nomenclature

AD	Axiomatic Design
CA	Customer Attributes
FR	Functional Requirements
DP	Design Parameters
PV	Process Variables
PRDM	Product-Related Dependencies Management
NSGA-II	Non-dominated sorting genetic algorithm -II
MaOP	Many-objective optimization problems

6.3 INTRODUCTION

Engineering design problems referred to as design hereafter, often require optimization in many facets related to different phases of product development. For instance, layout design or product architecture is a step of the product design that includes the selection and positioning of the system's components or subsystems. During this phase, the designers need to evaluate all concepts generated from a catalog containing models of components or subsystems as well as their positioning. Nevertheless, designers are often overwhelmed by the number of concepts and cannot manually try all the possible combinations. This can often lead to a non-optimal product design. Fortunately, computer-aided design tools allow designers to search and optimize product concepts based on multiple objective functions. Among these methods, evolutionary algorithms have been proven to effectively solve complex engineering problems [1]. However, evolutionary computation still struggles to solve MaOPs [2-4], which represent most design problems [1]. MaOP is a problem that contains more than 3 objective functions to optimize. Furthermore, researchers in the field of design are faced with the additional burden to develop a proper methodology to evaluate concepts of complex systems [5, 6].

The goal of the work presented in this paper is to develop a methodology, from a design perspective, that can reduce the number of objective functions for a design problem to optimize simultaneously several concepts using evolutionary computation. In the rest of this section, a brief literature review about this problematic will be presented.

Even though MaOP work is an active topic in research, a sizable effort is deployed in the development of methods to reduce the objective space for multi-objective optimization [7, 8]. Among the first to tackle the objective reduction issue in the field of the multi-objective evolutionary algorithm is Deb and Saxena [9]. Deb & Saxena first proposed a principal component analysis (PCA) to reduce the objective set by eliminating redundant objectives. Inspired by the idea, Brockhoff and Zitzler [10] improved the principle by investigating the effect of omitting or including objectives and evaluating the level of conflict between objectives. Saxena et al. [11] pushed further their PCA approach by proposing a framework that aims at handling linear and nonlinear objectives and reducing the noise of the input data as well as the number of parameters by adapting concepts from machine learning.

So far, most works lean toward automatically reducing objectives by omitting redundant and/or non-conflicting objectives. These techniques are adequate to solve most optimization problems. However, for design problems, omitting a redundant or non-conflicting objective can have an important effect on the reliability or functionality of the system which can lead to the design of poor performing systems or not meeting all the defined needs. Consequently, these strategies seem ill-adapted for design.

Alternatively, Bandyopadhyay and Mukherjee [12] developed a two-phase algorithm based on differential evolution multi-objective optimization. This strategy emphasizes dealing with conflicting objectives but does not reduce the number of objectives. Hence, the “curse of dimensionality” persists. De Freitas et al. [13, 14] use the aggregation tree method to evaluate the degree of conflict and harmony between objectives to find out if combining objectives would be advantageous. The whole strategy is adequate to find the degree of conflict between objectives; however, the time complexity of the aggregation tree is O (nm^3) where n and m are the numbers of solutions and objectives respectively. Hence, it is possible to see that the aggregation tree can slow down the algorithm.

In design, the problem of reducing the number of objectives is not generally focused upon. Instead, the existing tools help differentiate a good concept from a bad one. Then, they attempt to propose the most optimal concept for a specific problem [15, 16]. To do so, different methodologies and performance indices have been established to assign a score to a concept. Additionally, for designers, it is common to try to reduce the design space. Usually, they aim at finding potential regions where it would be possible to locate the global optimum. For example, Wang and Shan [17] propose a design space reduction method using rough sets, multi-objective optimization, and robust design optimization. First, samples of the design space are acquired and analyzed to find an attractive subspace where the optimization is carried out. Also, Wang and Simpson [18] work on reducing the design space using a combination of a kriging model and fuzzy c-means clustering. Albers et al. [19] give guidelines to ease the handling of objectives during the design process. One of the guidelines concerns the reduction of the objectives by carefully selecting them to define a set of relevant objectives. Hence, they do not explicitly aim at reducing the number of objectives of the specific set. Chen et al. [20] evaluate the product architecture using a multi-criteria evaluation based on TOPSIS. It is important to note that the criteria and the weights associated with each criterion are problem-specific and defined by the designer. Therefore, the designer needs to find a way to reduce the objectives.

There are a few relevant works in the literature on the combination of Axiomatic Design (AD) and multi-optimization design. Hirani and Suh [21] optimized the design of journal bearings using the Multi-Objective Genetic Algorithm (MOGA) and AD approaches. AD is used to analyze the redundancies and couplings of the original journal bearing system. Therefore, after using sensitivity analysis based on the results of AD analysis, the designer came up with uncoupled bearing objective functions for MOGA and conduct to an optimal design. Wu et al. [22] proposed a new approach of multi-objective optimization respecting the independent axiom of AD. They regrouped the important design variables into several specific objective functions representing the Design Parameters (DPs) to have an uncoupled design matrix between Functional Requirements (FRs) and Design Parameters (DPs). Then the optimization of every objective function is achieved through the function dependencies table and an optimal design of disc brakes has been solved to show the validity of this new approach.

In summary, the reduction of the number of objectives is used to lower the number of objectives of a specific set to ease the optimization. The methods that seem to be trending are either to omit objectives or to aggregate them. As mentioned before, the aggregation strategy will be considered in this paper since it seems to be more adapted for design problems.

Considering the works of Hirani and Suh [21] as well as Wu et al. [22], it is possible to observe that the strength of AD is to decompose a system into a minimum set of subsystems by avoiding redundant information. Furthermore, AD attempts to minimize the couplings between every subsystem of the set. Knowing this, AD is suited to identify design parameters of different levels of abstraction which is the first step of the methodology presented in this paper. The second step consists of analyzing the dependencies between design parameters at a specific level of abstraction to find a specific set of objectives. This can be carried out by the Product-Related Dependencies Management (PRDM) [23] which is used to identify the interactions between components of a system. The last step will be to combine AD and PRDM to aggregate the objectives by subsystems which will lead to the reduction of the number of objectives. The layout synthesis of an autonomous greenhouse will be the case study of this paper to test the proposed methodology and denote its limitations.

To the best of the authors' knowledge, an objective reduction strategy based on Axiomatic Design and Product-Related Dependencies has not been done yet. Knowing the current limitations of the evolutionary algorithm on solving MaOPs, the proposed methodology uses design tools to overcome these limitations.

The rest of this work will be divided as follows: In Section 6.4, the description of the methodology will be done as well as an introduction of our case study: designing an autonomous greenhouse. In Section 6.5, the results will be presented and discussed. Finally, we will conclude this work in Section 6.6.

6.4 MATERIALS AND METHODS

The overall methodology starts with the identification of the main subsystems and components by respecting both two axioms of Axiomatic Design (AD). Then the identification of components' dependencies is carried out with the Product-Related Dependencies Management (PRDM) developed by Torry-Smith et al. [23]. Based on these dependencies, the relations between components are highlighted and used to formulate objective functions that need to be optimized. Then, the objective aggregation is defined based on the identified subsystems. Finally, the aggregated objectives would be embedded in an evolutionary algorithm to find a final concept. Figure 6.1 illustrates the general idea of the methodology. Every step of the methodology will be detailed in the rest of this section.

6.4.1 Identification of the main subsystems and components

To get a well-defined system decomposition with an appropriate set of design components, a systematic design approach needs to be carefully carried out before the following design phase. Consequently, AD is considered in this article for the first phase of the design process because of its specific hierarchical mapping strategy highlighting the main subsystems and components.

As shown in Figure 6.2, there are four domains in AD: Customer Attributes (CA), Functional Requirements (FR), Design Parameters (DP), and Process Variables (PV). PVs would not be regarded in this article since we are focusing on the conceptual design problem. The first step of AD is to express the CAs into the first level of FRs. Furthermore, designers could first derive fitting subsystems and then their pertinent components from the consecutive mapping between FRs and DPs, as demonstrated in Figure 6.3. By respecting both the Independence Axiom that is to maintain the independence of functional requirements and the Information Axiom that is to minimize the information content of the design, AD will guide designers to scientific concepts with less coupling (uncoupled or decoupled in the sense of AD). Concepts with low coupling should help identify the minimum number of subsystems and components needed to accomplish the task of the system.

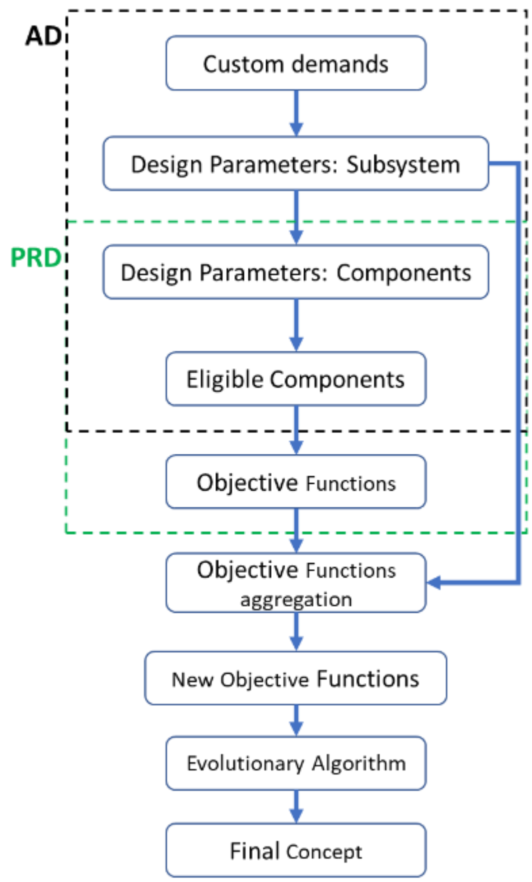


Figure 6.1 Overview of the methodology

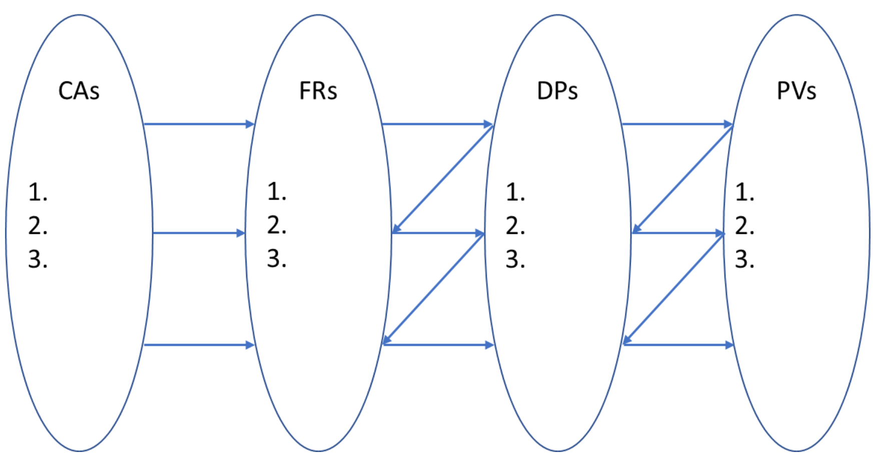


Figure 6.2 Summary of the axiomatic design (1) custom attributes (CAs) (2) function requirements (FRs) (3) design parameters (DPs) (4) process variables (PVs)

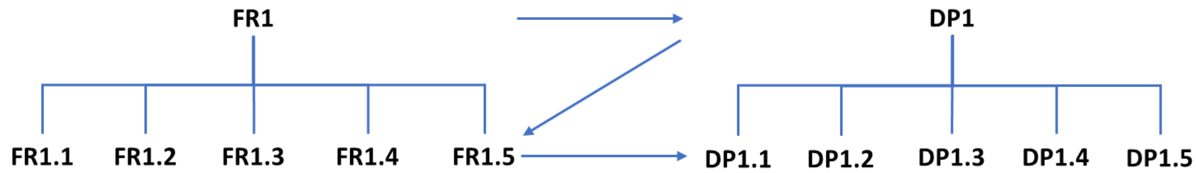


Figure 6.3 Zigzagging through FRs and DPs on different levels of abstraction

6.4.2 Identification of components' dependencies and objective functions

The output of the AD is several sets of concepts with low couplings between components or subsystems. Based on this, the goal is to establish the dependencies between these components to ensure that the system works properly. These dependencies are found using three generic terms to describe the products as well as their interactions. The first term is the function and is defined as the tasks that need to be accomplished by components, subsystems, or systems. Then, the second term is called property and refers to a property of the product such as the dimensions. The last term is the means referring to what is used to fulfill a function. Using this methodology, it is possible to identify dependencies between components, subsystems, or systems depending on the chosen level of abstraction [16, 23]. In this work, PRDM will be used to identify dependencies between components. These dependencies will then be translated into objectives from a design point of view since they are specific to a problem. We proposed the methodology presented in Table 6.1 to do the translation.

Table 6.1 Translation from dependencies to objective functions

Number	Name of dependency	Maximize or minimize	Objective function	Decision variables
1	Dependency 1	Maximize	Equation 1	V1,V2
2	Dependency 2	Minimize	Equation 2	V2
3	Dependency 3	Maximize	Equation 3	V1,V3
...	...	...	...	...

6.4.3 Objectives aggregation and weight's assignment

At this point, AD has allowed the designers to identify the subsystems and components with the minimized couplings. Then, PRDM was used to define the dependencies between components which were translated into objectives by the designers. Now, the aggregation process is accomplished by linking the objectives with the subsystems. Every subsystem contains components that are mandatory to fulfill its tasks. Knowing this, all the objectives that are needed to ensure the success of the subsystem's tasks will be clustered together. Within a subsystem, the assignment of objectives' weights can be done to prioritize dependencies with the most important influence on the functional requirements of the systems. Even though there are many ways to assign weights to objectives, in this work, the objectives' weights are set based on the designers' experience or by surveying experts in the field of design. Moreover, as weights tend to be problem-specific [15] a fine-tuning through trials and errors is performed. First, the objectives' weights are assigned by the designers by prioritizing the most important dependencies. Then, iteratively, the results of a short run of the algorithm are analyzed for fine-tuning the weights.

6.4.4 Choosing the evolutionary algorithm

The choice of the evolutionary algorithm depends on the problem to be optimized. In other words, to know which algorithm will output the best solution set, a comparison study must be made for a specific problem. However, such a comparison will not be made in this paper since it is not its main objective. Here, an adaptation of the second version of the non-sorting genetic algorithm (NSGA-II) [24] was chosen because we believe it can show how much impact the proposed objective aggregation has. Indeed, if we were to choose a many-objective optimization algorithm such as NSGA-III or MOEA/D, the comparison between the optimization with and without the proposed objective aggregation would be difficult to do since these algorithms are designed to solve MaOPs. However, NSGA-II is not as efficient. Consequently, the proposed objective aggregations could, for a particular design problem, make the NSGA-II efficient again while avoiding the use of a more sophisticated algorithm. The adaptation of the NSGA-II is made by the authors of this paper and is meant to accelerate the finding of concepts that respect constraints, to better set a termination condition while better explore the design space. The first operator that improves searching and finding new concepts respecting constraints is a directed mutation operator. This operator happens when at least one individual in the population does not respect all the constraints. The main idea is to select a random individual in the population and to apply a mutation operator on one of the decision variables used to evaluate if the constraints are respected. The termination condition is set by the user who decides on a number of iterations. After an iteration of the algorithm, if an individual with a better score is found, the iterations done so far will be reset to zero. The last operator added to the NSGA-II algorithm affects the probability of the mutation. The probability is affected by the termination condition. It is defined as follows:

$$p_m = 100\% - \left(\mathrm{X}\% + \frac{\mathrm{NIWI}}{(\mathrm{MNI}-\mathrm{NIMP})} * \Delta\%\right) \quad (1)$$

with

X%: initial probability

NIWI: number of iterations without improvement

MNI: maximum number of iterations

NIMP: number of iterations at lowest probability

Δ%: maximum added probability

By doing so, the exploration is more prominent when an improvement has been found and becomes less important when there is no improvement for many iterations.

6.4.5 Case study: autonomous greenhouse

As a case study, the layout design of an autonomous greenhouse is chosen since it is a mechatronic system simple enough to identify and understand its subsystems, components, and dependencies and complex enough to highlight the problematic of most design problems which is the presence of a high number of objectives.

First, the AD of an autonomous greenhouse has been done before by the authors [25]. Table 6.2 reports these results. To be more concise, only the design parameters to be evaluated in the next stage are listed in this table. The parameter FR1 responds to the task “conducting a botanic experience” and the parameter DP1 corresponds to the incubator system to be designed. Subsequently, the FR1 is decomposed into four parameters of the second hierarchy while respecting the necessary conditions of the experiment. The DP1 is therefore also decomposed into four design parameters, represented by four subsystems while satisfying the new FR1.1-FR1.4. The design process continues by zigzagging between the FRs and the DPs. Once the DPs can be chosen by a specific component product, we can stop zigzagging and start to define the related objective functions for layout design.

