North Carolina State University
Department of Mechanical and Aerospace Engineering

Valencia Polytechnic University
Higher Technical School of Design Engineering

EXPLORING SET-BASED OPTIMIZATION STRATEGIES FOR MARKET-DRIVEN DESIGN PROBLEMS

Master's thesis

Master in Aerospace Engineering

Author: *María José Marco Cutillas*
Tutor: *Scott Ferguson*
Cotutor: *José Martínez Casas*

North Carolina - May 18, 2023

Abstract

Product designers face the challenge of creating products in markets where customers have highly heterogeneous preferences. Such is the world of aircraft manufacturers, for example, where everything from small regional aircraft to large transatlantic aircraft are sought. Improving performance in market-related objectives, such as market share of preference, requires a product line (a set of related products that are offered by a single company). However, it is difficult to optimize the product line design problem due to the expansive mixed-integer design space. Advances in optimization techniques have been proposed to aid in the navigation of such large spaces so that the optimal solution can be found at a low computational cost. Yet, product line design problems still require many evaluations. What is more, many of these evaluations are spent driving the solution from one that is very close to optimal to one that is optimal. The product configuration team then uses the optimization results as a basis for allocating resources and building products. Often, these optimized results are not used directly since there are unmodeled aspects that need to be considered. Therefore, the effort spent taking the solution from very close to optimal to one that is optimal may be wasted.

This project is going to explore how the optimization problem for a market-driven design problem can be reformulated so that a set of optimal solutions is returned, rather than a single optimal point. This set of solutions can then be used as the starting point for the discussion by the product configuration team instead of a fully optimized result that will be changed anyways. In the context of optimization algorithms, the final solution would be the schema that characterizes the qualities of a *good solution*. For this purpose, several simulations are performed by means of the software *MATLAB*.

In the specific product line design problem employed, the customer preferences are represented using a hierarchical Bayes mixed logit model and the design space is a mixed-integer formulation. After doing an exploration of the space, the fully optimized product line solution is found using a genetic algorithm. In addition, a modified optimization problem is formulated in order to identify the best set of solutions to the problem and its specific qualities. Finally, both solutions will be compared.

The procedure used in the project can be extrapolated and used in multiple areas, including aviation. In fact, in the aerospace universe it can be used when proposing a line of aircraft by a manufacturer. With the formulation of the modified optimization problem, it would be possible to offer the characteristics and properties that the airplanes must present to have better market shares and be purchased more easily by the various airlines. This is just one example of the various applications that the optimization process developed in this paper allows.

Key words: Optimization, optimal solutions, solution characteristics, algorithm.

Resumen

Los diseñadores de productos se enfrentan al reto de crear productos en mercados donde los clientes tienen preferencias muy heterogéneas. Como es por ejemplo, el mundo de los fabricantes de aviones, donde se buscan desde aviones pequeños regionales hasta aviones grandes para vuelos transatlánticos. Para mejorar los resultados en los objetivos relacionados con el mercado, como la cuota de mercado de preferencia, se requiere una línea de productos (un conjunto de productos relacionados que ofrece una única empresa). Sin embargo, es difícil optimizar el problema del diseño de la línea de productos debido al extenso espacio de diseño mixto-entero. Se han propuesto avances en las técnicas de optimización para ayudar en la navegación de espacios tan grandes, de modo que se pueda encontrar la solución óptima con un bajo coste computacional. Sin embargo, los problemas de diseño de líneas de productos siguen requiriendo muchas evaluaciones. Es más, muchas de estas evaluaciones se emplean en conducir la solución desde una que está muy cerca de ser óptima hasta una que es óptima. A continuación, el equipo de configuración del producto utiliza los resultados de la optimización como base para asignar recursos y construir productos. A menudo, estos resultados optimizados no se utilizan directamente, ya que hay aspectos no modelados que deben tenerse en cuenta. Por lo tanto, el esfuerzo invertido en llevar la solución desde una muy cerca del óptimo a una que es óptima puede desperdiciarse.

Este proyecto va a explorar cómo el problema de optimización para un problema de diseño del mercado se puede reformular para que se devuelva un conjunto de soluciones óptimas, en lugar de un único punto óptimo. Este conjunto de soluciones se puede utilizar como punto de partida para la discusión por parte del equipo de configuración del producto. En el contexto de algoritmos de optimización como un algoritmo genético, la solución final sería el esquema que caracteriza las cualidades de una *buena solución.* Este esquema podría luego ser presentado al equipo de configuración del producto, en lugar de un resultado completamente optimizado que se cambiará de todos modos.

En el problema específico de diseño de líneas de productos empleado, las preferencias de los clientes se representan mediante un modelo logit mixto Bayes jerárquico y el espacio de diseño es una formulación mixta entera. Tras realizar una exploración del espacio, se encuentra la solución de línea de producto totalmente optimizada mediante un algoritmo genético. Además, se formula un problema de optimización modificado para identificar el mejor conjunto de soluciones del problema y sus cualidades específicas. Por último, se comparan ambas soluciones.

El procedimiento empleado en el proyecto puede extrapolarse y usarse en múltiples áreas, incluida la aviación . De hecho, en el universo aeroespacial se puede usar a la hora de proponer una línea de aviones por un fabricante. Con la formulación del problema de optimización modificada se podría ofrecer las características y propiedades que los aviones deben presentar para tener mejores cuotas de mercado y ser comprados con mayor facilidad por las diversas aerolíneas. Este es solo un ejemplo de las diversas aplicaciones que permite el proceso de optimización desarrollado en el presente documento.

Palabras clave: Optimización, soluciones óptimas, características de la solución, algoritmo.

Resum

Els dissenyadors de productes s'enfronten al repte de crear productes en mercats on els clients tenen preferències molt heterogènies. Com és per exemple, el món dels fabricants d'avions, on es busquen des d'avions xicotets regionals fins a avions grans per a vols transatlàntics. Per a millorar els resultats en els objectius relacionats amb el mercat, com la quota de mercat de preferència, es requereix una línia de productes (un conjunt de productes relacionats que ofereix una única empresa). No obstant això, és difícil optimitzar el problema del disseny de la línia de productes a causa de l'extens espai de disseny mixt-sencer. S'han proposat avanços en les tècniques d'optimització per a ajudar en la navegació d'espais tan grans, de manera que es puga trobar la solució òptima amb un baix cost computacional. No obstant això, els problemes de disseny de línies de productes continuen requerint moltes avaluacions. És més, moltes d'aquestes avaluacions s'empren a conduir la solució des d'una que està molt prop de ser òptima fins a una que és òptima. A continuació, l'equip de configuració del producte utilitza els resultats de l'optimització com a base per a assignar recursos i construir productes. Sovint, aquests resultats optimitzats no s'utilitzen directament, ja que hi ha aspectes no modelats que han de tindre's en compte. Per tant, l'esforç invertit a portar la solució des d'una molt prop de l'òptim a una que és òptima pot malgastarse.

Aquest projecte explorarà com el problema d'optimització per a un problema de disseny del mercat es pot reformular perquè es retorne un conjunt de solucions òptimes, en lloc d'un únic punt òptim. Aquest conjunt de solucions es pot utilitzar com a punt de partida per a la discussió per part de l'equip de configuració del producte. En el context d'algorismes d'optimització com un algorisme genètic, la solució final seria l'esquema que caracteritza les qualitats d'una *bona solució*. Aquest esquema podria després ser presentat a l'equip de configuració del producte, en lloc d'un resultat completament optimitzat que es canviarà de totes maneres.

En el problema específic de disseny de línies de productes emprat, les preferències dels clients es representen mitjançant un model logit mixt Bayes jeràrquic i l'espai de disseny és una formulació mixta sencera. Després de realitzar una exploració de l'espai, es troba la solució de línia de producte totalment optimitzada mitjançant un algorisme genètic. A més, es formula un problema d'optimització modificat per a identificar el millor conjunt de solucions del problema i les seues qualitats específiques. Finalment, es comparen totes dues solucions.

El procediment emprat en el projecte pot extrapolar-se i usar-se en múltiples àrees, inclosa l'aviació . De fet, en l'univers aeroespacial es pot usar a l'hora de proposar una línia d'avions per un fabricant. Amb la formulació del problema d'optimització modificada es podria oferir les característiques i propietats que els avions han de presentar per a tindre millors quotes de mercat i ser comprats amb major facilitat per les diverses aerolínies. Aquest és només un exemple de les diverses aplicacions que permet el procés d'optimització desenvolupat en el present document.

Paraules clau: Optimització, solucions òptimes, característiques de la solució, algorisme.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

AC	Acquisition Cost
DM	Displacement mutation
HB	Hierarchical Bayes
MDO	Multidisciplinary Design Optimization
MSRP	Manufacturer's Suggested Retail Price
NCSU	North Carolina State University
OX	Order Crossover
PBD	Point-Based Design
PMX	Partially Matched Crossover
SBCE	Set-Based Concurrent Engineering
SBD	Set-Based Design
SIM	Simple Inversion Mutation
SM	Scramble Mutation
SUS	Stochastic Universal Sampling
UPV	Polytechnic University of Valencia

Part I

MEMORANDUM

Introduction

1.1. Motivation

Product designers face the challenge of creating products in markets where customers have highly heterogeneous preferences. Improving performance in market-related objectives, such as market share of preference, requires a product line (a set of related products that are offered by a single company). However, optimizing product line design problems is challenging because there is an expansive mixed-integer design space. Advancements to optimization techniques have been proposed that aid in the navigation of such large spaces so that the optimal solution can be found with a reduced computational expense. Yet, even with advancements in optimization techniques, product line design problems still require many evaluations. Many of these evaluations are spent driving the solution from one that is *very close to optimal* to one that is *optimal*.

Optimization results are then used by a product configuration team as a discussion point toward allocating resources and building products. Often, these optimized results are not used directly. Rather, there are unmodeled aspects of the optimization problem that are taken into account. Therefore, the effort spent taking the solution from *very close to optimal* to one that is *optimal* may be wasted.

Therefore, this project intends to investigate how the optimization problem for a market-driven design problem can be formulated differently so that a set of optimal solutions is returned, instead of a unique optimal point. This group of solutions can then be employed as the starting point for the debate by the product configuration team, rather than a completely optimized result that will be modified regardless. In the context of optimization algorithms like a genetic algorithm, the final solution would be the schema that describes the characteristics of a *good solution*.

In essence, what is driving this work is the necessity to identify the properties of a good solution for a product line design problem with the objective of presenting this information to the end user. As a consequence, in this project, a modified optimization problem is formulated in order to identify the best set of solutions to the problem and its specific qualities.

The remainder of the project is structured as follows. First, a literature review is done in Section 1.2 where Set-Based Design (SBD) theory, principles and contributions are exposed. Then the specific objectives of this project are disclosed in Section 1.3 to complete the introduction of the topic. In Section 3 the method employed is explained. The results are discussed in Section 4 and finally, the article is concluded in Section 5.

1.2. Background

According to Clark Fujimoto [1], the earliest decisions in product development have the largest impact on the overall quality of the product (effectiveness) and the overall cost of the project (efficiency). Many approaches to engineering design are focused on reducing cycle time following the famous motto *do it right the first time*. In terms of design strategy, this has often been translated into a need to propose the right solution as fast as possible. As observed by Sobek Ward [2], when dealing with the development of complex products, many companies force the engineering teams to propose a feasible concept quickly so that it can then be optimized through numerous iteration loops. This pattern is understood as Point-Based Design (PBD) because it focuses on one solution at a time which is improved iteratively until a satisfactory design that meets the design requirements is reached. Figure 1.1 illustrates this view. [3] PBD allows the design engineers to arrive at a solution in a short time frame. However, once the design engineers commit to a solution in the design phase, it becomes difficult to modify the design in case the system requirements change during the later stages of the product development process. A possible remedy to the above shortcomings is to delay commitment to a single design early in the design stage. [4]

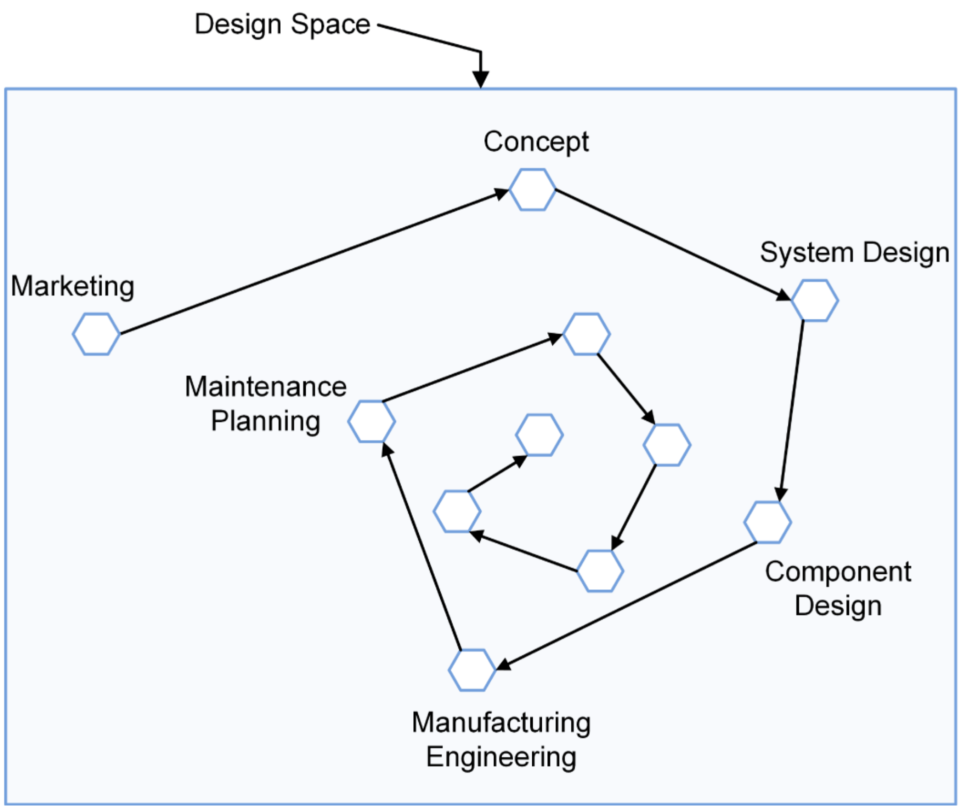


Figure 1.1: A point-based approach. [3]

Another design strategy, the set-based approach, has been the subject of a number of publications over the past 20 years. It is one of the pillars of lean thinking applied to product development, observed particularly in the automotive industry through companies such as Toyota, Honda, and Denso. [3] In fact, the term Set-Based Concurrent Engineering (SBCE) was introduced by Ward as a name for Toyota's method of managing product development processes. We use the equivalent term Set-Based Design (SBD). [5] Here, engineers may reason and communicate about an acceptable range of parametric values instead of the single best value at a time. SBD allows windows of possibilities to align gradually and therefore the best of all worlds to be projected. It is rather a convergence process than an evolution. Participants bring sets of possibilities to the table and juxtapose them to find an intersection of feasibility rather than successively criticizing and modifying a single option. Figure 1.2 illustrates the approach. [3]