9 main components were found with AD. The water tank and the water pump are used to contain and distribute respectively the nutritious solution to the seeds located in the pack-soil. The LED is the source of light to ensure the growth of the plants. Furthermore, the heater, the fan, and the heat sensor regulate the temperature of the environment. All the mentioned components so far are essential components for the survival and growth of the plants. The camera and the PCB are needed to monitor the health of the plants and the environment gas concentrations. Indeed, the camera allows to follow the visual aspect of the plant and the PCB gives us data about the CO2, O2, and other gas concentrations. This information can then be used for different scientific purposes such as plant behavior modeling.

In the author's previous work [26], a generic formulation of the layout design optimization for an autonomous greenhouse is given. Here, we will only report the objectives found using PRDM. To these, we also add another one which will be mentioned as the last point of the list in Table 6.3. In Table 6.3, DX, DY, and DZ are the dimensions of the components. Then, M, V, and A are the mass, voltage, and current respectively. Furthermore, PX, PY, PZ are the Cartesian coordinates of the components in the greenhouse reference frame. However, for objectives 13 and 14, the PX, PY and PZ are different coordinates in the greenhouse reference frame. These coordinates are explained in [26].

It is possible to see that the number of objectives (15) is too high for an evolutionary algorithm. Therefore, the chances of poor convergence or divergence are high. In Figure 6.4, we show the objectives and their corresponding components. Using this representation along with the relation between subsystems and components shown in Table 6.3, it is possible to aggregate objectives by subsystems as illustrated in Figure 6.5. The aggregation of the old objectives is done by a simple weight sum approach. As mentioned before the weights are defined by the designer as well as trials and errors. In our case, we assign a weight of 2 to objectives 5,6,7,11,12, and 14 and a weight of 1 to the rest of the objectives. It is also possible to see that objectives 3 & 4 are not taken into account during the aggregation phase. The reason is simply that they are considered as constraints defined by inequalities since the lowest mass and energy consumption are not necessary. We just want to get these values under their respective threshold. Equation 2 summarizes the optimization problem using the aggregation method. The O_i are the objective of Table 6.3 and the ω_i represent the weights of corresponding objectives. It is also important to mention that the weight of the objectives that need to be maximized shown in Table 3 (e.g., ω_2), are negative so that Equation 2 can be considered a minimization problem.

$$\begin{cases} Minimize\ \omega_5 O_5 + \omega_6 O_6 + \omega_7 O_7 + \omega_9 O_9 + \omega_{10} O_{10} \\ Minimize\ \omega_{13} O_{13} \\ Minimize\ \omega_8 O_8 + \omega_{11} O_{11} + \omega_{12} O_{12} \\ Minimize\ \omega_{14} O_{14} + \omega_{15} O_{15} \\ Minimize\ \omega_1 O_1 + \omega_2 O_2 \\ Subjected\ to \\ \text{No} - \text{overlapping constraints} \\ \text{Boundaries constraints} \\ \text{Field of view of LED and Camera constraints} \end{cases} \tag{2}$$

Table 6.2 Axiomatic design for an autonomous greenhouse

FRs		DPs	
FR1	Provide an environment conducive for the experience	DP1	Autonomous greenhouse
FR1.1	Provide necessary conditions for plant growth	DP1.1	Cultivation system
FR1.1.1	Artificial soil	DP1.1.1	Packsoil
FR1.1.2	Illumination	DP1.1.2	LEDs
FR1.2	Control the temperature at a specific value	DP1.2	Temperature control system
FR1.2.1	Active control	DP1.2.1	Thermo-heater
FR1.2.2	Measure local temperature	DP1.2.2	Temperature sensors
FR1.3	Circulate nutrient solution	DP1.3	Irrigation system
FR1.3.1	Provide for nutrient solution	DP1.3.1	Hydraulic pump
FR1.3.2	Store nutrient solution	DP1.3.2	Water tank
FR1.3.3	Deliver nutrient solution	DP1.3.3	Silicone tubes
FR1.3.4	Circulate the air	DP1.3.4	Fans
FR1.4	Check the growth state	DP1.4	Inspection system
FR1.4.1	Assemble the sensors	DP1.4.1	PCB
FR1.4.2	Check visually different state of growth	DP1.4.2	Camera

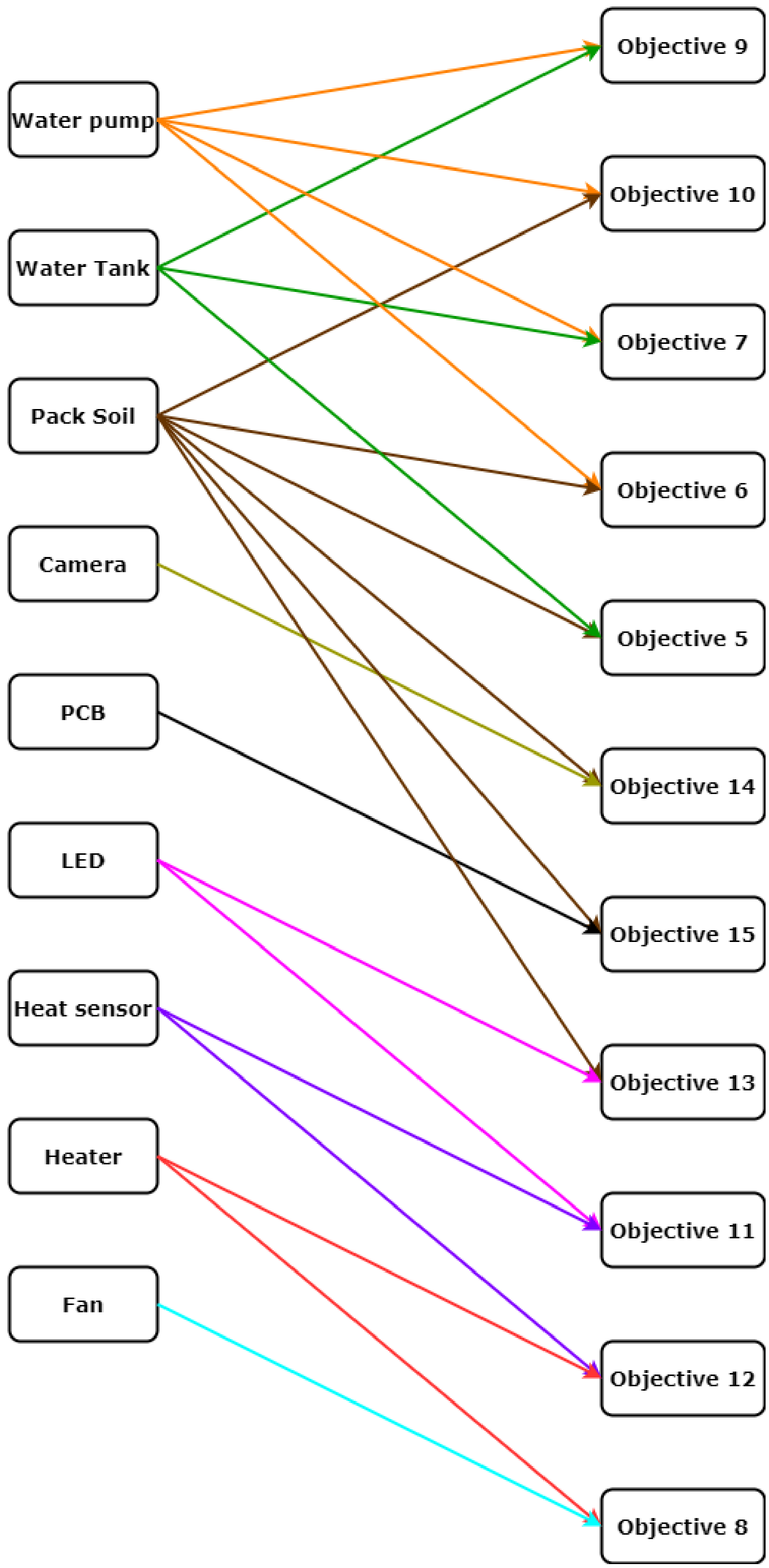


Figure 6.4 Relation between components and objectives

Table 6.3 List of dependencies and their corresponding objectives from [26]

Number	Name of dependency	Max or min	Objective function	Decision variables
1	Total volume	Min	Σ(DX*DY*DZ)	DX,DY,DZ
2	Volume of pack soil and water tank	Max	Σ(DX*DY*DZ)	DX,DY,DZ
3	Mass	Min	Σ M	M
4	Energy consumption	Min	Σ(V*A)	V,A
5	Distance between pack soil and water tank	Min	Eucledian Dist	PX,PY,PZ
6	Distance between pack soil and water pump	Min	Eucledian Dist	PX,PY,PZ
7	Distance between water tank and water pump	Min	Eucledian Dist	PX,PY,PZ
8	Distance between heater and fan	Min	Eucledian Dist	PX,PY,PZ
9	Distance between connection points of water tank and water pump	Min	Eucledian Dist	PX,PY,PZ
10	Distance between connection points of pack soil and water pump	Min	Eucledian Dist	PX,PY,PZ
11	Distance between LED and heat sensor	Max	Eucledian Dist	PX,PY,PZ
12	Distance between heater and heat sensor	Max	Eucledian Dist	PX,PY,PZ
13*	Lighting of the pack soil with the LEDs	Max	Eucledian Dist	PX,PY,PZ
14*	Capturing of the pack soil with the camera	Max	Eucledian Dist	PX,PY,PZ
15	Distance between PCB and pack soil	Max	Eucledian Dist	PX,PY,PZ

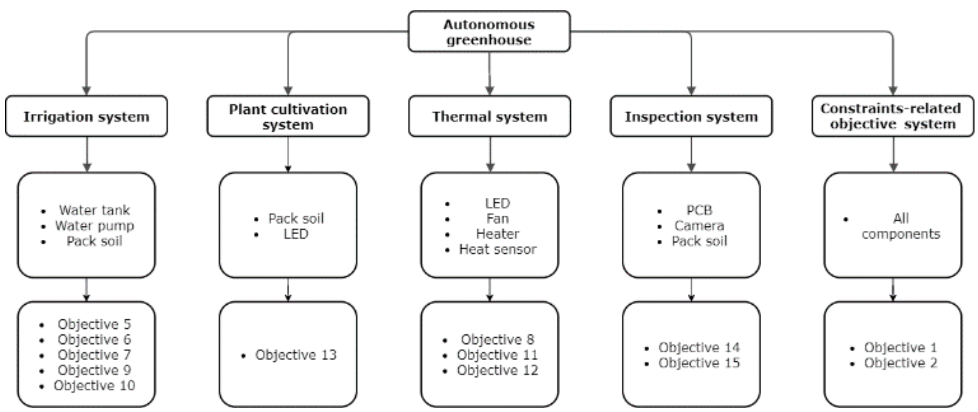


Figure 6.5 Objective aggregation based on subsystems

6.5 RESULTS AND DISCUSSION

In this section, we will present the output of the solution set from running our modified version of NSGA-II with and without the objective aggregation presented above. The population size is 50 and the run stop after 1500 consecutive iterations without improvements. Each solution presented in Figure 6.6, Figure 6.7, and Figure 6.8 were generated using the decision variables PX, PY, PZ, DX, DY, and DZ to position the components.

Table 6.4 Association color to components

Color	Component
Green	Water tank
Orange	Water pump
Brown	Pack soil
Yellow	Camera
Black	PCB
Magenta	LEDs
Purple	Heat sensor
Red	Heater
Turquoise	Fan
Turquoise/Red	Tubes

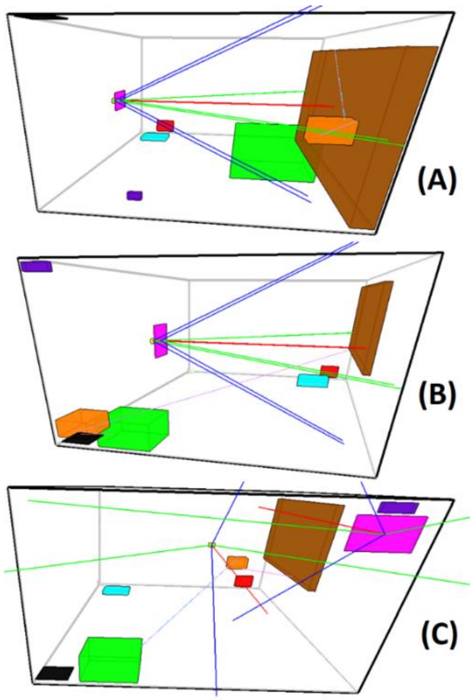
To analyze the results of Figure 6.6, Figure 6.7, and Figure 6.8, it is important to understand the color code associated with the different components shown in Table 6.4. Furthermore, the camera and LEDs have five axes. The axis in red is the line of sight of the component and the four other axes in green or blue represent the field of view of the component.

From an optimization point of view, the results are shown in Figure 6.6, Figure 6.7, and Figure 6.8 are all non-dominated solutions found from their respective run. However, from a design perspective, their design quality is not the same. Normally, both points of view do not see eye-to-eye on the value of output since the current optimization algorithm cannot fully consider the designer's experience and knowledge. As a result, the output of the algorithm is a collection of concepts that still need to be filtered by the designer. Hence, the rating of "good", "fair" and "mediocre" solutions are assigned by the authors. For each figure, we will explain how we rated the solution.

Figure 6.6 presents three different solutions without the objective aggregation. From a design point of view, it is possible to see that solution A is an example of a good solution. Indeed, the camera and LEDs are capturing and illuminating most of the pack-soil respectively. The heat sensor is far from the two different heat sources, the heater and the LEDs. The water tank, water pump, and pack soil are relatively close to one another which means the length of the tubes will be short. The pack-soil is large enough to give adequate growth space for the plants. The only downside is that the water tank is partially obstructing the field of view of the camera and LEDs.

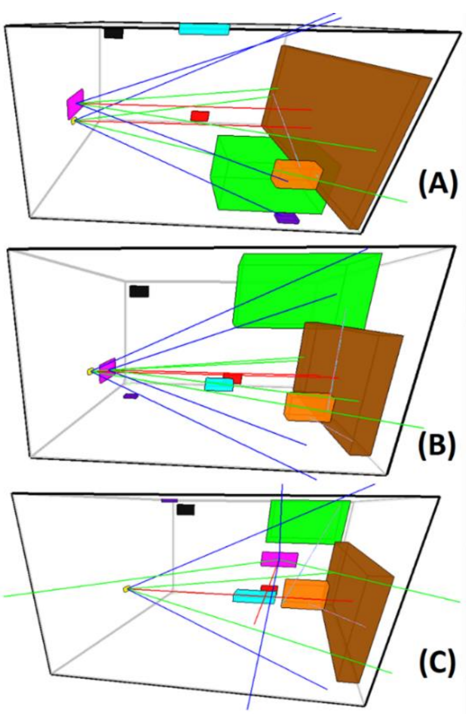
Then, solution B is an example of a “fair” solution. Indeed, most of the components’ location seems to be well placed except for one or two that prevent the concept to be as good. Here, the water tank is too small and far from the pack-soil. Moreover, the water pump is also far from the pack-soil. In this situation, moving the water tank and pump closer to the pack-soil is a posteriori modification that can be done since there is an empty space right beside the pack-soil. However, the repositioning of components might not always be as easy considering that their repositioning can affect the rest of the components. Finally, solution C is an example of a “mediocre” concept. The camera and the LED are not fulfilling their function properly since they are respectively filming and illuminating the side of the pack-soil. Then, the heat sensor is close to the LEDs which means that erroneous reading of the temperature will happen. Then, the water tank is once again too small and far from the pack-soil. The presence of “good”, “fair” and “mediocre” concepts is found without using objective aggregation. However, there are more concepts labeled as mediocre. Out of 50 automatically generated solutions, there were 30 mediocre concepts, 11 fair concepts, and only 9 good concepts. Using the proposed objective aggregation, we were able to increase the number of good and reduce the number of mediocre concepts respectively. Indeed, Figure 7 presents three examples of good, fair, and mediocre concepts using objective aggregation. Using the same arguments mentioned above, it is possible to see that solution A is a good concept; solution B is a fair concept and solution C is a mediocre concept. However, out of the 50 outputted solutions, there are 8 mediocre concepts, 9 fair concepts, and 33 good concepts. Hence, we can find more different good concepts. Figure 8 shows three different good concepts. Hence, the objective aggregation developed in this paper allowed the reduction of the number of objectives from 15 to 5 which lead to an increase in the number of good concepts generated from 9 to 33.

Considering that without the aggregation method, it was still possible to find 9 good concepts, why would one look for more good concepts? The answer lies in the design perspective. In the case study, we did the layout design of an autonomous greenhouse, however, the product design does not stop here. From the outputted design, the designers need to select a finite number of elite concepts for the following phase. These concepts will go to a detailed design phase and some of them might be prototyped to make sure that at least one of them is indeed feasible and performant. This means that the designer needs a selection of good concepts that are different enough from one another. Consequently, if there are 33 good concepts instead of 9, the designer has a higher chance to select a concept that would respect the unexpected constraints of the next phase of the product development.