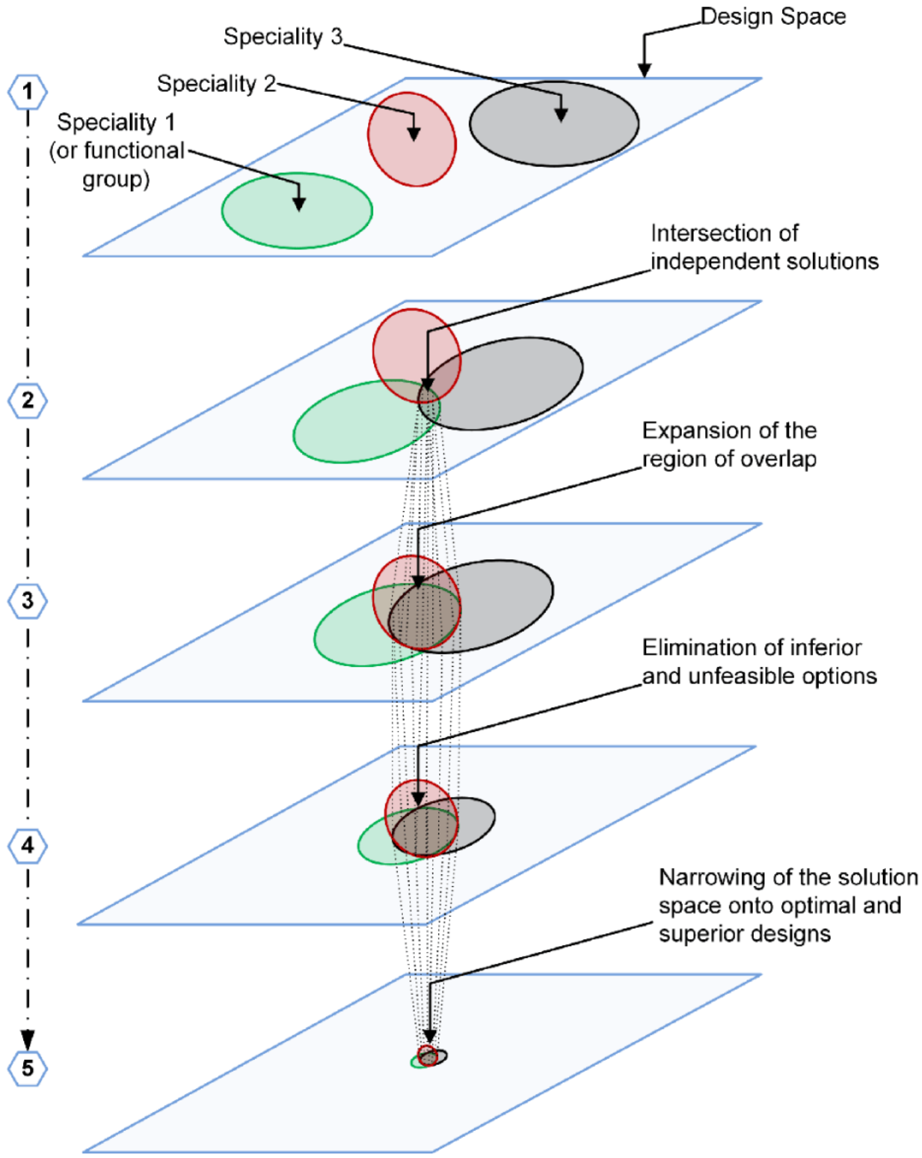


Figure 1.2: A set-based approach. [3]

SBD is based on three principles: map the design space, integrate by intersection and establish feasibility before commitment. Mapping the design space is identifying feasible designs comprising the feasible design set with respect to each design requirement and quantifying trade-offs between possible design solutions. [4] Integrating by intersection means looking for solutions within the intersections of sets or intervals imposing minimum constraint. Finally, establishing feasibility before commitment refers to the obligation of a design contributor to maintaining consistency with the preexisting design. This is radically different from point-based design, in which each design contribution may invalidate all previous work. [5] The feasible design space is gradually narrowed down by eliminating undesirable solutions as design requirements become more well-defined and constraints are tightened. [4] So in essence, SBD is trying to eliminate the weak options instead of directly picking the best one.

SBD is said to have many benefits. For example, it wastes little time on detailed designs that can't be built, it also helps delay decisions on variable values until they become essential for the completion of the project or in this approach the initiator of a change retains responsibility for maintaining consistency on the contrary to PBD, where each change may invalidate all previous decisions. [5]

The majority of publications about SBD consists of research papers that combine SBCE principles with complementary design methods to improve the design or development process. The publications in this category frequently focus on the design synthesis and selection process by portraying how the other method could adequately fit the SBCE concept generation, exploration and selection process. This denotes a growing interest in SBCE from diverse field areas. It also suggests a pervasive influence of SBCE, especially on methods and techniques for the early stages of the development process. The closest to what is being done in this project would include the combination of SBCE principles and techniques with a multi-objective problem or a multidisciplinary optimization. [3] Several real-world applications can be formulated as optimization problems. They are often characterized by multiple conflicting objectives and a wide range of uncertainties that have to be taken into account. [6]

For example, Hannapel Vlahopoulos [7] present the development and application of a new multidisciplinary design optimization (MDO) algorithm inspired by the principles of set-based design. The new MDO algorithm was developed with the core concept of describing the design using sets to incorporate features of set-based design and achieve greater flexibility than with a single-point optimization. Malak et al. [8] use utility theory to make set-based decisions. The interval dominance criterion was used to eliminate designs when there is no overlap in their uncertainty ranges. The maximality criterion was used to make decisions involving design variables with overlapping uncertainty intervals. Nahm and Ishikawa [9] accommodate designer preferences in the form of preference numbers and functions. The designer's preference structure spans design variables and requirements which may be a product of multidisciplinary analyses. However, the approach may not span arbitrarily shaped design spaces. Wang and Terpenny [10] used a genetic algorithm to evaluate alternative design trade-offs in a component-based system synthesis problem. [4]

1.3. Objectives

The main objective of this Master's thesis is to explore how the optimization problem for a market-driven design problem can be reformulated so that a set of optimal solutions is returned, rather than a single optimal point. This set of solutions can then be used as the starting point for the discussion by the product configuration team. In the context of optimization algorithms like a genetic algorithm, the final solution would be the schema that characterizes the qualities of a *good solution*. This schema could then be presented to the product configuration team, instead of a fully optimized result that will be changed anyways. For this purpose, several simulations are performed by means of the *software MATLAB*.

In order to meet these main objectives, a series of fundamental secondary objectives must first be completed:

- Understand and analyze the formulation of a product line design problem where customer preferences are represented using a hierarchical Bayes mixed logit model and the design space is a mixed-integer formulation. The particular problem definition used as a general model in this project was configured in prior work by the department.
- Execute the problem formulation several times for a large random sample of product lines to have awareness of how is the specific problem that is trying to be optimized.
- Identified the fully optimized product line solution. The optimal solution to the problem is computed in order to serve as a baseline for comparison and to have a first insight into how the set of solutions should be.
- Formulate a modified optimization problem focused on identifying the best set (schema) of solutions to the product line design problem.
- Evaluate how the schema solutions compare against the fully optimized product line solution.

After defining in detail the technical objectives of the project, it is necessary to set out purely academic objectives. As far as this is concerned, the main objective is to allow the author of the document to finish his Master's degree in Aerospace Engineering accredited by the Polytechnic University of Valencia. Furthermore, it is intended to apply appropriately and correctly some of the knowledge and skills acquired during this period with the ultimate goal of demonstrating that the training has been properly assimilated.

Finally, some professional objectives also need to be stated. The main thing regarding this aspect is that a professional environment has been achieved. This, in addition, has made it necessary to learn a daily methodology within a schedule and to develop the ability to meet several weekly items to be presented to the person in charge.

Theoretical fundaments

2.1. Genetic Algorithm

Genetic Algorithm (GA) is one of the first population-based stochastic algorithmS proposed in history. [11] Genetic algorithm is inspired by the biological evolution process and mimics the Darwinian theory of survival of the fittest in nature. GA was proposed by J.H. Holland in 1992 and its basic elements are chromosome representation, fitness selection, and biological-inspired operators. The biological-inspired operators are selection, mutation, and crossover. [12] Genetic algorithms are often viewed as function optimizers, although the range of problems to which genetic algorithms have been applied is quite broad. [13]

GA is a population-based algorithm and the new populations are produced by iterative use of genetic operators on individuals present in the population. [12] The starting population is a random population. This population can be generated from a Gaussian random distribution to increase the diversity. The main objective in the initialization step is to spread the solutions around the search space as uniformly as possible to increase the diversity of the population and have a better chance of finding promising regions. [11] Typically, the chromosomes, considered as points in the solution space, take the string format. In chromosomes, each locus i.e. specific position on a chromosome has different possible alleles, meaning variant forms of genes. [12] View Figure 2.1 for a wider understanding. The fitness (objective) function is used to assign a value evaluating the fitness of each individual for all the chromosomes in the population. For improving poor solutions, the best solutions are chosen randomly with a selection mechanism based on their fitness value. What increases local optima avoidance is the probability of choosing poor solutions as well. [11]

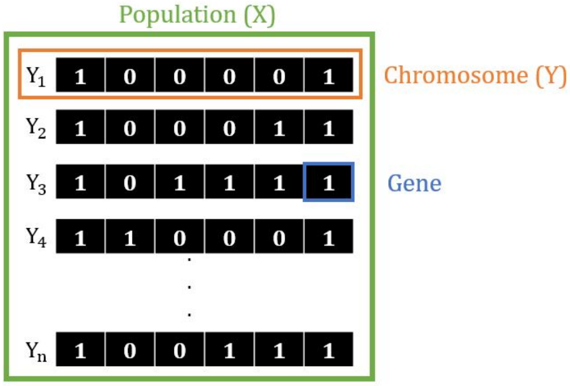


Figure 2.1: Diagram of population, chromosome and gene.