Solution A	
Pros	Cons
LED and camera illuminate and film the front of the pack soil	
Heat sensor is far from heat sources	
Water tank, water pump and pack soil are close	
Solution B	
Pros	Cons
LED and camera illuminate and film the front of the pack soil	Small water tank
Heat sensor is far from heat sources	Water pump and water tank are far from pack soil
	Small pack soil
Solution C	
Pros	Cons
Heater is far from heat sensor	LED and camera illuminate and film the side of the Pack soil
Water pump is close to pack soil	Heat sensor is close to LED
	Small water tank
	Small pack soil
	Water tank is far from pack soil

Figure 6.6 Solutions without objective aggregation: (a) good (b) fair (c) mediocre



Solution A	
Pros	Cons
LED and camera illuminate and film the front of the pack soil	
Heat sensor is far from heat sources	
Water tank, water pump and pack soil are close	
Solution B	
Pros	Cons
LED and camera illuminate and film the front of the pack soil	Heat sensor is close to LED
Heat sensor is far from heater	
Water tank, water pump and pack soil are close	
Solution C	
Pros	Cons
Water tank, water pump and pack soil are close	LED illuminate the side of the Pack soil
Camera film the front of the pack soil	Small water tank

Figure 6.7 Solutions with objective aggregation: (a) good (b) fair (c) mediocre

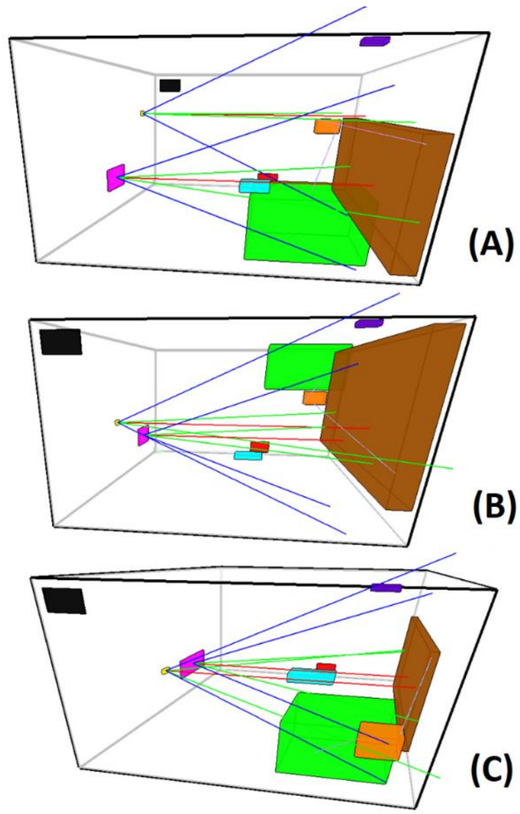


Figure 6.8 Different good solutions with objective aggregation

One can also wonder why there are still mediocre concepts outputted when the objective aggregation proposed in this work is used. The reason is that even though we reduce the number of objectives from 15 objectives to 5 objectives, the optimization problem remains complex and categorized as a many-objective optimization. Accordingly, this methodology cannot guarantee to find a set of objectives lower or equal to 3 objectives. However, the methodology, if well applied, can lower the number of objectives while conserving important objectives from a design point of view.

In future work, a study on the scalability of the aggregation method should be done. To achieve this, two steps are necessary. First, the aggregation method should be tested with a many-optimization algorithm since the proposed method does not guarantee a number of objectives lower or equal to 3. Second, more complex systems with more objectives functions need to be modeled and tested with the aggregation method and many-optimization algorithms. Furthermore, other aspects of the optimization process should be studied such as the speed of convergence in terms of the number of iterations required with and without the aggregation method. To do this, an estimate of the real Pareto front along with a more precise terminal condition is needed. Finally, the objectives' weight assignment should be changed and automated since the designers could unconsciously favor a design that might not be the near-optimal one.

6.6 CONCLUSIONS

In conclusion, we developed a methodology to reduce the number of objectives using design tools to aggregate objectives. This is meant to ease the task of optimizing the design with evolutionary algorithms and to have better control over the selection of objectives instead of automatically omit objectives as a designer would carefully reduce them. Even though the proposed methodology cannot guarantee a number of objectives lower or equal to three, it can be used to significantly reduce the number of objectives and allow for easier automated design synthesis. We applied the methodology to an autonomous greenhouse layout design task, and we were able to reduce the number of objectives from 15 to 5. In the meantime, the procedure resulted in producing additional 24 good concepts (total 33) instead of 9. Based on these good concepts, the AD could be used recursively to reduce the number of concepts during the detailed design phase by carefully adding information while respecting the information axiom. With the deepening of the design, more information about the system would be included (e.g. analysis on DPs-PVs of AD). Then, using our approach again, this iterative process could lead the designer to one or two concepts to be prototyped.

6.7 ACKNOWLEDGEMENTS

The authors would like to show their appreciation towards le Fonds de recherche du Québec – Nature et Technologies (FRQNT) and the Canadian Space Agency (CSA) for their financial support.

6.8 REFERENCES

[1] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-Objective Optimization: An Engineering Design Perspective," Berlin, Heidelberg, 2005, pp. 14-32: Springer Berlin Heidelberg.

[2] I. Hisao, T. Noritaka, and N. Yusuke, "Evolutionary many-objective optimization: A short review," in 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), 2008, pp. 2419-2426.

[3] S. Chand and M. Wagner, "Evolutionary many-objective optimization: A quick-start guide," Surveys in Operations Research and Management Science, vol. 20, no. 2, pp. 35-42, 2015/12/01/ 2015.

[4] W. Huang, Y. Zhang, and L. Li, "Survey on Multi-Objective Evolutionary Algorithms," Journal of Physics: Conference Series, vol. 1288, p. 012057, 2019/08 2019.

[5] J. M. Torry-Smith, A. Qamar, S. Achiche, J. Wikander, N. H. Mortensen, and C. During, "Challenges in designing mechatronic systems," Journal of Mechanical Design, vol. 135, no. 1, p. 011005, 2013.

[6] A. Mohebbi, L. Baron, S. Achiche, and L. Birglen, "Trends in concurrent, multi-criteria and optimal design of mechatronic systems: A review," in Innovative Design and Manufacturing (ICIDM), Proceedings of the 2014 International Conference on, 2014, pp. 88-93: IEEE.

[7] B. Li, J. Li, K. Tang, and X. Yao, "Many-Objective Evolutionary Algorithms: A Survey," ACM Comput. Surv., vol. 48, no. 1, p. Article 13, 2015.

[8] X. H. Nguyen, L. T. Bui, and C. T. Tran, "Improving many objective optimisation algorithms using objective dimensionality reduction," Evolutionary Intelligence, 2019/09/25 2019.

[9] K. Deb and D. Saxena, "Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," in Proceedings of the World Congress on Computational Intelligence (WCCI-2006), 2006, pp. 3352-3360.

[10] D. Brockhoff and E. Zitzler, "Objective Reduction in Evolutionary Multiobjective Optimization: Theory and Applications," Evolutionary Computation, vol. 17, no. 2, pp. 135-166, 2009.

[11] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective Reduction in Many-Objective Optimization: Linear and Nonlinear Algorithms," IEEE Transactions on Evolutionary Computation, vol. 17, no. 1, pp. 77-99, 2013.

[12] S. Bandyopadhyay and A. Mukherjee, "An Algorithm for Many-Objective Optimization With Reduced Objective Computations: A Study in Differential Evolution," IEEE Transactions on Evolutionary Computation, vol. 19, no. 3, pp. 400-413, 2015.

[13] A. R. R. Freitas, R. C. P. Silva, and F. G. Guimarães, "On the visualization of trade-offs and reducibility in many-objective optimization," presented at the Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation, Vancouver, BC, Canada, 2014. Available: https://doi.org/10.1145/2598394.2610550

[14] A. R. R. de Freitas, P. J. Fleming, and F. G. Guimarães, "Aggregation Trees for visualization and dimension reduction in many-objective optimization," Information Sciences, vol. 298, pp. 288-314, 2015/03/20/ 2015.

[15] A. Mohebbi, S. Achiche, and L. Baron, "Multi-criteria fuzzy decision support for conceptual evaluation in design of mechatronic systems: a quadrotor design case study," Research in Engineering Design, vol. 29, no. 3, pp. 329-349, 2018/07/01 2018.

[16] U. Chouinard, S. Achiche, and L. Baron, "Integrating negative dependencies assessment during mechatronics conceptual design using fuzzy logic and quantitative graph theory," Mechatronics, vol. 59, pp. 140-153, 2019/05/01/ 2019.

[17] G. G. Wang and S. Shan, "Design Space Reduction for Multi-Objective Optimization and Robust Design Optimization Problems," SAE Transactions, vol. 113, pp. 101-110, 2004.

[18] G. G. Wang and T. Simpson, "Fuzzy clustering based hierarchical metamodeling for design space reduction and optimization," Engineering Optimization, vol. 36, no. 3, pp. 313-335, 2004/06/01 2004.

[19] A. Albers, S. Klingler, and B. Ebel, "Modeling systems of objectives in engineering design practice," in DS 75-1: Proceedings of the 19th International Conference on Engineering Design (ICED13), Design for Harmonies, Vol. 1: Design Processes, Seoul, Korea, 19-22.08. 2013, 2013.

[20] R. Chen, Y. Liu, H. Fan, J. Zhao, and X. Ye, "An integrated approach for automated physical architecture generation and multi-criteria evaluation for complex product design," Journal of Engineering Design, vol. 30, no. 2-3, pp. 63-101, 2019/03/04 2019.

[21] H. Hirani and N. P. Suh, "Journal bearing design using multiobjective genetic algorithm and axiomatic design approaches," Tribology International, vol. 38, no. 5, pp. 481-491, 2005/05/01/ 2005.

[22] Z. Wu, X. Cheng, and J. Yuan, "Applying Axiomatic Design Theory to the Multi-objective Optimization of Disk Brake," Berlin, Heidelberg, 2012, pp. 62-73: Springer Berlin Heidelberg.

[23] J. M. Torry-Smith, N. H. Mortensen, and S. Achiche, "A proposal for a classification of product-related dependencies in development of mechatronic products," Research in Engineering Design, vol. 25, no. 1, pp. 53-74, 2014.

[24] D. Kalyanmoy, Multi objective optimization using evolutionary algorithms. John Wiley and Sons, 2001.

[25] Y. Ma, " Design conceptuel d’un incubateur pour un nanosatellite en combinant la conception axiomatique et un index unifié de performance mécatronique (MMP)," Masters thesis, École Polytechnique de Montréal, 2018.

[26] Y.-S. Law, Y. Ma, A. Vadean, G. Beltrame, and S. Achiche, "Evolutionary layout design synthesis of an autonomous greenhouse using product-related dependencies," Artificial Intelligence for Engineering Design, Analysis and Manufacturing (Submitted), 2019.

CHAPTER 7 ARTICLE 3: CONCURRENT PRODUCT MODULE IDENTIFICATION AND EVOLUTIONARY PRODUCT DESIGN OPTIMIZATION USING COMPLEX NUMBERS DSM AND PRODUCT-RELATED DEPENDENCY MANAGEMENT

The article presented by Yann-Seing Law, Giovanni Beltrame, Aurelian Vadean, and Sofiane Achiche in this chapter is submitted in the journal *Design Science*, Design Society, 2021.

Law-Kam Cio, Y., Beltrame, G, Vadean, A., & Achiche, S. (2021). Concurrent product module identification and evolutionary product design optimization using complex numbers DSM and product-related dependency management. *Design Science* (Submitted on the 13 April 2021).

7.1 Abstract

As technology progresses, more products increase in complexity such as mechatronic systems which in turn makes them more challenging to design. This design challenge is usually explained by the need for experts from different design domains as well as by the increased number and complexity of components integrated into the product. To alleviate the burden of designing such products, many industries and researchers are attracted to the concept of product modularization. Indeed, from an economic perspective, modularity can reduce the cost of product development by using the same modules throughout multiple product platforms. And, from a design perspective, the formation of modules shortens the product development by allowing concurrent development of product modules. The first step in achieving a modular product is to identify the possible parts of the product that can form a module. This identification can be difficult to achieve considering the trade-offs between modular design and design performance. Indeed, the design performance can be deteriorated by the formation of a module if negative product dependencies between components or modules are not accounted for. Hence, the product module identification must be carried out carefully by concurrently considering the positive and negative dependencies between components and the optimization of the design performance. To achieve this, the proposed methodology uses the product-related dependency management and the complex design structure matrix to model positive and negative dependencies and to compute the combination potency between components to create modules. This methodology is then integrated into a modified non-

dominated sorting genetic algorithm III to concurrently optimize the design and identify the product modules. The methodology is exemplified through the case study of a layout design of an autonomous greenhouse. By applying the proposed methodology to the case study, it was possible to generate concepts that decreased the number of modules from 9 down to 4 while still ensuring the optimization of the design performance.

7.2 Introduction

Mechatronic product design is a complex task to achieve for many reasons such as the need for multidisciplinary knowledge [1, 2]. To obtain a final product design, one must go through the whole design process composed of conceptual design, layout design, detailed design, testing, and much more. In each of these phases, one of the main challenges is to find near-optimal designs considering the information available. The problem statement of a mechatronic product design presents multiple conflicting objectives which means multiple concepts with different strengths and weaknesses will result from product optimization. From these outputted concepts, the designers must choose which concepts are favored. Many researchers in the field of engineering design, evolutionary computation, and generative design are still working on solving the optimization of complex real-life product design problems such as the optimization of mechatronic products. Therefore, from a design perspective, there still is a crucial need for a better methodology to model [3] and evaluate a design [4]. On the other hand, evolutionary computation researchers develop new algorithms to solve an optimization problem.

One of the trending engineering research axes is product modularity. Based on Gershenson et al. [5] work, the precise definition of product modularity has not been unanimously defined among the research community. However, most definitions share the same idea that combining or regrouping components into modules increases the modularity of a product. Many benefits come with the modularization of a product as it becomes easier to replace a part when it is broken or when the system needs to be upgraded. Moreover, it is also easier to redesign the layout of the product by changing the position of the modules or testing different modules to compare their performance. Additionally, modularity allows for the development of product families which reduces the number of parts to be dealt with by both the designer and the company. The impact of modularity on product and product process performances has also been evaluated up to a certain degree with more empirical studies [6, 7]. Generally, product modularity improves product design and development.

In this work, concepts from engineering design and evolutionary computation are combined to overcome issues related to product module identification during the layout design phase. Product module identification can be seen as combining two or more components, subsystems, or systems to form a module. To achieve product module identification, one must face three main issues.

The first issue is to correctly model the product-related dependencies [8, 9]. A component's dependency can be defined as how a component affects another one. The dependency between two components can be detrimental. For example, a heat source close to a heat sensor will be caused the heat sensor to capture erroneous values of the environment. The dependency can also be beneficial such as a heat source close to a fan that can help regulate the temperature of a room.

The second issue is to define the combination compatibility between components. The combination of two components implies that they are physically integrated to form a module. Considering that two components can have both beneficial and detrimental dependencies at the same time, it can be difficult to decide whether they should be combined or not within the same module.

The last issue is to identify product modules that do not deteriorate the design optimizer performances. Hence, the product module identification methods must be well integrated into the design optimization process. For example, evolutionary computation has operators such as crossover that will need to be adapted to maintain the design space exploration capabilities of the evolutionary algorithm.

The main objective of this paper is to integrate product module identification during the product design optimization process. The contribution lies in product module identification based on product-related dependency management and complex number design structure matrix representation which are used to evaluate the combination compatibility between components. The product module identification also utilizes the computation power using an evolutionary algorithm to generate and evaluate modules. The methodology will be tested on a mechatronic product which is the layout design of an autonomous greenhouse.

7.3 Literature review

Even though product modularity has beneficial effects on product design and development, there are still many research topics that need to be studied. One of the main challenges, mentioned by Hölttä et De Weck. [10], is evaluating when to favor a modular design or an integral design. Many researchers are working on overcoming this challenge. For example, AlGeddawy et al. [11] achieved modular product design based on the design for assembly. To do this, a combination of Cladistics which is a classification tool used in biology, design structure matrix (DSM), and the principle of product granularity level was used. This methodology allows the designer to have a compromise between product modularity and integration. Moreover, Höltta et al. [12], mentioned that there is a tradeoff between modularity and the performance of the product. If the constraints and performances of the product are more rigid, an integral design seems to be more adequate. However, if the performance requirements are flexible, a modular design can be favored for product variety, ease of re-design, maintenance, and repairs, etc. From this, it is possible to realize that the degree of modularity depends on the product specifications and purposes. To evaluate the degree of modularity needed for a product, many papers can be found in the literature. For an exhaustive literature review, it is recommended to look at Gershenson et al. [13] as well as a more recent review of Bonvoisin et al. [14]. This paper focuses on the product module identification based on product-related dependencies through product design optimization using evolutionary computation.

Yu et al. [15] automatically developed modular product architectures through DSM clustering. DSM clustering was achieved using a specialized genetic algorithm (GA) to manage the different possible clusters. The minimum description length was used as the objective function, which is a model that approximates the system. Xiaogang et al. [16] used a GA to reorganize the DSM to bring the values closer to the diagonal. A coordination cost function that was computed with the modified DSM was used to evaluate which cluster is better. Wrigley et al. [17] optimized the layout of a light water modular nuclear reactor power plant. Module identification was done by translating the process and instrumentation diagram into a DSM. Then, a clustering algorithm reorganized the DSM by maximizing the measure of effectiveness which is dependent on the connection penalty. Then, with these modules, the layout optimization was done using a GA by minimizing the piping distance between modules. Cheng et al. [18] presented a modularization method based on axiomatic design and design structure matrix. The axiomatic design was used to decompose the system in terms of functional requirements, physical solution, and process. Then, the DSMs presented the interaction between design parameters. Finally, a GA optimized the minimal description length for DSM clustering in order to achieve design parameters' module identification. In these works, the product module identification is achieved by reorganizing the DSM and clustering components based on component-component interactions. The issue with these methodologies is that the effect of modularisation on the design performance is not included in the design optimization or even design selection. Hence, the set of modules found with DSM clustering is not necessarily the optimal set of possible modules in terms of design performance.

Tseng et al. [19] designed a product for the green life cycle using a GA algorithm that optimizes the "liaison" intensity. If the liaison intensity between two components is high, then these two components are more likely to be in the same module. Consequently, the idea is to maximize the liaison intensity in a module and to minimize the liaison intensity between modules. Paras et al. [20] used a grouping GA to redesign used products in the garment industry. The product was divided into a manageable number of parts. Then, the grouping GA combined different parts to achieve the redesign requirements. The objective called design fitness is a weighted sum of four connection attributes. Connection attributes are related to the connection points of two parts. Similar connection attributes are better for the redesign. In the last two works, the modularization process is done mainly by considering the problem-specific variables. For mechatronic product design, a more generalized methodology is needed due to the multidisciplinary and complex nature of mechatronic products.

Xiao et al. [21] methodology first modeled the functional and structural aspects of the system using the relationship constraint network model. Then, a GA combined with fuzzy pattern recognition was used for module decomposition. The objective function is the proximity of the pattern of a solution with a user-defined ideal pattern. This methodology allows the designer to find a near user-defined ideal pattern forming the product; therefore, the goal being reproducing expert tacit knowledge. However, the near-optimal pattern is not guaranteed since it is user-dependent. Meng et al. [22] achieved module identification for product families using a single objective GA. The single objective is a weighted sum aggregation of four different objectives related to a modular design for product families: structural independency, functional independency, localization of change, and isolation of individualization. Shan et al. [23-25] used the weighted sum of four types of DSM for a model formulation which is functional, geometrical, physical, and auxiliary. Then, evolutionary computation algorithms were used to optimize an objective function based on the life cycle of the product and two principles of modular design. The first modular design principle is components within a module have high coupling. The second one is that the coupling between modules is low. The evolutionary computation algorithms used were particle swarm optimization, improved particle swarm optimization, and harmony search algorithms. Kreng and Lee [26] offered a four-phase methodology. The first phase is analyzing the functional and physical interactions between components. Phase two is finding the proper modularity metrics or modular driver which will guide the modularity process and defining their importance using a hierarchical analytic process. Also, the relationship between modular drivers and components was quantified in a correlation matrix. The third phase consists of modeling the modularization using a non-linear programming model. In this phase, the objective function was computed following the concept that a module will be composed of components with similar module drivers. The final phase is optimizing the objective function using a grouping GA that clusters components into modules. In these works, the optimization considers modular metrics as well as product-related metrics. However, they use a single objective optimization for a multi-objective optimization problem which means they will converge on one design based on the selection of promising designs. Furthermore, the design performance is not included in the optimization process. Xu et al. [27] used a two-step methodology for the modular design filament winding process equipment. The first step is to find the modules from components using a grouping GA and modular driving forces. The second step is to optimize the performance requirement using a non-dominated sorting genetic algorithm II and the modules found in the first step. Even though multi-objective optimization is used, the modularization process and product optimization are done separately. As mention above, this could lead to a set of suboptimal modules for the optimization design. Wei et al. [28] used three principles for modular design. The first one is internal interaction which represents the interaction between components within a module. The second is external interaction which is the interactions between modules. The last one is the overall reliability principle which dictates that components affecting the same functional requirements should be combined into a module. To identify modules, a multi-objective optimization is done where all three principles are concurrently optimized using the improved strength Pareto evolutionary algorithm (ISPEA2). Hence, a Pareto set of solutions is found. Finally, to choose the best set from the Pareto set, a fuzzy selection mechanism is used to eliminate the bias from the human selection. Modular metrics are used for the optimization of the modular process using multi-objective optimization. However, the design performance is not included in the optimization process.

As mentioned by Gershenson et al. [13], there is a need for flexibility in the modularization of a product to consider modules in the early design stages such as layout design. Moreover, Bonvoisin et al. [14] also express a lack of flexibility during the modularization process. Metrics and methods for modularization are often problem-specific which restrict their use. Hence, in this work, a flexible modularization method combining product-related dependency management, complex design structure matrices, and multi-objective evolutionary computation is proposed. The developed methodology allows the modularization process in the early design phase of a product design while considering its design performance. To validate it, the layout design of an autonomous greenhouse will be used as a case study.

The remaining part of the paper will be structured as follows: Section 7.4 describes the proposed methodology. Section 7.5 presents the layout design of an autonomous greenhouse as a case study. Section 7.6 reports the results and analysis of the layout design of an autonomous greenhouse. Finally, section 7.7 concludes this paper.

7.4 System design description

Figure 7.1 shows the overview of the methodology as well as the contribution of this paper. The problem statement and optimization problem formulation have been treated in our previous work [29]. Part of it will be explained in Section 7.5 for completeness. However, the main focus will be the component combination modeling as well as its integration in the optimization algorithm.

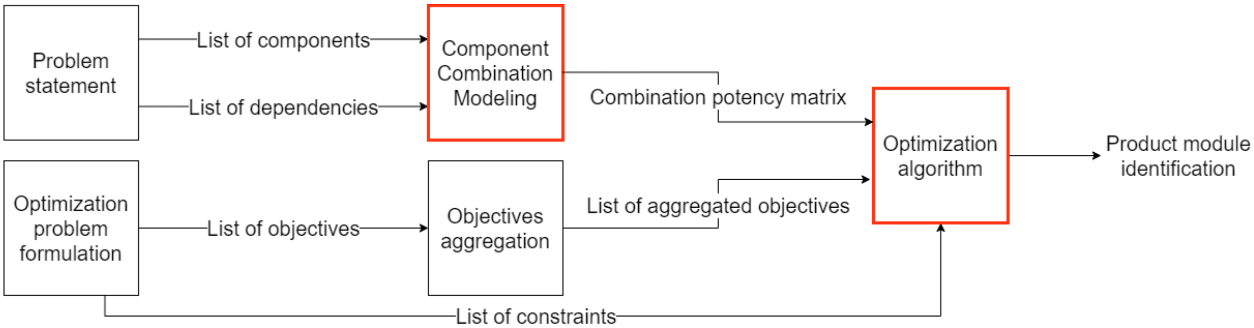


Figure 7.1 Overview of the methodology. The red squares represent the contribution of this paper.

7.4.1 Modeling components' dependencies

To model components' dependencies, complex DSMs developed in the authors' previous work. [3] are used. These DSMs use the complex number notation $a_{ij} + b_{ij}j$, as shown in the component-component matrix of Table 7.1, to differentiate negative and positive dependencies. The positive one is the real part of the complex value and the negative one is the imaginary part. This notation has the advantage of conserving and accumulating positive and negative dependencies separately during the aggregation of matrices. In this methodology, the chosen aggregation method is a weighted sum of all the DSMs.

Table 7.1 Concept of the complex DSM

Component	A	B	C	…
A	$a_{11} + b_{11}j$	$a_{12} + b_{12}j$	$a_{13} + b_{13}j$	$a_{1n} + b_{1n}j$
B	$a_{21} + b_{21}j$	…	…	…
C	$a_{31} + b_{31}j$	…	…	….
D	$a_{n1} + b_{n1}j$	…	…	$a_{nn} + b_{nn}j$

7.4.2 Modeling the component combination

For the DSM aggregation process, a matrix named expert combination matrix (ECM) is added to take into consideration the designer experience in combining components. ECM is introduced since the designers might know that two components cannot be combined into a module or that the work needed to combine two components consume too many resources in terms of time and cost even though there are no clear adverse effects. Once the aggregation matrix is obtained, a matrix called combination potency matrix (CPM) can be computed. This matrix also has the number of components as the number of rows and columns (i.e. squared matrix). Each element of the matrix is a value given by $f\left(a_{ij} + b_{ij}j\right)$ as shown in Table 7.2. The complex number $a_{ij} + b_{ij}j$ value is found in the aggregation matrix. The idea is to find a function $f\left(a_{ij} + b_{ij}j\right)$ that allow the quantification of the combination potency based on the complex number notation.

Table 7.2 Concept of computing the combination potency matrix (CPM)

Component	**A**	**B**	**C**	**…**
A	$f(a_{11} + b_{11}j)$	$f(a_{12} + b_{12}j)$	$f(a_{13} + b_{13}j)$	$f(a_{1n} + b_{1n}j)$
B	$f(a_{21} + b_{21}j)$	…	…	…
C	$f(a_{31} + b_{31}j)$	…	…	….
D	$f(a_{n1} + b_{n1}j)$	…	…	$f(a_{nn} + b_{nn}j)$

The concept of complex DSMs has not been widely used. Hence, there is no formulation of $f\left(a_{ij} + b_{ij}j\right)$ given in the literature, therefore in this work, a simple ratio $f\left(a_{ij} + b_{ij}j\right) = \frac{a_{ij}}{b_{ij}}$ will be used. The chosen function is based on a simple and intuitive view of the matter. Indeed, a ratio of positive dependencies and negative dependencies gives a range of values from 0 to ∞. Where 0 means that there is no combination potency between two components and ∞ means that there is a high combination potency. In Table 7.3, a scale of numerical and qualitative values is shown based on the different values of the real and imaginary parts.

Table 7.3 Qualitative and numerical values are associated with complex number notation.

Case number	**Different cases of a_{ij} and b_{ij}**	**Numerical value**	**Qualitative value**
1	a_{ij}>0 & b_{ij} = 0	∞	High combination potency
2	a_{ij}>b_{ij}	]1, ∞ [	Combination potency
3	b_{ij} = 0 & a_{ij} = 0	1^{+}	Independent
4	b_{ij} = a_{ij} (b_{ij}≠0 & a_{ij}≠0)	1	Neutral

5	a_{ij}<b_{ij}	]0,1[	Low combination potency
6	a_{ij} =0 & b_{ij}> 0	0	No combination potency

It is important to mention the difference between cases 3 and 4. The combination process of case 3 should be easier than the one for case 4, since in case 3, the two components are independent due to the absence of dependencies. Moreover, in case 4 the designers need to deal with adverse effects. For this reason, the numerical value for case 3 is slightly higher than for case 4 which is why the numerical value for case 3 is represented by the " 1^+ " symbol.

7.4.3 Integrating the component combination in evolutionary computation

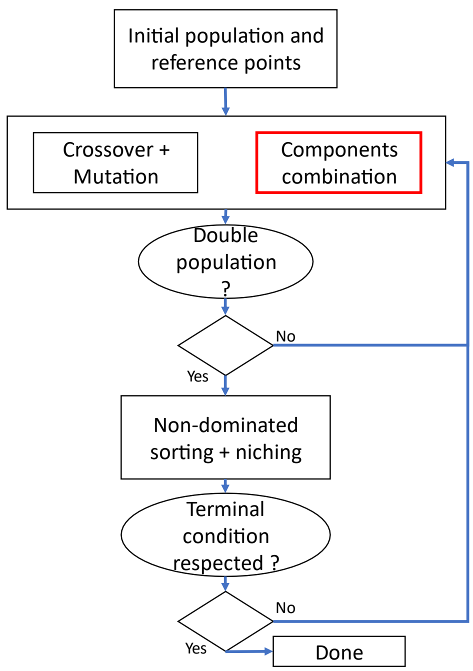


Figure 7.2 Overview of the modified NSGA-III algorithm. The red square represents the added operator.

From Figure 7.2, not only the methodology allows individuals with modules to compete with individuals without modules, but it allows competition between modules. Hence, it is possible to evaluate the effect of modularization on the design performance (i.e. objectives), thus, if a specific module is beneficial, it will improve one or more objectives. Using this strategy, the algorithm will favor optimal modules, if they exist, based on the design performance. However, the optimization algorithm must be adapted to achieve this.

The chosen evolutionary algorithm in this work is a modified version of the non-dominated sorting genetic algorithm -III (NSGA-III) [30]. Part of these modifications is presented in a previous work of the authors, Law et al. [31]. The components' combination has also a chance of being performed during the reproduction phase of the NSGA-III (See Figure 7.2). There is a probability associated with the occurrence of the combination operator just like the mutation operator. The components' combination is done on a clone of a random individual from the population.

A combination tracker vector (CTV) is also introduced to know which initial components are now combined. Indeed, as the optimization progress, combinations are done stochastically, hence, two individuals can have different modules. The CTV is used to know which components are combined into a module and to adapt operators and/or mechanisms of the optimization algorithm. To better understand how CTV is modified throughout the optimization process, Table 7.4 represents the formation of a module of one individual. The first column shows the vector's length with the initial number of components or the initial state of CTV. The second column shows the impact of forming one module on the CTV. The module is formed by integrating the 7th components in the 2nd one which is why the value of the 2nd column, 7th row is set to 2. The last column represents how the CTV will be changed if another component is added to the module. The 9th component is also integrated with the 2nd component which is why the value of the 3rd column, 9th row is 2. The module is now formed of the 2nd, 7th, and 9th components.

Table 7.4 Example of the use of CTV

Initial	**Comp2and7 combined**	**Comp9and7 combined**
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6

7	2	2
8	8	8
9	9	2

The CTV also offers another advantage. As mentioned earlier, some evolutionary operators will be affected by the combination of components. For this implementation, the crossover operator is the only one that has a conflict with the combination of components. Indeed, applying a crossover operator between two solutions with different combinations is most likely to output an implausible individual as shown in Figure 7.3. In this work, an implausible individual is defined as an individual with too many or too few components. This is where the CTV can be useful. To avoid implausible individuals, a crossover between two individuals can happen if their CTVs are identical. In other words, the crossover operation can happen when two individuals have identical modules. However, if their CTVs are not identical, but share an identical module at the crossover point, an exchange between these identical modules will be done. Otherwise, no crossover operator is done between the two individuals. This modification of the crossover operation is done to ensure the exploration of the design space.

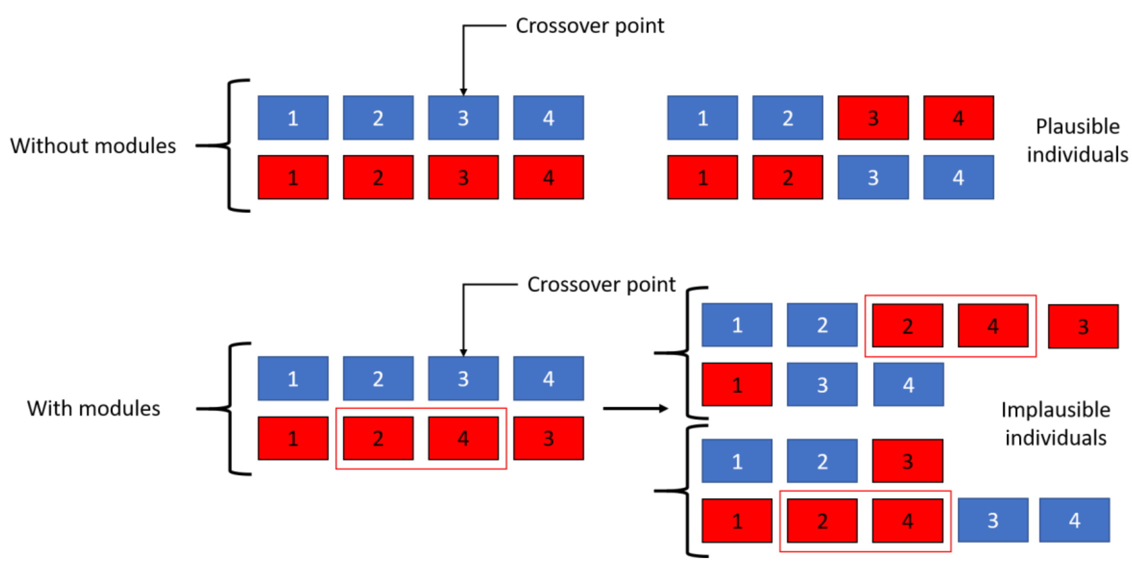


Figure 7.3 Comparison of crossover operations on individuals with and without modules

7.5 Case study: Autonomous greenhouse layout design

In [29, 31], the authors proposed a case study based on components' placement and dependencies for the layout design of an autonomous greenhouse. This case study is chosen because it can greatly benefit from modularity. Indeed, by integrating components into modules, it is possible to reduce the volume occupied by components which will increase the volume allocated for the growth of the plants. Hence, the autonomous greenhouse will better fulfill its main function which is to ensure the growth and survival of plants. The formulation will briefly be presented here.

The main components are presented in Table 7.5 and the new DSMs are presented in Figure 7.4. The DSMs have been modified using the scale for complex notation presented in Table 7.6.

Table 7.5 Main components of an autonomous greenhouse

Component's number	Component's name
1	Heater
2	Water tank
3	Pack-soil (contains seeds and soil)
4	Heat sensor
5	Camera
6	Fan
7	LED (lights)
8	Water pump
9	PCB (sensors such as O_2, CO_2, etc.)