The GA algorithm is stochastic, so one might ask how reliable it is. What makes this algorithm reliable and able to estimate the global optimum for a given problem is the process of maintaining the best solutions in each generation and using them to improve other solutions. As such, the entire population becomes better generation by generation. The crossover between individuals results in exploiting the *area* between the given two parent solutions to create better off-springs. This algorithm also benefits from mutation. This operator randomly changes the genes in the chromosomes, which maintains the diversity of the individuals in the population and increases the exploratory behaviour of GA. Similar to nature, the mutation operator might result in a substantially better solution and lead other solutions towards the global optimum. [11]

In other words, the procedure of GA is as follows. A population (X) of n chromosomes is initialized randomly. The fitness of each chromosome in X is computed. Two chromosomes say Y_1 and Y_2 are selected from the population X according to the fitness value. A crossover operator with crossover probability (C_p) is applied on Y_1 and Y_2 to produce an offspring say Z. Thereafter, a mutation operator is applied on produced offspring (Z) with mutation probability (M_p) to generate Z'. The new offspring Z' is placed in the new population. The selection, crossover, and mutation operations will be repeated on the current population until the new population is complete. [12] Refer to the Figure 2.2 for a clearer view of the process.

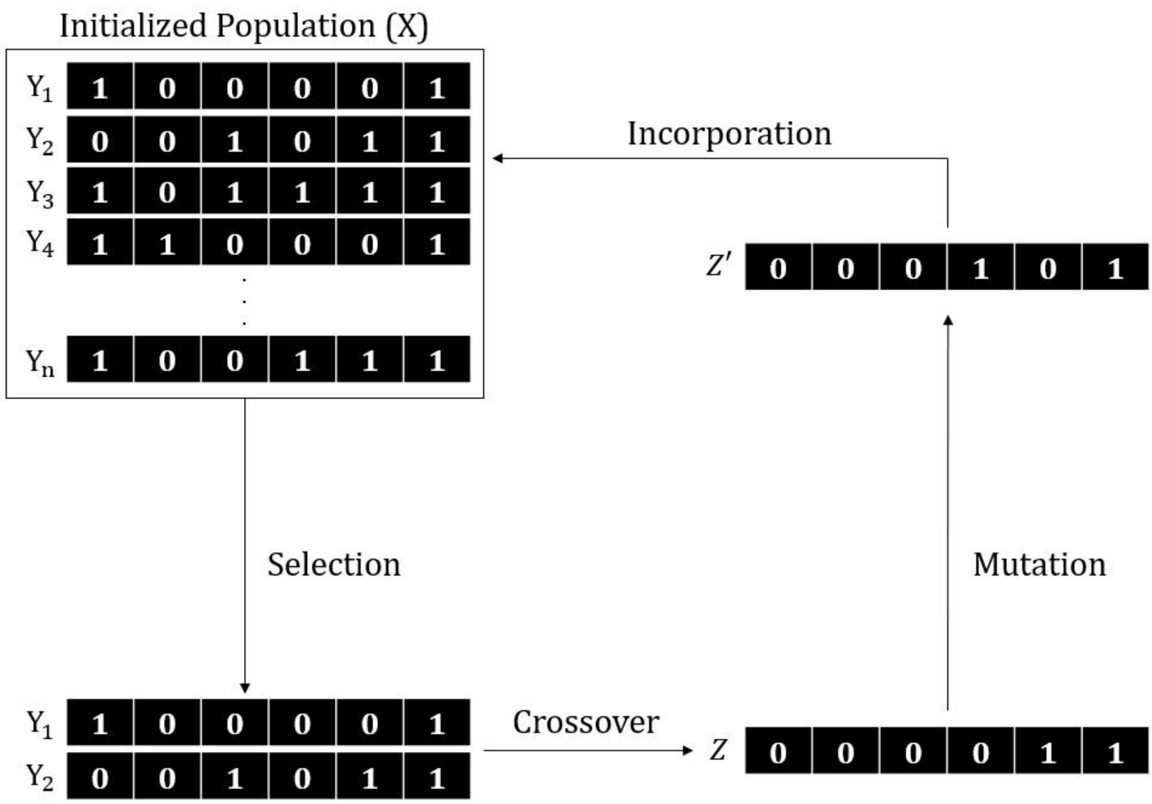


Figure 2.2: Procedure of GA.

GAs used a variety of operators during the search process. These operators are encoding schemes, selection, crossover and mutation. Figure 2.3 depicts the operators used in GAs.In the next sections, they are going to be explained in more detail. [12]

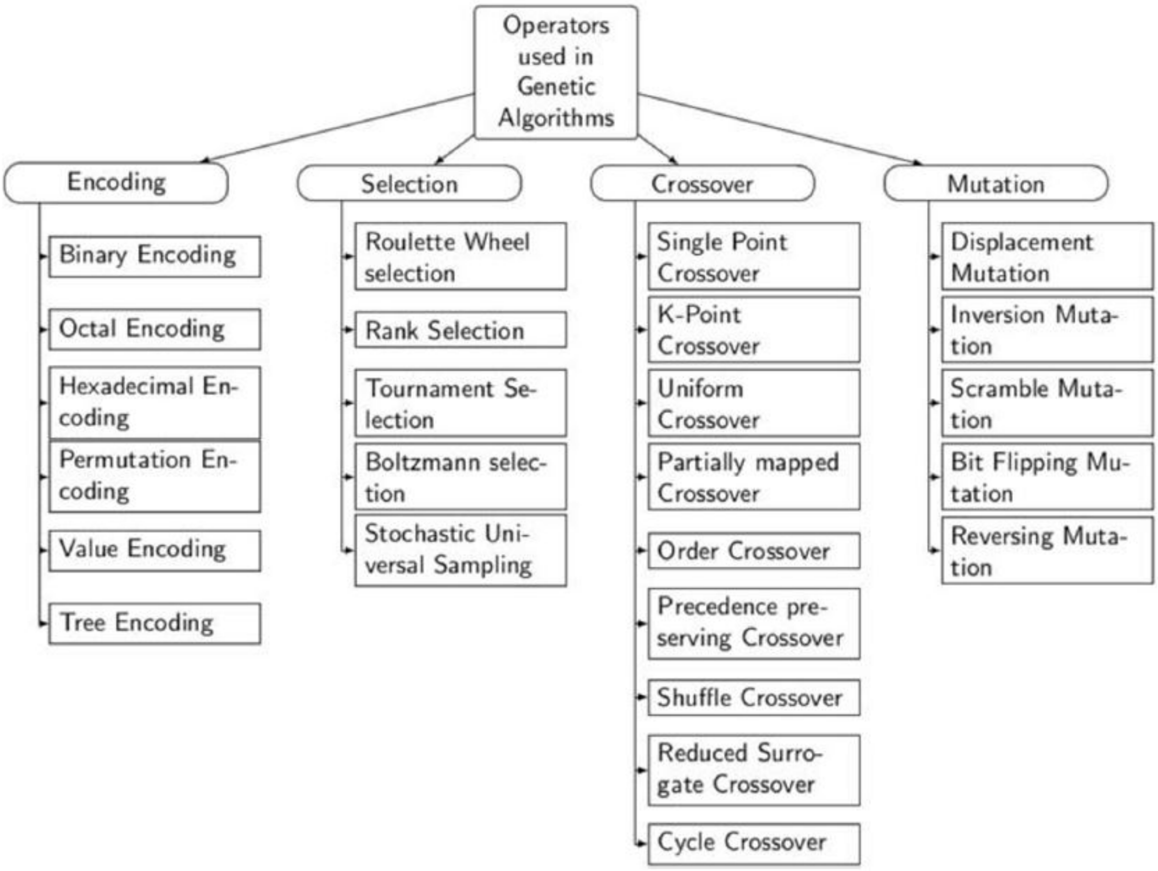


Figure 2.3: Operators used in GA. [12]

2.1.1. Encoding schemes

For most computational problems, the encoding scheme plays an important role. The given information has to be encoded in a particular bit string. The encoding schemes are differentiated according to the problem domain. The well-known encoding schemes are binary, octal, hexadecimal, permutation, value-based and tree. [12]