Using the formulation, the layout design of most systems can be modeled by three matrices. The first one is the closeness matrix and represents how far or close a component should be from another one. The second matrix is the field of view (FOV) matrix which indicates that a component needs to be within or outside of the FOV of another one. The last one is the physical connection matrix and defines the number of connections between two components. The aggregation of these three matrices is computed using the weighted sum approach. Using the aggregated matrix, the CPM is calculated and shown in Figure 7.5. These matrices were adapted to the complex scale presented in Ugo et al. [3]:

Table 7.6 Scale for complex number notation from Ugo et al. [3]

Qualitative value	Detrimental	Undesirable	Neutral	Desirable	Necessary
Complex value	0 + 2j	0 + 1j	0 + 0	1+0j	2+0j

	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	0	0	0	2j	0	2	0	0	0
C2	0	0	1	0	0	0	0	1	0
C3	0	1	0	0	0	0	0	1	0
C4	2j	0	0	0	0	0	2j	0	0
C5	0	0	0	0	0	0	0	0	0
C6	2	0	0	0	0	0	0	0	0
C7	0	0	0	2j	0	0	0	0	0
C8	0	1	1	0	0	0	0	0	0
C9	0	0	0	0	0	0	0	0	0

(a)

	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	0	0	0	0	0	0	0	0	0
C2	0	0	0	0	0	0	0	0	0
C3	0	0	0	0	2	0	2	0	0
C4	0	0	0	0	0	0	0	0	0
C5	0	0	2	0	0	0	0	0	0
C6	0	0	0	0	0	0	0	0	0
C7	0	0	2	0	0	0	0	0	0
C8	0	0	0	0	0	0	0	0	0
C9	0	0	0	0	0	0	0	0	0

(b)

	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	0	0	0	0	0	0	0	0	0
C2	0	0	0	0	0	0	0	1	0
C3	0	0	0	0	0	0	0	1	0
C4	0	0	0	0	0	0	0	0	0
C5	0	0	0	0	0	0	0	0	0
C6	0	0	0	0	0	0	0	0	0
C7	0	0	0	0	0	0	0	0	0
C8	0	1	1	0	0	0	0	0	0
C9	0	0	0	0	0	0	0	0	0

(c)

	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	0	0	0	2j	0	2	0	0	0
C2	0	0	1	0	0	0	0	2	0
C3	0	1	0	0	2	0	2	2	0
C4	2j	0	0	0	0	0	2j	0	0
C5	0	0	2	0	0	0	0	0	0
C6	2	0	0	0	0	0	0	0	0
C7	0	0	2	2j	0	0	0	0	0
C8	0	2	2	0	0	0	0	0	0
C9	0	0	0	0	0	0	0	0	0

(d)

C1	Heater
C2	Water tank
C3	Pack soil
C4	Heat sensor
C5	Camera
C6	Fan
C7	LED
C8	Water pump
C9	PCB

Figure 7.4 DSMs of the layout design of an autonomous greenhouse. DSM of closeness (a) DSM of the field of view (b) DSM of physical connections (c) Aggregated DSM (d)

	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	0	1^+	1^+	0	1^+	∞	1^+	1^+	1^+
C2	1^+	0	∞	1^+	1^+	1^+	1^+	∞	1^+
C3	1^+	∞	0	1^+	∞	1^+	∞	∞	1^+
C4	0	1^+	1^+	0	1^+	1^+	0	1^+	1^+
C5	1^+	1^+	∞	1^+	0	1^+	1^+	1^+	1^+
C6	∞	1^+	1^+	1^+	1^+	0	1^+	1^+	1^+
C7	1^+	1^+	∞	0	1^+	1^+	0	1^+	1^+
C8	1^+	∞	∞	1^+	1^+	1^+	1^+	0	1^+
C9	1^+	1^+	1^+	1^+	1^+	1^+	1^+	1^+	0

C1	Heater
C2	Water tank
C3	Pack soil
C4	Heat sensor
C5	Camera
C6	Fan
C7	LED
C8	Water pump
C9	PCB

Figure 7.5 CPM of the layout design of an autonomous greenhouse

In this case study, it is important to note that there is a difference between the closeness and the combination of two components. To illustrate this, the water tank will be taken as an example. The water tank could be close to the LEDs without any consequences. However, combining the water tank and the LEDs is not favored for reasons such as water and electronics are not compatible, and their combination can be troublesome, but possible.

To input this sort of information, the ECM is used since its purpose is to include expert knowledge about the physical integration between components as mentioned above. Hence, the water tank and electric components have a value of 1j since such a combination is not favored since it would make the product design and development more complex. Furthermore, a value of 1 is given to the ECM element of the LED and the camera to avoid lens flare since the light could prevent the camera from achieving its functional requirement. Including the ECM in the aggregation process, the new aggregated matrix and CPM are given in Figure 7.6. It is possible to see that the input of the designer on the combination of components has an important impact on the CPM. For example, the LED and camera have high combination potency as opposed to their potency being neutral before.

	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	0	1j	0	0	0	0	0	0	0
C2	1j	0	0	1j	1j	1j	1j	1j	1j
C3	0	0	0	0	0	0	0	0	0
C4	0	1j	0	0	0	0	0	0	0
C5	0	1j	0	0	0	0	1	0	0
C6	0	1j	0	0	0	0	0	0	0
C7	0	1j	0	0	1	0	0	0	0
C8	0	1j	0	0	0	0	0	0	0
C9	0	1j	0	0	0	0	0	0	0

(a)

	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	0	1j	0	2j	0	2	0	0	0
C2	1j	0	1	1j	1j	1j	1j	2+j	1j
C3	0	1	0	0	2	0	2	2	1j
C4	2j	1j	0	0	0	0	2j	0	0
C5	0	1j	2	0	0	0	1	0	0
C6	2	1j	0	0	0	0	0	0	0
C7	0	1j	2	2j	1	0	0	0	0
C8	0	2+j	2	0	0	0	0	0	0
C9	0	1j	1j	0	0	0	0	0	0

(b)

	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	0	0	1^+	0	1^+	∞	1^+	1^+	1^+
C2	0	0	∞	0	0	0	0	2	0
C3	1^+	∞	0	1^+	∞	1^+	∞	∞	1^+
C4	0	0	1^+	0	1^+	1^+	0	1^+	1^+
C5	1^+	0	∞	1^+	0	1^+	∞	1^+	1^+
C6	∞	0	1^+	1^+	1^+	0	1^+	1^+	1^+
C7	1^+	0	∞	0	∞	1^+	0	1^+	1^+
C8	1^+	2	∞	1^+	1^+	1^+	1^+	0	1^+
C9	1^+	0	1^+	1^+	1^+	1^+	1^+	1^+	0

(c)

C1	Heater
C2	Water tank
C3	Pack soil
C4	Heat sensor
C5	Camera
C6	Fan
C7	LED
C8	Water pump
C9	PCB

Figure 7.6 Integrating the ECM. ECM (a) New aggregated matrix (b) New CPM (c)

In this paper, the objective function from the authors' previous work has been used and reported in Table 7.7 to obtain layout designs of an autonomous greenhouse. This table shows the relationship between the dependencies found using product-related dependency management and the objective functions. These objectives treat information about the placement and size of components to ensure the survival and growth of the plants. DX, DY, and DZ represent the dimensions of a component, and PX, PY, and PZ its position within the greenhouse. Finally, M, V, and A are the mass, voltage, and current of a component respectively.

Furthermore, the strategy for the aggregation of objectives is illustrated in Figure 7.7. The methodology used to aggregate these objectives uses axiomatic design to identify the sub-systems and components of the greenhouse. Then based on the dependencies and objective functions of the layout design (Table 7.7) and the axiomatic design, the objective functions are aggregated by associating them to subsystems. More details of this methodology can be found in the author's previous work [31].

Table 7.7 Dependencies and objective functions of the layout design of an autonomous greenhouse

Number	Name of dependency	Max or min	Objective function	Decision variables
1	Total volume	Min	Σ(DX*DY*DZ)	DX,DY,DZ
2	Volume of pack soil and water tank	Max	Σ(DX*DY*DZ)	DX,DY,DZ
3	Mass	Min	Σ M	M
4	Energy consumption	Min	Σ(V*A)	V,A
5	Distance between pack soil and water tank	Min	Eucledian Dist	PX,PY,PZ
6	Distance between pack soil and water pump	Min	Eucledian Dist	PX,PY,PZ
7	Distance between water tank and water pump	Min	Eucledian Dist	PX,PY,PZ
8	Distance between heater and fan	Min	Eucledian Dist	PX,PY,PZ
9	Distance between connection points of water tank and water pump	Min	Eucledian Dist	PX,PY,PZ
10	Distance between connection points of pack soil and water pump	Min	Eucledian Dist	PX,PY,PZ
11	Distance between LED and heat sensor	Max	Eucledian Dist	PX,PY,PZ
12	Distance between heater and heat sensor	Max	Eucledian Dist	PX,PY,PZ
13*	Lighting of the pack soil with the LEDs	Max	Eucledian Dist	PX,PY,PZ
14*	Capturing of the pack soil with the camera	Max	Eucledian Dist	PX,PY,PZ
15	Distance between PCB and pack soil	Max	Eucledian Dist	PX,PY,PZ

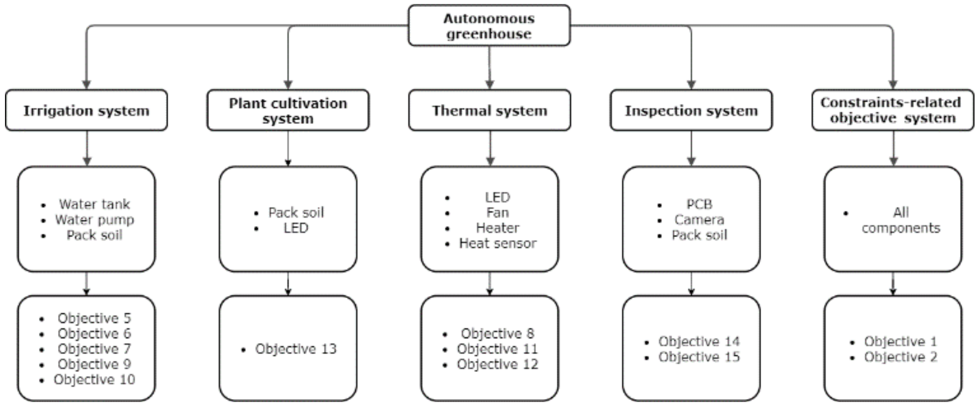


Figure 7.7 Objectives aggregation of the layout design of an autonomous greenhouse.

7.6 Results & Discussion

The simulation parameters are presented in Table 7.8. The parameters in dark gray are the algorithm parameters of the modified NSGA-III. The reference vector size and population size have been chosen based on the recommendation of Deb and Jain [30]. The rest of the parameters were fine-tuned by trials and errors. The probabilities of combination and mutation were defined so that the modularity process would not be done too quickly to allow the comparison of different levels of modularity within a simulation run. The design parameters in light gray are from the authors' previous work [29]. For completeness, the components' parameters from [29] are also reported in Table 7.9. It is important to mention that the modularization process starts only when the entire population is constraints-free. By doing so, it is possible to compare the feasible non-modular designs to feasible modular designs. If the modularization process is better, the non-modular design should completely disappear after a while.

Table 7.8 Simulation parameters. Parameters in dark gray are algorithm parameters and those in light gray are design parameters

Parameters	Values
Population size	212
Reference vector size	210
Number of unimproved generations to terminate the algorithm	1000

Probability of combination	80%
Initial probability of mutation	40%
Final probability of mutation	10%
Maximum voltage for one component in a solution	9 V
Maximum current for one component in a solution	1000 mA
Max mass of a solution	1500 g
Maximum energy consumption of an individual	15 W
Greenhouse dimension	450 x 300 x 300 mm^3

Table 7.9 Components' parameters

<u>Pack-soil</u> Dimensions range: 250 x 175 x 8 to 450 x 300 x 20 mm^3 Mass range: 300 to 425 g	<u>Water tank</u> Dimensions range: 50 x 50 x 50 to 100 x 100 x 100 mm^3 Mass range:150 to 1200 g
<u>Heater</u> Dimensions range: 30 x 30 x 5 to 80 x 80 x 10 mm^3 Mass range: 20 to 50 g Voltage range: 3.3 to 12 V Current range: 400 to 7000 mA	<u>Heat sensor</u> Dimensions range: 12 x 12 x 5 to 25 x 25 x 10 mm^3 Mass range: 0.1 to 1 g Voltage range: 1.7 to 3.6 V Current range: 0.01 to 0.02 mA
<u>Camera</u> Dimensions range: 10 x10 x 2.5 to 22 x 26 x 11 mm^3 Mass range: 0.1 to 6.4 g Voltage range: 1.7 to 5 V Current range: 50 to 160 mA Field of view: 60 to 90 °	<u>LED</u> Dimensions range: 40 x 40 x 1.84 to 100 x 100 x 2 mm^3 Mass range: 10 to 35 g Voltage range: 2.9 to 3.7 V Current range: 700 to 1400 mA Field of view: 60 to 90 °

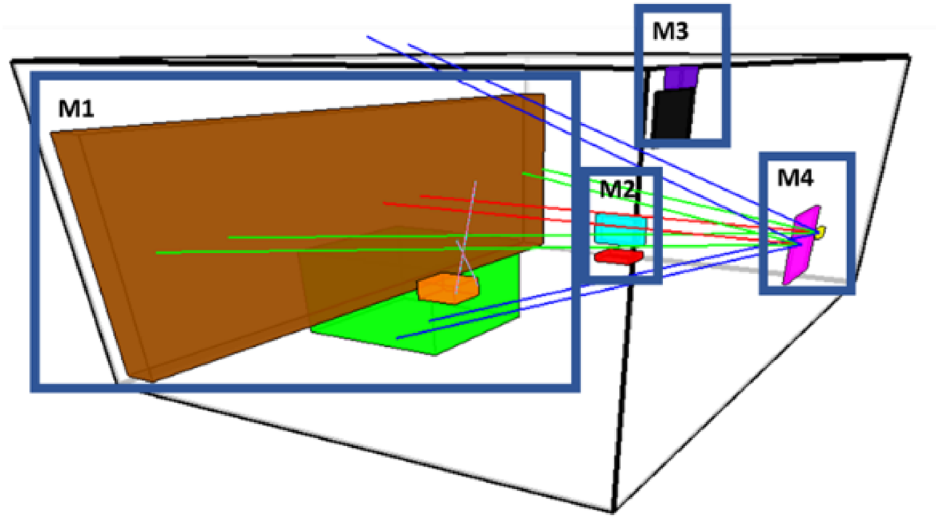
<u>Fan</u> Dimensions range: 40 x40 x 10 to 80 x 80 x 25 mm^3 Mass range: 18.6 to 62.6 g Voltage range: 2 to 5.5 V Current range: 66 to 170 mA	<u>Water Pump</u> Dimensions range: 32 x 32 x 23 to 54 x54 x 46 mm^3 Mass range: 80 to 150 g Voltage range: 3 to 12 V Current range: 200 to 500 mA
<u>PCB</u> Dimensions range: 30 x 30 x 1 to 50 x 50 x 4 mm^3 Mass range: 5 to 10 g Voltage range: 3.3 to 6 V Current range: 5 to 50 mA	

To show the applicability of the methodology presented above, a comparison of simulation runs done with and without the modularization process is carried out. Figure 7.8 shows different examples of layouts of 10 simulation runs without modularization. 10 simulation runs were done for reproducibility purposes. Hence, there were many duplicated layouts and similar layouts that were outputted from these simulation runs. Therefore, the presented layouts in Figure 7.8 are the most common and near-optimal ones. By analyzing the layouts of this study case, it is possible to notice that the optimization of the layout unveils clusters of components. Indeed, all the examples show physical proximity between the pack soil, water tank, and water pump. The same conclusion can be made for the heater and the fan as well as the camera and the LED. As for the heat sensor and the PCB, only layout A of Figure 7.8 shows a possible combination. The red, blue, and green lines in each figure are for the line of sight and field of view of the components. The red lines are the line of sight of the component starting from the center of the component. The blue and green lines are the field of view of the components in the form of a cone.

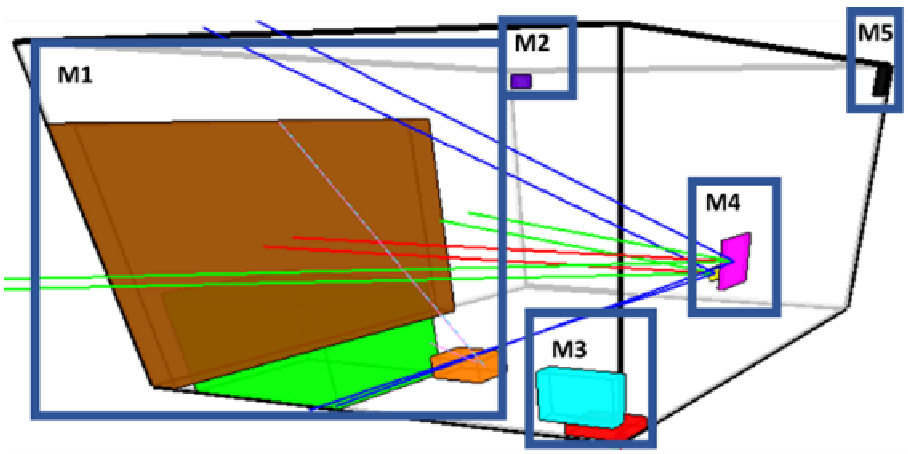
Figure 7.9, Figure 7.10, and Figure 7.11 show the results of 10 simulation runs. The layouts were chosen to show the different levels of modularization while achieving the functional requirements.