Binary encoding is the commonly used encoding scheme. Each gene or chromosome is represented as a string of 1 or 0. In binary encoding, each bit represents the characteristics of the solution. In the octal encoding scheme, the gene or chromosome is represented in the form of octal numbers (0–7). On the other hand, in the hexadecimal encoding scheme, the gene or chromosome is represented in the form of hexadecimal numbers (0–9, A-F). The permutation encoding scheme is generally used in ordering problems. In this encoding scheme, the gene or chromosome is represented by the string of values that represents the position in a sequence. These values can be real, integer numbers, or characters. This encoding scheme can be helpful in solving problems where more complicated values are used, since binary encoding may fail in such problems. In tree encoding, the gene or chromosome is represented by a tree of functions or commands. These functions and commands can be related to any programming language. [12]

2.1.2. Selection techniques

Selection is an important step in genetic algorithms that determines whether a particular string will participate in the reproduction process or not. The selection step is sometimes also known as the reproduction operator. The convergence rate of GA depends upon the selection pressure. The well-known selection techniques are the roulette wheel, rank, tournament, stochastic universal sampling and Boltzmann. [12]

Roulette wheel selection maps all the possible strings onto a wheel and assigns probabilities proportional to their fitness (objective) values to each individual. This wheel is then rotated randomly to select specific solutions that will participate in the formation of the next generation. In this selection method, as can be seen in Figure 2.4 the best individual has the largest share of the roulette wheel, while the worst individual has the lowest share. This way this mechanism simulates the natural selection of the fittest individual in nature. Since a roulette wheel is a stochastic operator, poor individuals have a small probability of participating in the creation of the next generation. If a poor solution is *lucky*, its genes move to the next generation. Discarding such solutions will reduce the diversity of the population and should be avoided. [11] This method suffers from many problems such as errors introduced by its stochastic nature. [12]

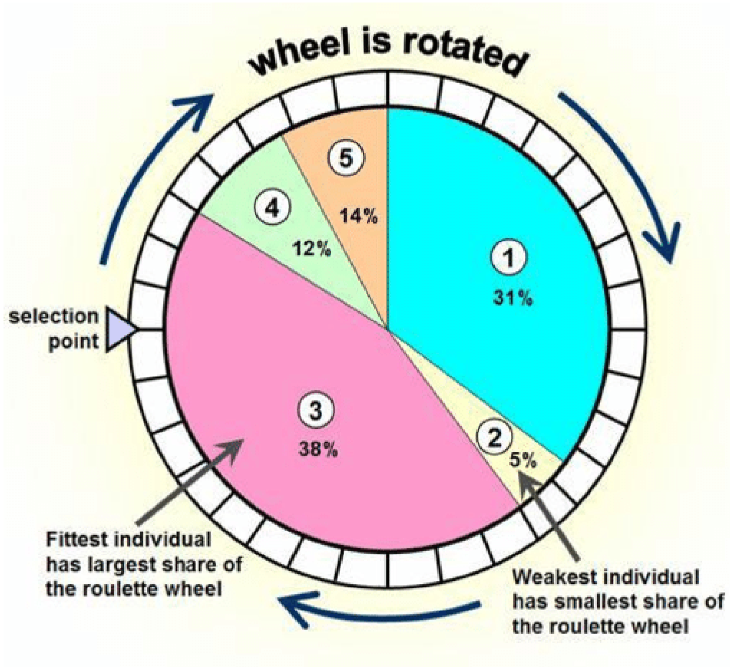


Figure 2.4: Roulette wheel selection method. [14]

De Jong and Brindle modified the roulette wheel selection method to remove errors by introducing the concept of determinism in the selection procedure. Rank selection is the modified form of Roulette wheel selection. It utilizes the ranks instead of the fitness value. So, each individual gets a chance to get selected according to their ranks. The rank selection method reduces the chances of prematurely converging the solution to local minima. [12]

Anther technique is Tournament selection which was first proposed by Brindle in 1983. In this case, the individuals are selected according to their fitness values from a stochastic roulette wheel in pairs. After selection, the individuals with higher fitness values are added to the pool of the next generation. In this method of selection, each individual is compared with all n-1 other individuals if it reaches the final population of solutions. Stochastic Universal Sampling

(SUS) is an extension of the existing roulette wheel selection method. It uses a random starting point in the list of individuals from a generation and selects the new individual at evenly spaced intervals. It gives an equal chance to all the individuals in getting selected for participating in crossover for the next generation. Finally, Boltzmann selection is based on entropy and sampling methods, which are used in Monte Carlo Simulation. It helps in solving the problem of premature convergence. The probability is very high for selecting the best string, while it executes in very less time. However, there is a possibility of information loss. [12]

2.1.3. Crossover operators

After selecting the individuals using a selection operator, they have to be employed to create the new generation. In nature, the chromosomes in the genes of a male and a female are combined to produce a new chromosome. This is simulated by combining two solutions (parent solutions) selected by any of the methods previously explain to produce two new solutions (offspring solutions) in the GA algorithm. [11] The well-known crossover operators are single-point, k-point, uniform, partially matched, order, precedence preserving, shuffle, reduced surrogate and cycle crossover. [12]

In a single-point crossover, a random crossover point is selected. The genetic information of two parents which is beyond that point will be swapped with each other in the new offspring. In a two-point and k-point crossover, two or more random crossover points are selected and the genetic information of parents will be swapped as per the segments that have been created. In a uniform crossover, parents can not be decomposed into segments. The parents can be treated as each gene separately. It is randomly decided whether it is needed to swap the gene with the same location of another chromosome. Figure 2.5 shows the swapping of genetic information with a one-point, two-point and uniform crossover. [12]

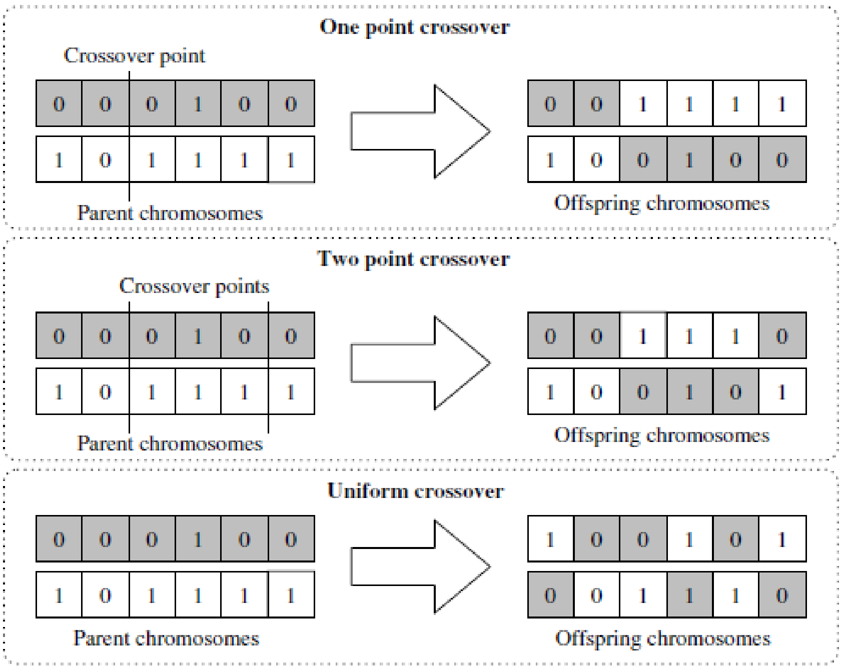


Figure 2.5: One-point, two-point, and uniform crossover methods. [15]

Regarding Partially matched crossover (PMX), it is the most frequently used crossover operator since performs better than most of the other crossover operators. In this case, two parents are chosen for mating. One parent donates some part of genetic material and the corresponding part of the other parent participates in the child. Once this process is completed, the left-out alleles are copied from the second parent. Order crossover (OX) copies one (or more) parts of a parent to the offspring from the selected cut-points and fills the remaining space with values other than the ones included in the copied section. Finally, shuffle crossover was proposed to reduce the bias introduced by other crossover techniques. It shuffles the values of an individual solution before the crossover and unshuffles them after the crossover operation is performed. [12]

2.1.4. Mutation operators

Mutation is the last evolutionary operator, in which one or multiple genes are altered after creating children solutions. The mutation rate is set to low in GA because high mutation rates convert GA to a primitive random search. The mutation operator maintains the diversity of the population by introducing another level of randomness. In fact, this operator prevents solutions from becoming similar and increases the probability of avoiding local solutions in the GA algorithm. [11] The well-known mutation operators are displacement, simple inversion and scramble mutation.

Displacement mutation (DM) operator displaces a substring of a given individual solution within itself. The place is randomly chosen from the given substring for displacement such that the resulting solution is valid as well as a random displacement mutation. The simple inversion mutation operator (SIM) reverses the substring between any two specified locations in an individual solution, see Figure 2.6. SIM is an inversion operator that reverses the randomly selected string and places it at a random location. The scramble mutation (SM) operator places the elements in a specified range of the individual solution in a random order and checks whether the fitness value of the recently generated solution is improved or not. [12]

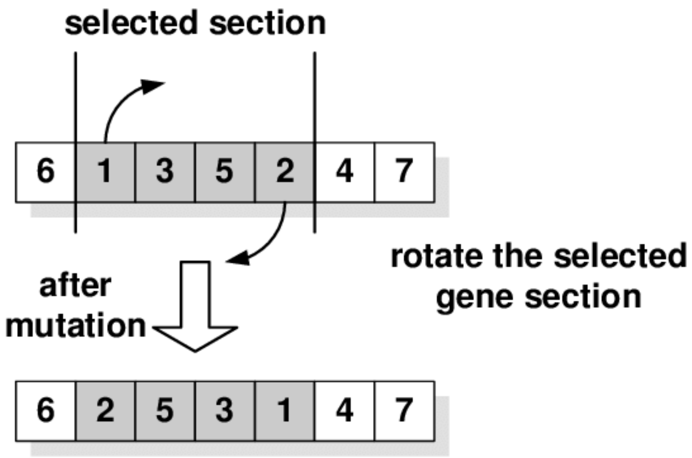


Figure 2.6: Simple Inversion Mutation operator. [16]

Methodology

3.1. Problem definition

The case study problem is based on prior work fielded by the Department of Mechanical and Aerospace Engineering at North Carolina State University. It consists of an MP3 product line optimization problem. The customer preference model for the MP3 market simulator was constructed from 205 choice-based conjoint surveys containing twelve choice tasks each. *Sawtooth CBC/HB* software was used to fit a Hierarchical Bayes (HB) mixed logit model that estimates the observed part-worths for each attribute level for each respondent. This model was fit using the default settings and the part-worths for the price levels were constrained to be monotonically decreasing (i.e. the part-worth for $49 is greater than the part-worth for $99). Each respondent in the HB model has fifty five total part-worths; forty six for the features, eight for the price levels, and one for the outside good (i.e. the not buy or walk away option). Table 3.1 shows the breakdown of the seven product attributes, with their respective possible levels, as well as the price levels used in the survey. [17]

The cost model for the MP3 market simulator (Table 3.2) shows the price that will be charged to the customer associated with each level selected in the final design. This model was constructed by first allocating the difference in the Manufacturer's Suggested Retail Price (MSRP) of different iPods to the various features in Table 3.1 and then dividing those values in half. This assumes that the MSRP of an iPod is 100% of its cost. As an example, consider two similar iPods that vary only in storage size. The incremental cost of increasing storage size is estimated as half of the difference in the MSRP of these two iPods. Note that iPods were used only to build the cost model as they presented a large amount of feature diversity from a single manufacturer. Future work should include brand as an attribute in the market model, allowing cost structures to be defined more generally. [17]

To ensure that the MSRP of all potential products fell within the price levels used in the survey ($49-$699), a base price of $52 was selected. The base price refers to the selling price of a product whose attributes are all configured to level 1. In addition, competition was added to the MP3 market simulation to increase the difficulty of finding optimal product lines. [17]

Essentially, the design variables that can be controlled for this problem are the first seven attributes that define each product's configuration. So, if one product with *Video only*, *App only*, *Touchpad*, *4.5 in. diag*, *64 GB*, *Red* and *Custom graphic* is designed, the design string would be [3 3 2 4 4 4 3], which corresponds to the level of each attribute. The code elaborated in *MATLAB* will calculate the price of the product based on the features that have been defined and will return the market share associated, given as a number between 0 and 100.

	Attributes							
Level	**Photo/ Video/ Camera**	**Web/ App/ Ped**	**Input**	**Screen Size**	**Storage**	**Shell Color**	**Shell Overlay**	**Price**
1	None	None	Dial	1.5 in. diag	2 GB	Black	No pattern/ graphic	$49
2	Photo only	Web only	Touchpad	2.5 in. diag	16 GB	White	Custom pattern	$99
3	Video only	App only	Touchscreen	3.5 in. diag	32 GB	Silver	Custom graphic	$199
4	Photo and Video	Ped only	Buttons	4.5 in. diag	64 GB	Red	Custom pattern and graphic	$299
5	Photo and Lo-res camera	Web and App only	-	5.5 in. diag	160 GB	Orange	-	$399
6	Photo and Hi-res camera	App and Ped only	-	6.5 in. diag	240 GB	Green	-	$499
7	Photo Video and Lo-res camera	Web and Ped only	-	-	500 GB	Blue	-	$599
8	Photo Video and Hi-res camera	Web, App and Ped	-	-	750 GB	Custom	-	$699

Tabla 3.1: Levels per attribute and price levels for the MP3 player problem. [17]

	Attributes						
Level	**Photo/ Video/ Camera**	**Web/ App/ Ped**	**Input**	**Screen Size**	**Storage**	**Shell Color**	**Shell Overlay**
1	$0	$0	$52	$0	$0	$0	$0
2	$5	$20	$57	$25	$45	$10	$5
3	$10	$20	$92	$45	$120	$10	$10
4	$15	$10	$72	$60	$200	$10	$15
5	$17	$40	-	$70	$250	$10	-
6	$30	$30	-	$80	$300	$10	-
7	$32	$30	-	-	$350	$10	-
8	$42	$50	-	-	$400	$20	-

Tabla 3.2: Cost per feature for the MP3 player problem. [17]

3.2. Approach

Our approach requires a rigorous algorithm that can solve even blackbox optimization problems. A blackbox is defined as any process that when provided an input, returns an output, but the inner workings of the process are not analytically available. [18]. Blackboxes can be computationally expensive, which poses a practical challenge to optimization. In addition, the functions underlying these models are not necessarily smooth, continuous, or differentiable. Therefore, derivatives of such blackbox models are difficult to approximate numerically. [4] Genetic Algorithms (GA) are a popular choice for blackbox optimization problems since they do not require derivative information. However, convergence of GA is highly dependent on tuning the hyperparameters, encoding, and crossover techniques for the problem considered. [18] [4]

The specific problem analyzed here is not a blackbox, because it is possible to identify its inner workings, as it was explained in Section 3.1. However, it can be treated as one since the interest of this project is not to examine the objective function to be optimized but to create a code that can be used for any kind of problem, including blacboxes. So, treating the case of study as a blackbox, its input is a product line that contains three products. This means that the design string for the MP3 player contains 21 integer variables, seven for each product. Each variable corresponds to the attribute's level, which ranges from 1 to the maximum level of the specific attribute which can be 4, 6 or 8. The output as it was stated in Section 3.1 is the market share associated with the product offerings created, in the shape of the percentage of total sales. For a better comprehension see Figure 3.1.

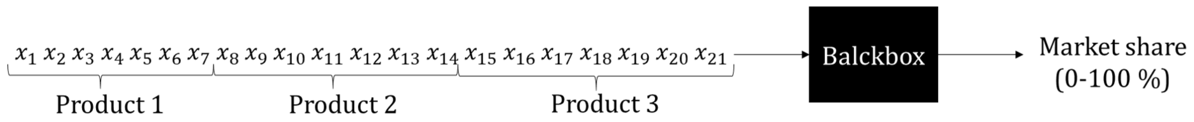


Figure 3.1: Blackbox schema.

Prior to the optimization, several simulations of the objective function were executed for random combinations of design strings. This way, an exploration of the space and the problem was carried out to help to understand in greater depth what the problem faced consists of. Once this exploration was done, the next step is to find the fully optimized product line solution for the MP3 case. To conduct the optimization, a genetic algorithm via the optimization Toolbox of *MATLAB* is used. After that, the optimization problem is modified in order to find the best set of solutions to the product line design problem. Now, the input is the same string but adding at the end the indexes of the attributes of the string that can be changed without modifying excessively the market share. To conduct the optimization, a genetic algorithm is written in *MATLAB*. This time, the output and what is being optimized is the mean of market shares when the variables selected in the input are varied randomly ten times.