Figure 7.9 shows four examples of highly modular layouts of an autonomous greenhouse. These layouts have 4 modules. As expected from the CPM of as well as the modules identified in Figure 7.8, the water tank and pack soil are combined due to their high combination potency for each layout. For the same reason, it is possible to see that the camera and LED are also combined. For layout D, the heat sensor and the PCB are combined. The CPM shows that this combination can happen without impacting the system performance and layout A of Figure 7.8 shows that their combination can be expected. Looking at the CPM, the combination of the heater and fan is most likely to happen. This is confirmed as it is possible to see all the layouts of Figure 7.9 except C has the heater and fan combined. For layout C, the heater is the only component of its module, and the fan is combined with the camera, LED, and PCB. Furthermore, layout A and B include the PCB with the module composed of the camera and LED. These combinations were expected since the PCB has an independent dependency with all the components except the water tank. Finally, the water pump is combined with the water tank and pack soil even though the designer's suggestion was to avoid combining these two components in the ECM. This goes to show that their combination would greatly benefit the design performances even if it can also be difficult to achieve by the designer.

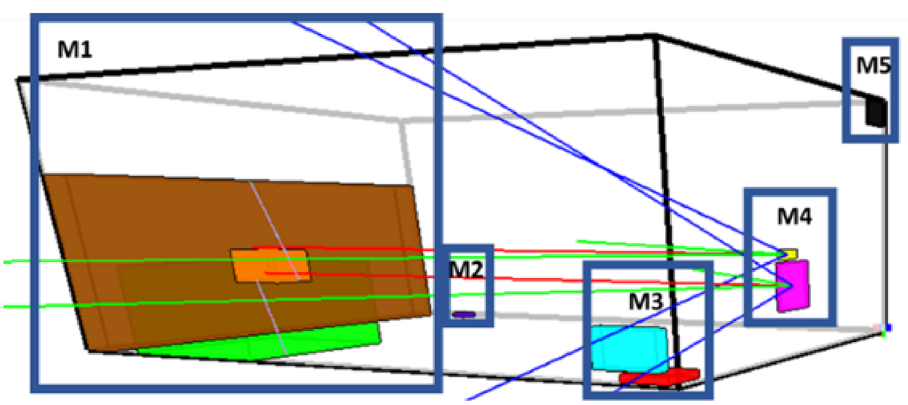
Figure 7.9 shows four examples of highly modular layouts of an autonomous greenhouse. These layouts have 4 modules. As expected from the CPM of as well as the modules identified in Figure 7.8, the water tank and pack soil are combined due to their high combination potency for each layout. For the same reason, it is possible to see that the camera and LED are also combined. For layout D, the heat sensor and the PCB are combined. The CPM shows that this combination can happen without impacting the system performance and layout A of Figure 7.8 shows that their combination can be expected. Looking at the CPM, the combination of the heater and fan is most
Figure 7.10 shows four examples of moderate modular layouts. The number of modules for these layouts is 5. It is possible to see that the water tank, water pump, and pack soil still form a module for every layout. The heater and fan module have only been present in layout A. For the other layouts, the fan is either combined with the camera and LED module (layout C) or is a module of its own (layout B and D). The separation of the heater and fan was not expected, considering that the CPM shows an infinite combination potency between these components.



A	
M1	PS/WT/WP
M2	Heater/Fan
M3	Heat sensor/PCB
M4	Camera/LED

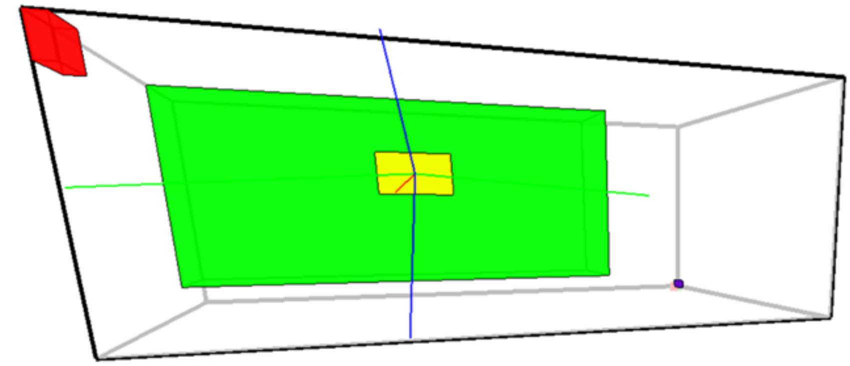


B	
M1	PS/WT/WP
M2	Heat sensor
M3	Heater/Fan
M4	Camera/LED
M5	PCB

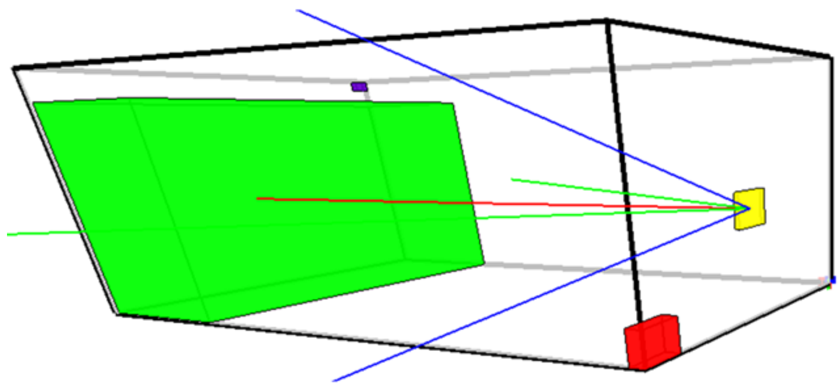


C	
M1	PS/WT/WP
M2	Heat sensor
M3	Heater/Fan
M4	Camera/LED
M5	PCB

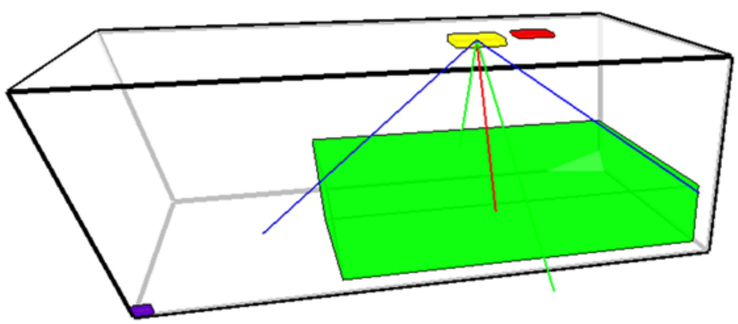
Figure 7.8 Examples of layouts are generated without modularization. Highlighting the possible modules.



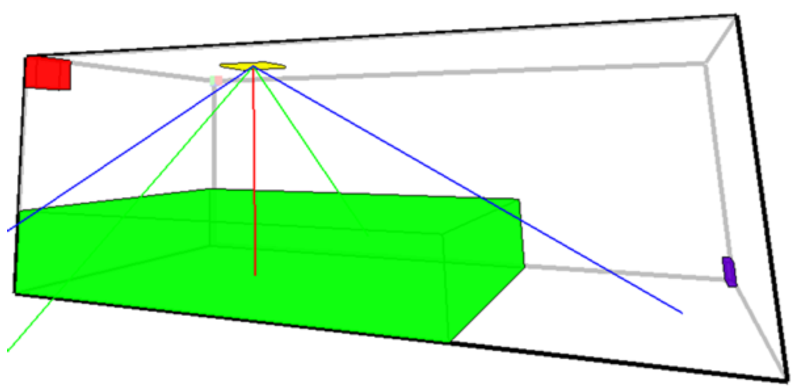
A	
Red	Heater/Fan
Green	WT/WP/PS
Purple	Heat sensor
Yellow	Camera/LED/PCB



B	
Red	Heater/Fan
Green	WT/WP/PS
Purple	Heat sensor
Yellow	Camera/LED/PCB



C	
Red	Heater
Green	WT/WP/PS
Purple	Heat sensor
Yellow	Camera/LED/Fan/PCB



D	
Red	Heater/Fan
Green	WT/WP/PS
Purple	Heat sensor/PCB
Yellow	Camera/LED

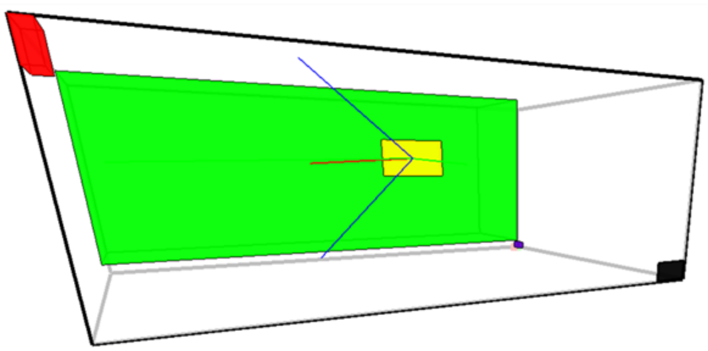
Figure 7.9 Examples of highly modular layouts

However, it seems that the combination of these two simply did not happen due to the stochastic nature of the optimization because layouts B and D show that the heater and fan are really close to one another similar to the layouts of Figure 7.8.As for layout C, the fan simply got combine with the camera and LED first. Then, the combination of the heater with this module was not optimal since the heater would be closer to the heat sensor.

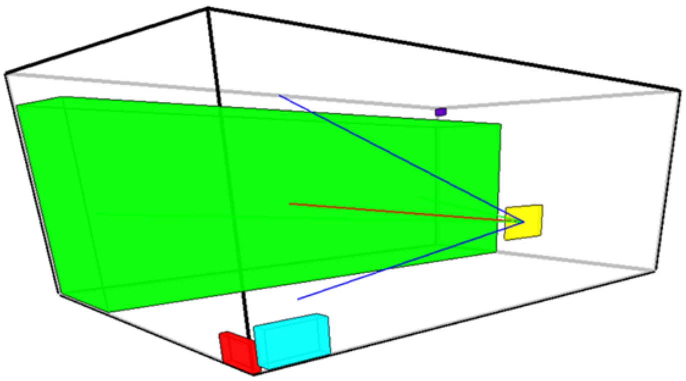
Figure 7.11 shows four examples of low modular layouts. This shows more than 5 modules. These layouts are similar to the layout of Figure 7.8. Indeed, it is still possible to see a tendency of combining the water tank, water pump, and pack soil into a module. The same goes for the heater and fan as well as the camera and LED. Hence, in the layouts of Figure 7.11, some of these modules have been made or partially made.

In all the examples of Figure 7.9, Figure 7.10, and Figure 7.11, the water tank and water pump have been combined even though the designer indicated that this combination is not suggested in the ECM. The reason why this combination is preferred can be explained by the fact that objectives 1, 7, and 9 of Table 7.7 greatly benefit from this combination. Indeed, combining the water tank and water pump uses less volume and tubes. Hence, the optimization shows the designer that this combination is more advantageous for the design of a greenhouse even if it has to overcome negative dependencies.

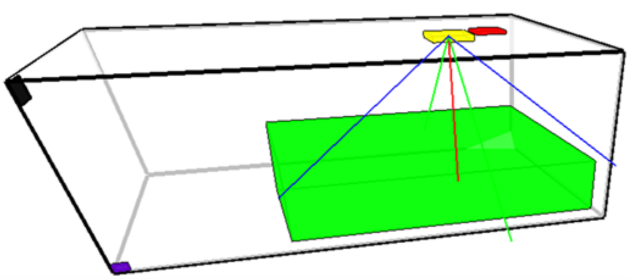
It is possible to observe the benefits of modularity on the layout design of an autonomous greenhouse by analyzing the results of Figure 7.9, Figure 7.10, and Figure 7.11. The more modular is the layout, the more available space there is for the growth of the plants. Furthermore, the module containing the pack soil and the water tank usually has a bigger volume that implies more seeds and/or more water can be used to grow plants. Finally, the field of view of the camera and LED projecting on the pack soil are less obstructed by other components. This is mainly due to the combination of the pack soil, water tank, and water pump since the water tank and water pump contributed the most to the obstruction of fields of view as shown in Figure 7.8.



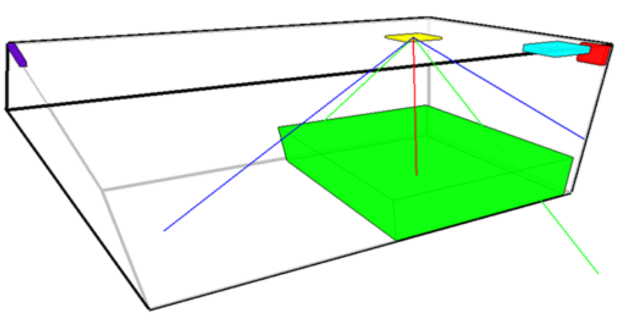
A	
Red	Heater/Fan
Green	WT/WP/PS
Purple	Heat sensor
Yellow	Camera/LED
Black	PCB



B	
Red	Heater
Green	WT/WP/PS
Purple	Heat sensor
Yellow	Camera/LED/PCB
Turquoise	Fan

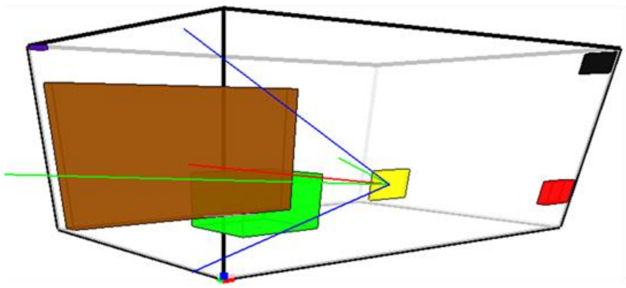


C	
Red	Heater
Green	WT/WP/PS
Purple	Heat sensor
Yellow	Camera/LED/Fan
Black	PCB

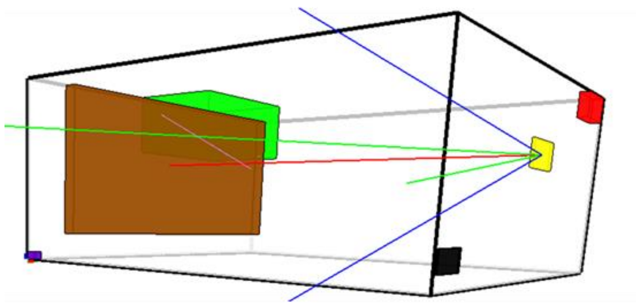


D	
Red	Heater
Green	WT/WP/PS
Purple	Heat sensor/PCB
Yellow	Camera/LED
Turquoise	Fan

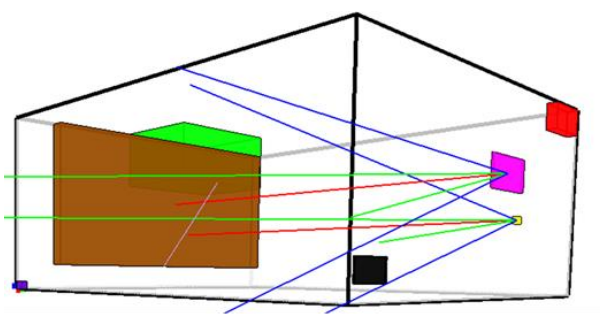
Figure 7.10 Examples of moderate modular layouts



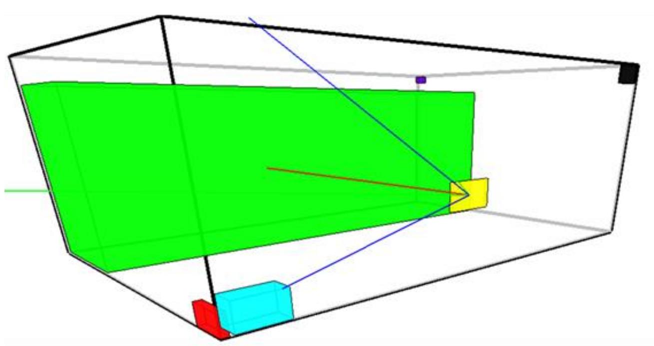
A	
Red	Heater/Fan
Green	WT/WP
Brown	PS
Purple	Heat sensor
Yellow	Camera/LED
Black	PCB



B	
Red	Heater/Fan
Green	WT/WP
Brown	PS
Purple	Heat sensor
Yellow	Camera/LED
Black	PCB



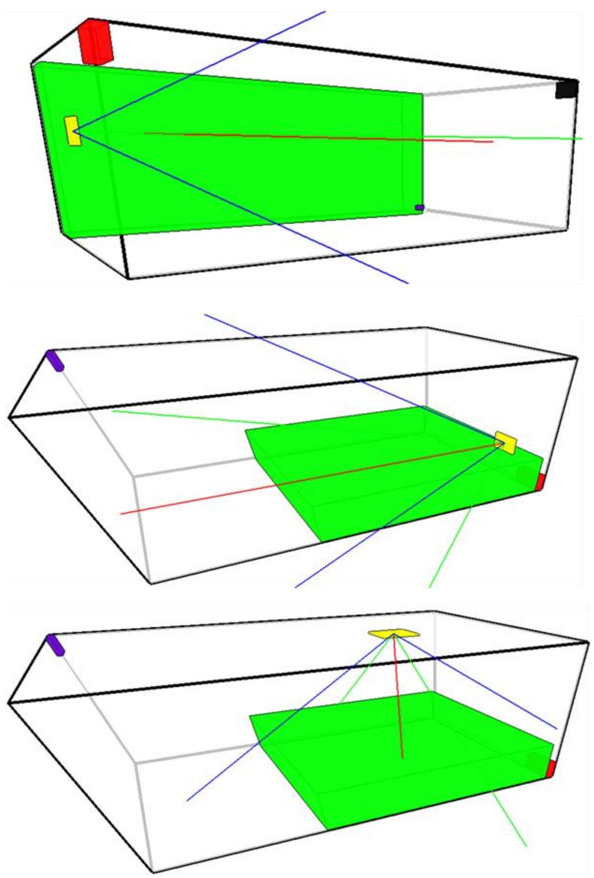
C	
Red	Heater/Fan
Green	WT/WP
Brown	PS
Purple	Heat sensor
Yellow	Camera
Magenta	LED
Black	PCB



D	
Red	Heater
Green	WT/WP/PS
Purple	Heat sensor
Yellow	Camera/LED
Turquoise	Fan
Black	PCB

Figure 7.11 Examples of low modular layouts

It is important to mention that some of the outputted layouts were poor. Layout A from Figure 7.12 cannot be used to ensure the growth and survival of the plants due to the undesired direction of the camera and LED module. The filming and illumination of the plant cannot be fulfilled adequately with this configuration. Layout B also shows this problem as well as the heater and fan module being blocked by the pack soil, water tank, and water pump module. This can cause a suboptimal performance of the fan which can lead to a local heat point or cold point. Layout C also shows this problem.