It is required to note that multiple runs of the genetic algorithm code must be performed for the case study to account for the stochastic nature of the GA. Finally, add that the results of every step mentioned above are included and explained in more detail in Section 4.

Results and discussion

4.1. Exploration of the space

As described in Section 3.2, multiple simulations of the objective function are executed for random combinations of design strings before starting the whole optimization process. The objective of this study is to explore the space in order to understand in greater depth what the problem faced consists of. More specifically, the objective function was executed 1000 times with random values for the 21 integer variables (three products with seven attributes each). The market share obtained for each simulation can be appreciated in Figure 4.1, where the *x-axis* simply indicates the simulation index from 1 to 1000. As can be seen, the space analyzed does not have any anomalies, the market share can take any value from approximately 5 to almost 70 % just by modifying arbitrarily the levels assigned to each attribute in the design string.

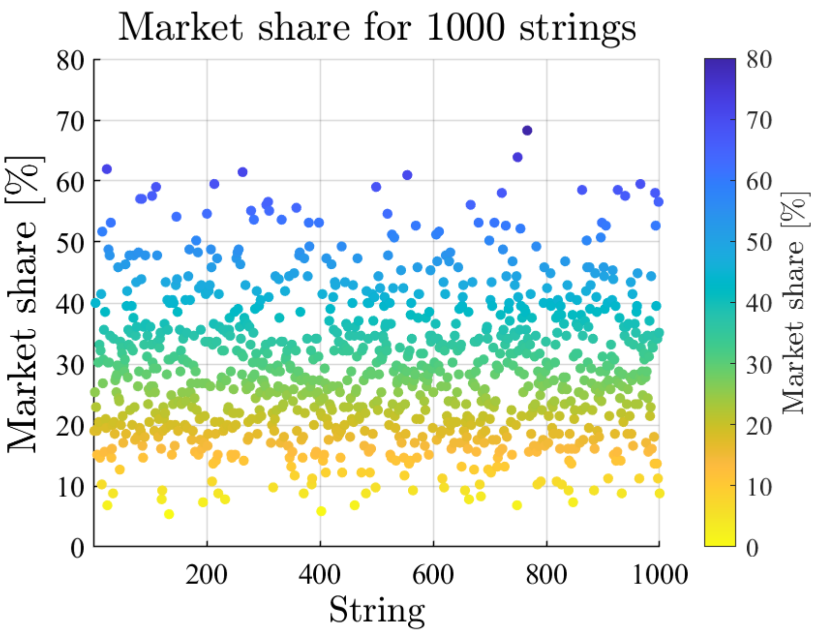


Figure 4.1: Market share for the 1000 random simulations.

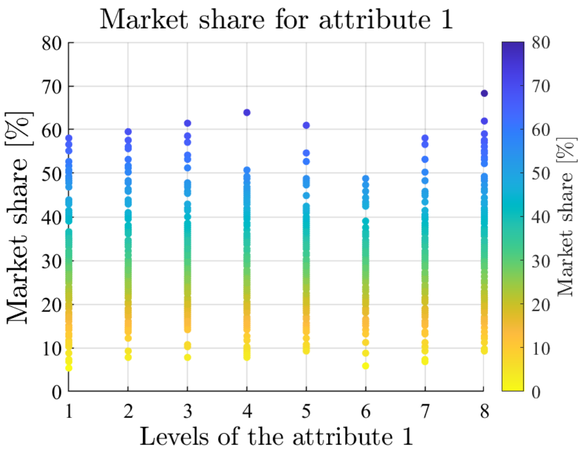
A statistical analysis of the data is done to comprehend better the space. The results of the mean, median, standard deviation, maximum and minimum are collected in Table 4.1. The string that provides the maximum market share is [8 2 2 6 8 1 4| 8 8 3 3 2 3 3| 2 8 2 4 1 3 2].

Mean	29.41 %
Median	28.54 %
Standard deviation	11.18 %
Maximum	68.29 %
Minimum	5.37 %

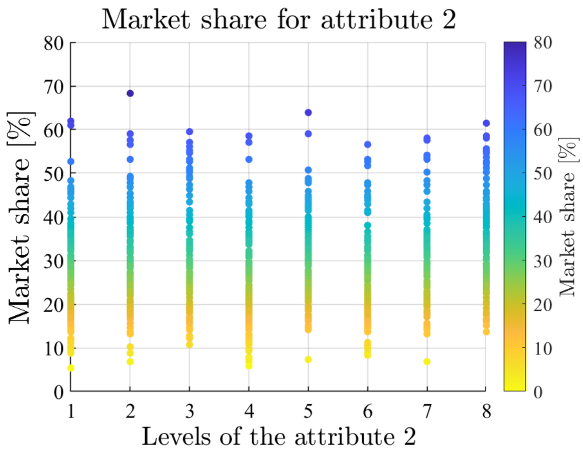
Tabla 4.1: Statistical analysis of the 1000 simulations.

In order to know if there is any kind of level in each one of the attributes that may be better than the rest, the results are classified by levels for every attribute. The attributes from one to four are shown in Figure 4.2 and attributes from five to seven are included in Figure 4.3. Note that not all the attributes have the same number of levels. In fact, attributes one, two, five and six have eight levels; attributes three and seven have four levels; and attribute four has six levels.

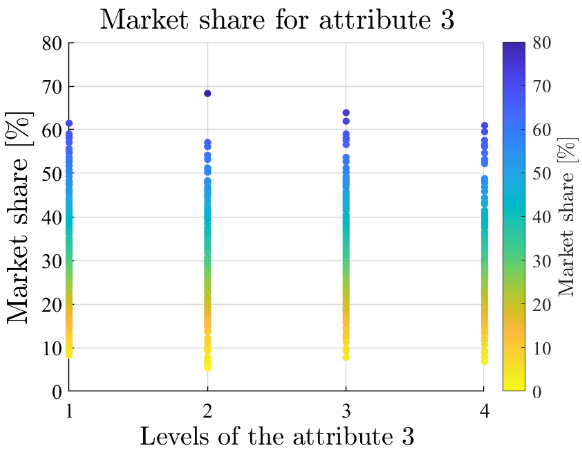
It can be deduced from all the graphs that there is not a superior level in any of the attributes. In other words, any level can lead to a high market share always that is combined with the right levels in the rest of the attributes. Likewise, any level can give a really low market share if it is put together with the wrong levels in the others attributes. For example, it can be seen in Figure 4.2c that for attribute 3 level 2 can lead to a market share as bad as 5.37 % or as good as 68.29 %. The result only depends on the combination with the other attribute levels.