A	
Red	Heater/Fan
Green	WT/WP/PS
Purple	Heat sensor
Yellow	Camera/LED
Black	PCB

B	
Red	Heater/Fan
Green	WT/WP/PS
Purple	Heat sensor/PCB
Yellow	Camera/LED

C	
Red	Heater/Fan
Green	WT/WP/PS
Purple	Heat sensor/PCB
Yellow	Camera/LED

Figure 7.12 Examples of inadequate layouts

The presented results showed that the proposed methodology achieved product module identification while ensuring the functional requirements of the product. However, the methodology could be improved by overcoming one of the issues which is the absence of an exploratory mechanism (e.g. niching techniques) among the set of modules integrated into the optimization algorithm. Indeed, the formation of one module can prevent the formation of another one. Hence, it is possible to obtain a local optimal set of modules if it takes over the population. A local optimal can also mislead the designer and cause the final product to be less optimal or to not fulfill the customers' needs. To overcome this issue, a niching mechanism such as an adapted version of the crowding distance based on the CTV might be a solution in future works.

Another issue is that the CPM has static values that depend on the designers' weight assignments and the chosen computation method. Hence, the probability of combining two components is greatly impacted by the designers. In the case where the designer input is inaccurate, two mechanisms could be explored to rectify this. The first one is the exploration and study of the computation method other than the ratio between positive and negative dependencies. The second one is to make the CPM values dynamic and learn through trials using reinforcement learning algorithms.

7.7 Conclusions

This paper presents a methodology to concurrently optimize a product design while identifying product modules. The methodology uses complex DSM to evaluate the combination potency between components based on their positive and negative dependencies. The combination potency is then summarized into a matrix called a combination potency matrix (CPM). The CPM is then integrated into a modified version of the NSGA-III to simultaneously accomplish the product module identification and layout design optimization of an autonomous greenhouse. The proposed methodology allowed us to find concepts of the layout of an autonomous greenhouse with different levels of modularization. The modularization process was able to reduce the number of physical components from 9 down to a minimum of 4 modules. The benefits of modularization for the autonomous greenhouse are more space for the growth of the plants, lower obstruction of the filming and illuminating of the camera and LED respectively, and larger pack-soil and water tank which means more seeds can be planted. The proposed methodology was able to obtain these benefits while ensuring the fulfillment of the functional requirements of the autonomous greenhouse. However, poor concepts were still found due to the conflict between modules and objectives functions as well as the definition of the dominance in the non-dominated sorting. To improve the proposed methodology some modifications as been suggested such as a dynamic CPM as well as including diversity among the set of modules.

7.8 Acknowledgments

The authors would like to acknowledge the financial support of the Fond de recherche du Québec – Nature et Technologies (FRQNT).

7.9 References

[1] J. M. Torry-Smith, A. Qamar, S. Achiche, J. Wikander, N. H. Mortensen, and C. During, "Challenges in designing mechatronic systems," Journal of Mechanical Design, vol. 135, no. 1, p. 011005, 2013.

[2] A. Mohebbi, L. Baron, S. Achiche, and L. Birglen, "Trends in concurrent, multi-criteria and optimal design of mechatronic systems: A review," in Innovative Design and Manufacturing (ICIDM), Proceedings of the 2014 International Conference on, 2014: IEEE, pp. 88-93.

[3] U. Chouinard, Y.-S. Law-Kam Cio, A. Vadean, and S. Achiche, "Concurrent modeling of positive and negative dependencies in the design structure matrix using complex numbers " presented at the DS 103: Proceedings of the 22nd International DSM Conference (DSM 2020), MIT, Cambridge, Massachusetts, October, 13-15, 2020.

[4] A. Mohebbi, S. Achiche, and L. Baron, "Multi-criteria fuzzy decision support for conceptual evaluation in design of mechatronic systems: a quadrotor design case study," Research in Engineering Design, vol. 29, no. 3, pp. 329-349, 2018/07/01 2018, doi: 10.1007/s00163-018-0287-6.

[5] J. K. Gershenson, G. J. Prasad, and Y. Zhang, "Product modularity: Definitions and benefits," Journal of Engineering Design, vol. 14, no. 3, pp. 295-313, 2003/09/01 2003, doi: 10.1080/0954482031000091068.

[6] F. A. S. Piran, D. P. Lacerda, L. F. R. Camargo, C. F. Viero, A. Dresch, and P. A. Cauchick-Miguel, "Product modularization and effects on efficiency: An analysis of a bus manufacturer using data envelopment analysis (DEA)," International Journal of Production Economics, vol. 182, pp. 1-13, 2016/12/01/ 2016, doi: https://doi.org/10.1016/j.ijpe.2016.08.008.

[7] S. K. Vickery, X. Koufteros, C. Dröge, and R. Calantone, "Product Modularity, Process Modularity, and New Product Introduction Performance: Does Complexity Matter?," Production and Operations Management, vol. 25, no. 4, pp. 751-770, 2016, doi: 10.1111/poms.12495.

[8] U. Chouinard, S. Achiche, and L. Baron, "Integrating negative dependencies assessment during mechatronics conceptual design using fuzzy logic and quantitative graph theory," Mechatronics, vol. 59, pp. 140-153, 2019/05/01/ 2019, doi: https://doi.org/10.1016/j.mechatronics.2019.03.009.

[9] J. M. Torry-Smith, N. H. Mortensen, and S. Achiche, "A proposal for a classification of product-related dependencies in development of mechatronic products," Research in Engineering Design, vol. 25, no. 1, pp. 53-74, 2014.

[10] K. Hölttä-Otto and O. de Weck, "Degree of Modularity in Engineering Systems and Products with Technical and Business Constraints," Concurrent Engineering, vol. 15, no. 2, pp. 113-126, 2007, doi: 10.1177/1063293x07078931.

[11] T. AlGeddawy, S. N. Samy, and H. ElMaraghy, "Best design granularity to balance assembly complexity and product modularity," Journal of Engineering Design, vol. 28, no. 7-9, pp. 457-479, 2017/09/02 2017, doi: 10.1080/09544828.2017.1325859.

[12] K. Hölttä, E. S. Suh, and O. de Weck, "Tradeoff between modularity and performance for engineered systems and products," in DS 35: Proceedings ICED 05, the 15th International Conference on Engineering Design, Melbourne, Australia, 15.-18.08. 2005, 2005, pp. 449-450 (exec. Summ.), full paper no. DS35_60. 1.

[13] J. K. Gershenson, G. J. Prasad, and Y. Zhang, "Product modularity: measures and design methods," Journal of Engineering Design, vol. 15, no. 1, pp. 33-51, 2004/02/01 2004, doi: 10.1080/0954482032000101731.

[14] J. Bonvoisin, F. Halstenberg, T. Buchert, and R. Stark, "A systematic literature review on modular product design," Journal of Engineering Design, vol. 27, no. 7, pp. 488-514, 2016/07/02 2016, doi: 10.1080/09544828.2016.1166482.

[15] T.-L. Yu, A. A. Yassine, and D. E. Goldberg, "A Genetic Algorithm for Developing Modular Product Architectures," in ASME 2003 International Design Engineering Technical

Conferences and Computers and Information in Engineering Conference, 2003, vol. Volume 3b: 15th International Conference on Design Theory and Methodology, pp. 515-524, doi: 10.1115/detc2003/dtm-48647. [Online]. Available: https://doi.org/10.1115/DETC2003/DTM-48647

[16] X. Xiaogang, L. Chao, Y. Jian, and C. Yahua, "An analytical method based on design structure matrix for modular identification," in 2006 7th International Conference on Computer-Aided Industrial Design and Conceptual Design, 17-19 Nov. 2006 2006, pp. 1-4, doi: 10.1109/CAIDCD.2006.329358.

[17] P. A. Wrigley, P. Wood, P. Stewart, R. Hall, and D. Robertson, "Module layout optimization using a genetic algorithm in light water modular nuclear reactor power plants," Nuclear Engineering and Design, vol. 341, pp. 100-111, 2019/01/01/ 2019, doi: https://doi.org/10.1016/j.nucengdes.2018.10.023.

[18] Q. Cheng, G. Zhang, P. Gu, and X. Shao, "A product module identification approach based on axiomatic design and design structure matrix," Concurrent Engineering, vol. 20, no. 3, pp. 185-194, 2012, doi: 10.1177/1063293x12453350.

[19] H.-E. Tseng, C.-C. Chang, and J.-D. Li, "Modular design to support green life-cycle engineering," Expert Systems with Applications, vol. 34, no. 4, pp. 2524-2537, 2008/05/01/ 2008, doi: https://doi.org/10.1016/j.eswa.2007.04.018.

[20] M. K. Paras, L. Wang, Y. Chen, A. Curteza, R. Pal, and D. Ekwall, "A Sustainable Application Based on Grouping Genetic Algorithm for Modularized Redesign Model in Apparel Reverse Supply Chain," Sustainability, vol. 10, no. 9, p. 3013, 2018.

[21] Y. Xiao, G. Luo, J. Ma, L. Hao, and H. Sun, "An approach for module decomposition based on fuzzy pattern recognition," in 2009 IEEE International Conference on Industrial Engineering and Engineering Management, 8-11 Dec. 2009 2009, pp. 1528-1532, doi: 10.1109/IEEM.2009.5373087.

[22] X. Meng, Z. Jiang, and G. Q. Huang, "On the module identification for product family development," The International Journal of Advanced Manufacturing Technology, vol. 35, no. 1, pp. 26-40, 2007/11/01 2007, doi: 10.1007/s00170-006-0712-2.

[23] Q. Shan, Y. Chen, and G. Yan, "Product Module Identification Based on PSO," in 2008 International Symposium on Information Science and Engineering, 20-22 Dec. 2008 2008, vol. 1, pp. 231-235, doi: 10.1109/ISISE.2008.261.

[24] Q. Shan and Y. Chen, "Research on Numerical Module Identification Approaches," in 2009 Eighth IEEE/ACIS International Conference on Computer and Information Science, 1-3 June 2009 2009, pp. 672-676, doi: 10.1109/ICIS.2009.72.

[25] Q. Shan and Y. Chen, "Harmony Search Algorithm and its Application to Product Module Identification," Advanced Materials Research, vol. 225-226, pp. 1100-1104, 2011, doi: 10.4028/www.scientific.net/AMR.225-226.1100.

[26] V. B. Kreng and T.-P. Lee, "Modular product design with grouping genetic algorithm—a case study," Computers & Industrial Engineering, vol. 46, no. 3, pp. 443-460, 2004/06/01/ 2004, doi: https://doi.org/10.1016/j.cie.2004.01.007.

[27] X. Xu, W. Zhang, and X. Ding, "Modular design method for filament winding process equipment based on GGA and NSGA-II," The International Journal of Advanced Manufacturing Technology, vol. 94, no. 5, pp. 2057-2076, 2018/02/01 2018, doi: 10.1007/s00170-017-0929-2.

[28] W. Wei, A. Liu, S. C. Y. Lu, and T. Wuest, "A multi-principle module identification method for product platform design," Journal of Zhejiang University-SCIENCE A, vol. 16, no. 1, pp. 1-10, 2015/01/01 2015, doi: 10.1631/jzus.A1400263.

[29] Y.-S. Law-Kam Cio, Y. Ma, A. Vadean, G. Beltrame, and S. Achiche, "Evolutionary layout design synthesis of an autonomous greenhouse using product-related dependencies," Artificial Intelligence for Engineering Design, Analysis and Manufacturing(AI EDAM), pp. (1-16), 2020, doi: https://doi.org/10.1017/S0890060420000384.

[30] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," IEEE Transactions on Evolutionary Computation, vol. 18, no. 4, pp. 577-601, 2014, doi: 10.1109/TEVC.2013.2281535.

[31] Y.-S. Law, Y. Ma, A. Vadean, G. Beltrame, and S. Achiche, "Objective reduction using axiomatic design & product-related dependencies: a layout synthesis of an autonomous greenhouse

case study," in Proceedings of the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, St. Louis, MO, USA, 2020 (Accepted).

CHAPTER 8 GENERAL DISCUSSION

The developed design tool presented in this thesis showed the potential of integrating product-related dependencies and evolutionary algorithms for the conceptual design phase. This tool explored the possible placements and choices of components for a mechatronic product and yielded near-optimal layout designs. Indeed, 50 near-optimal layout designs of an autonomous greenhouse were obtained in 15-20 minutes. The developed design tool has, however, many limitations. The most important one is the suitable adaptation of the optimization process to the problem, the preferences of the user, and the compatibility with other tools.

Considering that the performance of the algorithm depends on the problem at hand [54], a first recommendation would be to include more evolutionary algorithms [55]. At the moment, the tool has an implementation of GA, NSGA-II, and NSGA-III [56].

Second, the adaptation of the optimization process needs to be improved considering that optimization formulation can vary depending on the mechatronic system studied. For example, other mechatronic systems can be defined as a high-dimensional optimization problem where the challenges lie in managing the huge number of design parameters (i.e. “the curse of dimensionality”). Once again, some algorithms focus on solving these problems such as cooperative co-evolutionary algorithms [57]. A computer-aided design tool should be able to provide at least one optimization algorithm to solve at least one kind of optimization problem to ensure the design of a variety of mechatronic systems.

Third, the preferences of the user are defined before the optimization process. However, the number of optimal layout designs can be large enough for the user to struggle with the comparison and selection of designs. Hence, a post-optimization process [58, 59] can be used by the user to pinpoint the desired region of the Pareto front. Another possible complementary mechanism could be the relaxation of the dominance definition in algorithms using the non-dominated sorting algorithm [60]. The principle of the relaxation of the dominance is to classify some good solutions, but not optimal, as the best solutions of the generation. Hence, giving them a chance to improve through the follow-up generation.

Fourth, a deeper investigation of the modularization process based on product-related dependency management would be suggested. More precisely, the numerical value computation of the combination potency must be reviewed. At the moment, a simple ratio between positive and negative dependencies is used. However, it might not be the most adequate method to compute the combination potency. One might start this investigation by comparing complexity metrics []. Complexity metrics are used in engineering design to quantify the complexity of concepts to compare them.

Fifth, the design tool could be improved by interfacing it with other computer-aided design tools for the next phases of the product design process. Usually, the following step of the conceptual design is the detailed design. This phase starts with a better assessment of the shapes and dimensions of the components using computer-aided tools such as Catia [3]. Then, one could also run simulations on the concept.

By considering the computer-aided tool specification defined in section 4.1, it is possible to notice that the developed computer-aided tool fulfills the essential need. Indeed, the user can input the information necessary for the generation and evaluation of layout designs. Then, the computer-aided tool can generate at least one near-optimal layout design. The user can also visualize the layout design concept in 3D. As for the optional needs, the tool can achieve only one of these needs. It is possible to view more than one layout design. However, it is not possible to see two layout designs simultaneously through some sort of comparison interface. The other two optional needs that are under development concern the user interface and the knowledge transfer from this tool to another one. Indeed, the current user interface of the tool consists of the user writing directly in the code and defined the components, parameters, etc. Hence, the user interface cannot be easily used. Finally, the tool has not been interfaced with other tools. Indeed, the proposed layout designs are not saved into a standard format (e.g. .stl) which means that other tools cannot upload the files easily.

The learning curve associated with the use of the developed computer-aided tool is the main constraint of this research project. Ideally, the tool needs to be easy to learn and use. This constraint is mainly not respected considering that a proper user interface has not been implemented yet. However, the methodology can be learned with ease to a certain degree. Indeed, the learning curve of the methodology would not be too steep for someone who has experience in designing mechatronic systems. Even though the proposed methodology differentiates itself from others in many points, it is still based on common guidelines of engineering design.

Overall, the objectives set in Chapter 2 are achieved through the developed tools in this thesis. Indeed, the computer-aided tool is built along with a methodology to establish the problem statement and translate it into an optimization problem. Then, an evolutionary computing algorithm is used to solve the optimization and output a set of near-optimal layout designs within a reasonable amount of time (e.g. 15-20 min for 50 layout designs of an autonomous greenhouse). The only objective that is incomplete is the test and validation of the computer-aided tool. This objective is considered incomplete since that it is possible to confirm that the outputted layout design makes sense through a designer's expert knowledge. However, to truly define the impact and validity of the tool during the conceptual design, the product must be physically built and tested. Due to the lack of time and resources, the prototypes of the case study are unfinished and could not validate the computer-aided tools within this thesis. The validation of the tool could start by adapting our recent work [61] on prototyping an autonomous greenhouse for space biology uses of cheap off-the-shelf components. These components and experimental setup can be used to validate the layout designs obtained by the proposed tool. Then, a comparison between the different layout designs could be done in terms of the growth and health of the plants.

CHAPTER 9 CONCLUSION AND RECOMMANDATIONS

In this thesis, a computer-aided tool is made during the conceptual design phase. The tool focuses on the identification and exploitation of the product-related dependencies to choose and place the components within a product, in other words, to achieve the layout design of the product. This aims at helping the user generate and evaluate layout designs. The generating aspect of this process is automated with the aid of evolutionary computing to reduce the time-consuming and resource-consuming burden of the designer and engineers. For example, after a 15–20 minutes simulation, the user was able to obtain 50 layout designs of an autonomous greenhouse. However, the developed tools still have many challenges to overcome in order to improve the combination of the engineering design and the evolutionary computing domains. Considering that not all engineers and designers are experts in evolutionary computing, an effort must be done to adapt the evolutionary computing algorithm in solving different types of product design. On the other hand, the engineering design methodology must be developed while considering the optimization methods to obtain a high-quality product in a reasonable amount of time. Furthermore, the engineering design methodology must profit from the computer power, hence, part of these methodologies must be put into an intuitive software to improve the design experience and remote collaborative design.

In future works, the computer-aided tool must offer a more intuitive user interface with guidelines on the workflow of the methodology developed in this thesis. Furthermore, the tool must be able to handle and solve different types of product design problems. To achieve this, adding and adapting evolutionary computing algorithms is recommended. Complementary engineering design tools can also help in solving many types of problems. The computer-aided tool should also ensure the proper transfer of knowledge to other detailed design tools to improve the design process workflow. Finally, the impact and usefulness of the developed computer-aided tool must be evaluated in two ways. First by comparing the performance and development time using with and without the computer-aided tool. Second, by evaluating the performances of many mechatronic products developed with the aid of the tool.

REFERENCES

[1] S. Eppinger and K. Ulrich, *Product design and development*. McGraw-Hill Higher Education, 2015.

[2] G. Pahl and W. Beitz, *Engineering design: a systematic approach*. Springer Science & Business Media, 2013.

[3] D. systemes. "Catia." https://www.3ds.com/products-services/catia/products/ (accessed 2021).

[4] Siemens. "NX." https://www.plm.automation.siemens.com/global/en/products/nx/ (accessed 2021).

[5] Microsoft. "Microsoft Project." https://www.microsoft.com/en-ca/microsoft-365/project/project-management-software (accessed 2021).

[6] SAS. "JMP." https://www.jmp.com/en_us/software/data-analysis-software.html (accessed 2021).

[7] SysML. "SysML Open Source Project - What is SysML? Who created SysML?" https://sysml.org/ (accessed 2021).

[8] Altair. "Altair HyperStudy." https://www.altair.com/hyperstudy/ (accessed 2021).

[9] smartdraw. "Engineering Drawing Software." https://www.smartdraw.com/cad/engineering-drawing-software.htm (accessed 2021).

[10] H. Lipson and M. Shpitalni, "Conceptual design and analysis by sketching," *AI EDAM*, vol. 14, no. 5, pp. 391-401, 2000.

[11] T. Vuletic, A. Duffy, L. Hay, C. McTeague, L. Pidgeon, and M. Grealy, "The challenges in computer supported conceptual engineering design," *Computers in Industry*, vol. 95, pp. 22-37, 2018/02/01/ 2018, doi: https://doi.org/10.1016/j.compind.2017.11.003.

[12] A. Mohebbi, L. Baron, S. Achiche, and L. Birglen, "Trends in concurrent, multi-criteria and optimal design of mechatronic systems: A review," in *Innovative Design and Manufacturing (ICIDM), Proceedings of the 2014 International Conference on*, 2014: IEEE, pp. 88-93.

[13] J. M. Torry-Smith, A. Qamar, S. Achiche, J. Wikander, N. H. Mortensen, and C. During, "Challenges in designing mechatronic systems," *Journal of Mechanical Design*, vol. 135, no. 1, p. 011005, 2013.

[14] U. Chouinard, S. Achiche, and L. Baron, "Integrating negative dependencies assessment during mechatronics conceptual design using fuzzy logic and quantitative graph theory," *Mechatronics*, vol. 59, pp. 140-153, 2019/05/01/ 2019, doi: https://doi.org/10.1016/j.mechatronics.2019.03.009.

[15] J. M. Torry-Smith, N. H. Mortensen, and S. Achiche, "A proposal for a classification of product-related dependencies in development of mechatronic products," *Research in Engineering Design*, vol. 25, no. 1, pp. 53-74, 2014.

[16] A. Mohebbi, S. Achiche, and L. Baron, "Multi-criteria fuzzy decision support for conceptual evaluation in design of mechatronic systems: a quadrotor design case study," *Research in Engineering Design,* vol. 29, no. 3, pp. 329-349, 2018/07/01 2018, doi: 10.1007/s00163-018-0287-6.

[17] S. Chand and M. Wagner, "Evolutionary many-objective optimization: A quick-start guide," *Surveys in Operations Research and Management Science,* vol. 20, no. 2, pp. 35-42, 2015/12/01/ 2015, doi: https://doi.org/10.1016/j.sorms.2015.08.001.

[18] X. Chen, P. Wang, D. Zhang, and H. Dong, "Gradient-based multidisciplinary design optimization of an autonomous underwater vehicle," *Applied Ocean Research,* vol. 80, pp. 101-111, 2018/11/01/ 2018, doi: https://doi.org/10.1016/j.apor.2018.08.006.

[19] M. Bidoki, M. Mortazavi, and M. Sabzehparvar, "A new approach in system and tactic design optimization of an autonomous underwater vehicle by using Multidisciplinary Design Optimization," *Ocean Engineering,* vol. 147, pp. 517-530, 2018/01/01/ 2018, doi: https://doi.org/10.1016/j.oceaneng.2017.10.050.

[20] F. Guo, S. Zhao, X. Zhou, J. Fan, and D. Mi, "Conceptual Multidisciplinary Design Optimization of Commercial Aero-engine System," *IOP Conference Series: Materials Science and Engineering,* vol. 685, p. 012021, 2019/11/22 2019, doi: 10.1088/1757-899x/685/1/012021.

[21] X. Sun, J. Ge, T. Yang, Q. Xu, and B. Zhang, "Multifidelity Multidisciplinary Design Optimization of Integral Solid Propellant Ramjet Supersonic Cruise Vehicles," *International Journal of Aerospace Engineering,* vol. 2019, p. 5192424, 2019/04/28 2019, doi: 10.1155/2019/5192424.

[22] A. Sgueglia *et al.,* "Multidisciplinary Design Optimization Framework with Coupled Derivative Computation for Hybrid Aircraft," *Journal of Aircraft,* vol. 57, no. 4, pp. 715-729, 2020/07/01 2020, doi: 10.2514/1.C035509.

[23] Y. Werner, T. Vietor, M. Weinert, and T. Erber, "Multidisciplinary design optimization of a generic b-pillar under package and design constraints," *Engineering Optimization,* pp. 1-18, 2020, doi: 10.1080/0305215X.2020.1837791.

[24] K. Lynch, R. Ramsey, G. Ball, M. Schmit, and K. Collins, "Conceptual design acceleration for cyber-physical systems," in *2017 Annual IEEE International Systems Conference (SysCon),* 24-27 April 2017 2017, pp. 1-6, doi: 10.1109/SYSCON.2017.7934771.

[25] M. E. Amine, N. Perry, and J. Pailhès, "A risk-based approach to drive conceptual design taking into account low-maturity products," *International Journal on Interactive Design and Manufacturing (IJIDeM),* vol. 11, no. 3, pp. 677-688, 2017/08/01 2017, doi: 10.1007/s12008-016-0354-z.

[26] B. Chen and Y.-B. Xie, "A complex-number-domain-based conceptual design synthesis for multidisciplinary products," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* vol. 231, no. 7, pp. 1292-1307, 2017, doi: 10.1177/0954406216668207.

[27] B. Chen and Y. Xie, "A computational approach for the optimal conceptual design synthesis based on the distributed resource environment," *International Journal of Production*

Research, vol. 55, no. 20, pp. 5881-5901, 2017/10/18 2017, doi: 10.1080/00207543.2017.1302619.

[28] Y. Yi, W. Li, and M. Xiao, "Mode Pursuing Sampling Method for Multidisciplinary Deisgn Optimization in Ship Conceptual Design," in *2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD))*, 9-11 May 2018 2018, pp. 559-563, doi: 10.1109/CSCWD.2018.8465215.

[29] H. Hong, Z. Jiang, and Y. Yin, "An intelligent conceptual design framework for complex machines," *Procedia CIRP*, vol. 72, pp. 586-591, 2018/01/01/ 2018, doi: https://doi.org/10.1016/j.procir.2018.03.041.

[30] N. Jelev, A. Keane, and C. Holden, "Pattern Search Algorithm for Blackboard-Based Multidisciplinary Design Optimization Frameworks," *Journal of Aircraft*, vol. 56, no. 1, pp. 121-136, 2019/01/01 2018, doi: 10.2514/1.C034897.

[31] Z. Hu, C. Rao, C. Tao, P. R. N. Childs, and Y. Zhao, "A case-based decision theory based process model to aid product conceptual design," *Cluster Computing*, vol. 22, no. 4, pp. 10145-10162, 2019/07/01 2019, doi: 10.1007/s10586-017-1190-z.

[32] S. G. Kontogiannis and M. A. Savill, "A generalized methodology for multidisciplinary design optimization using surrogate modelling and multifidelity analysis," *OPTIMIZATION AND ENGINEERING*, 2020.

[33] U. Chouinard, S. Achiche, C. Leblond-Ménard, and L. Baron, "Assessment of dependencies in mechatronics conceptual design of a quadcopter drone using linguistic fuzzy variables," in *DS 87-4 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 4: Design Methods and Tools, Vancouver, Canada, 21-25.08. 2017*, 2017.

[34] A. Drira, H. Pierreval, and S. Hajri-Gabouj, "Facility layout problems: A survey," *Annual Reviews in Control*, vol. 31, no. 2, pp. 255-267, 2007, doi: 10.1016/j.arcontrol.2007.04.001.

[35] G. Moslemipour, T. S. Lee, and D. Rilling, "A review of intelligent approaches for designing dynamic and robust layouts in flexible manufacturing systems," *The International Journal of Advanced Manufacturing Technology*, vol. 60, no. 1-4, pp. 11-27, 2012.

[36] A. Ahmadi, M. S. Pishvaee, and M. R. Akbari Jokar, "A survey on multi-floor facility layout problems," *Computers & Industrial Engineering*, vol. 107, pp. 158-170, 2017, doi: 10.1016/j.cie.2017.03.015.

[37] H. Fukushima, K. Kon, and F. Matsuno, "Model predictive formation control using branch-and-bound compatible with collision avoidance problems," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1308-1317, 2013.

[38] G.-S. Liu, B.-X. Zhang, H.-D. Yang, X. Chen, and G. Q. Huang, "A branch-and-bound algorithm for minimizing the energy consumption in the PFS problem," *Mathematical Problems in Engineering*, vol. 2013, 2013.

[39] G. Belov and G. Scheithauer, "A cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths," *European Journal of Operational Research*, vol. 141, no. 2, pp. 274-294, 2002.

[40] J. M. Belenguer and E. Benavent, "A cutting plane algorithm for the capacitated arc routing problem," *Computers & Operations Research*, vol. 30, no. 5, pp. 705-728, 2003.

[41] Z. Qian *et al.*, "Expert-guided evolutionary algorithm for layout design of complex space stations," *Enterprise Information Systems*, vol. 11, no. 7, pp. 1078-1093, 2017.

[42] Q.-L. Lin, H.-C. Liu, D.-J. Wang, and L. Liu, "Integrating systematic layout planning with fuzzy constraint theory to design and optimize the facility layout for operating theatre in hospitals," *Journal of Intelligent Manufacturing*, vol. 26, no. 1, pp. 87-95, 2013, doi: 10.1007/s10845-013-0764-8.

[43] N. Zhou, A. Agogino, and K. S. Pister, "Automated design synthesis for micro-electro-mechanical systems (MEMS)," in *ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2002: American Society of Mechanical Engineers, pp. 267-273.

[44] B. Chen and L. Deng, "The Application of Genetic Algorithm in Layout Design of Oil Rig Driller Console," *Advanced Materials Research*, vol. 216, pp. 171-175, 2011, doi: 10.4028/www.scientific.net/AMR.216.171.

[45] S. Behbahani and C. W. de Silva, "Mechatronic Design Evolution Using Bond Graphs and Hybrid Genetic Algorithm With Genetic Programming," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 1, pp. 190-199, 2013, doi: 10.1109/tmech.2011.2165958.

[46] L. Xiong, R. Molfino, and M. Zoppi, "Fixture layout optimization for flexible aerospace parts based on self-reconfigurable swarm intelligent fixture system," *The International Journal of Advanced Manufacturing Technology*, vol. 66, no. 9-12, pp. 1305-1313, 2013.

[47] L. Deng, G. Wang, and S. Yu, "Layout Design of Human-Machine Interaction Interface of Cabin Based on Cognitive Ergonomics and GA-ACA," *Comput Intell Neurosci*, vol. 2016, p. 1032139, 2016, doi: 10.1155/2016/1032139.

[48] M. M. Andreasen, "Machine design methods based on a systematic approach–Contribution to a design theory," *Diss. Lund Institute of Technology*, 1980.

[49] M. M. Andreasen and T. C. McAloone, "" Joining three heads"-experiences from mechatronic projects," in *DFX 2001: Proceedings of the 12th Symposium on Design for X, Neukirchen/Erlangen, Germany, 11.-12.10. 2001*, 2001.

[50] M. Danilovic and T. R. Browning, "Managing complex product development projects with design structure matrices and domain mapping matrices," *International journal of project management*, vol. 25, no. 3, pp. 300-314, 2007.

[51] U. Lindemann, M. Maurer, and T. Braun, *Structural complexity management: an approach for the field of product design*. Springer Science & Business Media, 2008.

[52] J. Gausemeier, U. Frank, J. Donoth, and S. Kahl, "Specification technique for the description of self-optimizing mechatronic systems," *Research in Engineering Design*, journal article vol. 20, no. 4, p. 201, February 05 2009, doi: 10.1007/s00163-008-0058-x.

[53] D. Tang, G. Zhang, and S. Dai, "Design as integration of axiomatic design and design structure matrix," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 3, pp. 610-619, 2009.

[54] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation,* vol. 1, no. 1, pp. 67-82, 1997.

[55] W. Huang, Y. Zhang, and L. Li, "Survey on Multi-Objective Evolutionary Algorithms," *Journal of Physics: Conference Series,* vol. 1288, p. 012057, 2019/08 2019, doi: 10.1088/1742-6596/1288/1/012057.

[56] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," *IEEE Transactions on Evolutionary Computation,* vol. 18, no. 4, pp. 577-601, 2014, doi: 10.1109/TEVC.2013.2281535.

[57] X. Ma *et al.*, "A Survey on Cooperative Co-Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation,* vol. 23, no. 3, pp. 421-441, 2019, doi: 10.1109/TEVC.2018.2868770.

[58] H. Wang, M. Olhofer, and Y. Jin, "A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges," *Complex & Intelligent Systems,* vol. 3, no. 4, pp. 233-245, 2017/12/01 2017, doi: 10.1007/s40747-017-0053-9.

[59] G. Yu, Y. Jin, and M. Olhofer, "References or Preferences – Rethinking Many-objective Evolutionary Optimization," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, 10-13 June 2019 2019, pp. 2410-2417, doi: 10.1109/CEC.2019.8790106.

[60] L. S. Batista, F. Campelo, F. G. Guimarães, and J. A. Ramírez, "A comparison of dominance criteria in many-objective optimization problems," in *2011 IEEE Congress of Evolutionary Computation (CEC)*, 5-8 June 2011 2011, pp. 2359-2366, doi: 10.1109/CEC.2011.5949909.

[61] C. M. Trouillefou *et al.*, "An autonomous plant growing miniaturized incubator for a Cubesat," *Acta Astronautica,* vol. 179, pp. 439-449, 2021/02/01/ 2021, doi: https://doi.org/10.1016/j.actaastro.2020.11.009.