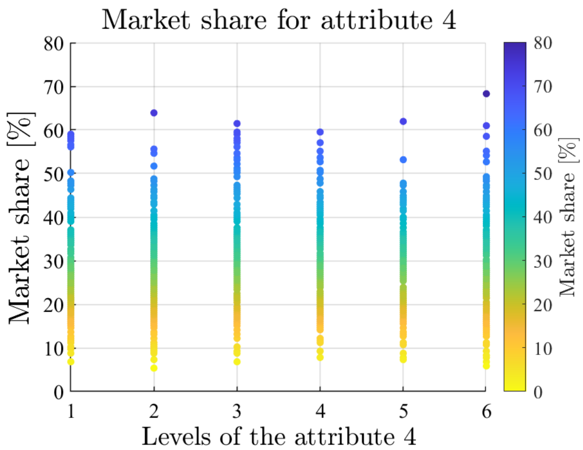
(a) Attribute 1



(b) Attribute 2

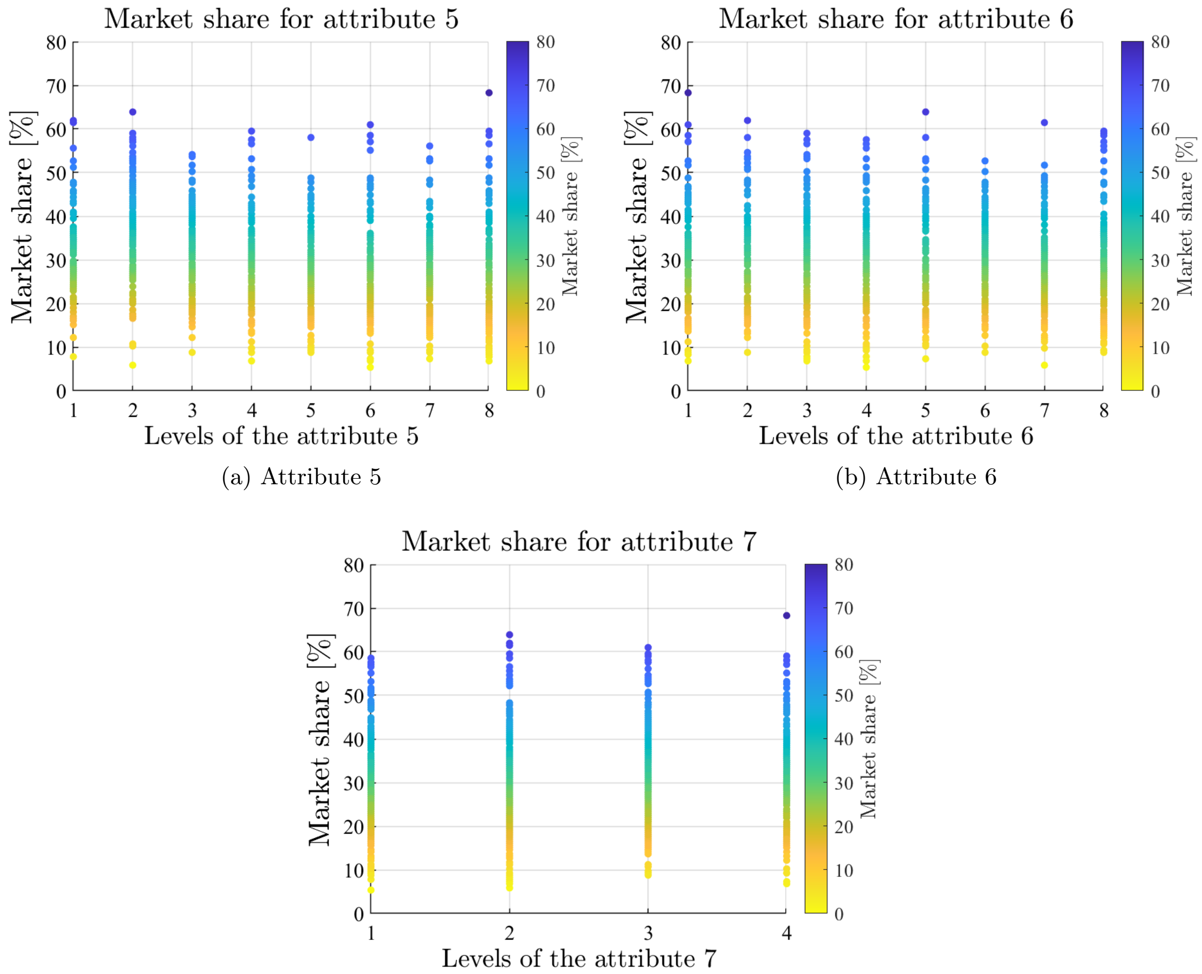


(c) Attribute 3



(d) Attribute 4

Figure 4.2: Market share for attributes 1-4 by levels.



(a) Attribute 5

(b) Attribute 6

(c) Attribute 7

Figure 4.3: Market share for attributes 5-7 by levels.

Now, starting from the best design string of the last study [8 2 2 6 8 1 4| 8 8 3 3 2 3 3| 2 8 2 4 1 3 2], each of the attributes is going to be changed individually leaving the rest with the levels that lead to the maximum market share. Therefore, the process consists of changing the level of each attribute for every product randomly 100 or 50 times; 50 if the attribute changed is the third or the seventh since there are not that many possible combinations. The market share obtained for each simulation can be appreciated in Figure 4.4. As can be seen, the second and fifth attributes have more variability than the rest. This means that these attributes need probably to be fixed because modifying them could reduce the market share considerably. On the contrary, changing the levels of attributes one, six and seven would not affect remarkably the market share. So, these attributes might be more flexible.%

This can be also deduced from Table 4.2 where the mean and standard deviation for each attribute are collected. Effectively, it can be appreciated that the standard deviation for the second and fifth attributes is higher (9.38 % and 9.18 % respectively) while their means are lower (58.72 % and 52.57 % respectively). However, the standard deviation of the first, sixth and seventh attributes is smaller (3.25 %, 2.83 % and 2.04 % respectively) and their means are higher (64.71 %, 64.42 % and 67.64 % respectively).

Besides, Figures 4.5 and 4.6 can also demonstrate this fact. In these graphs, the market share of every simulation is classified by levels for every attribute. Figures 4.5b and 4.6c, corresponding with attributes two and five, present much more dispersion than the rest of the attributes. On the contrary, the first, sixth and seventh attributes (Figures 4.5a, 4.6d and 4.6e) present results not widely scattered.

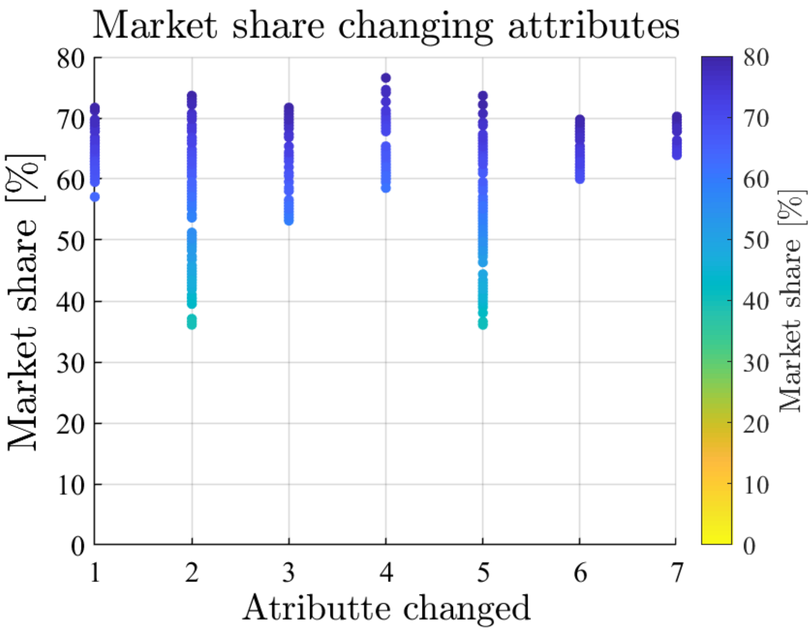
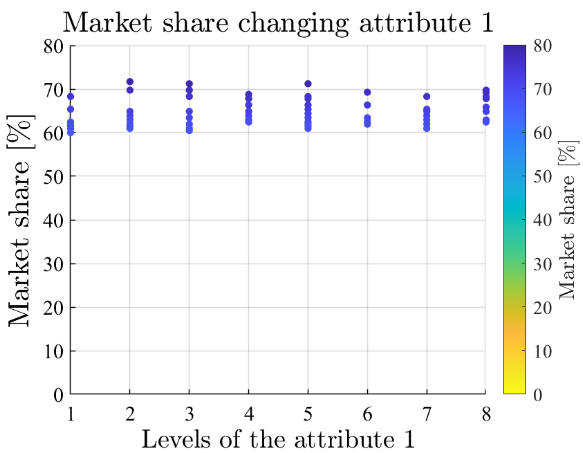


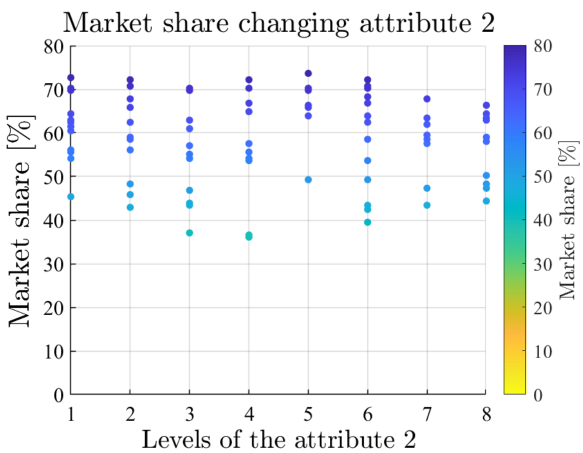
Figure 4.4: Market share depending on the attribute changed.

Atribute	Mean	Standard Deviation
1	64.71 %	3.25 %
2	58.72 %	9.38 %
3	62.52 %	5.55 %
4	66.95 %	5.07 %
5	52.57 %	9.18 %
6	64.42 %	2.83 %
7	67.64 %	2.04 %

Tabla 4.2: Statistical analysis of the seven attributes changing their levels randomly.

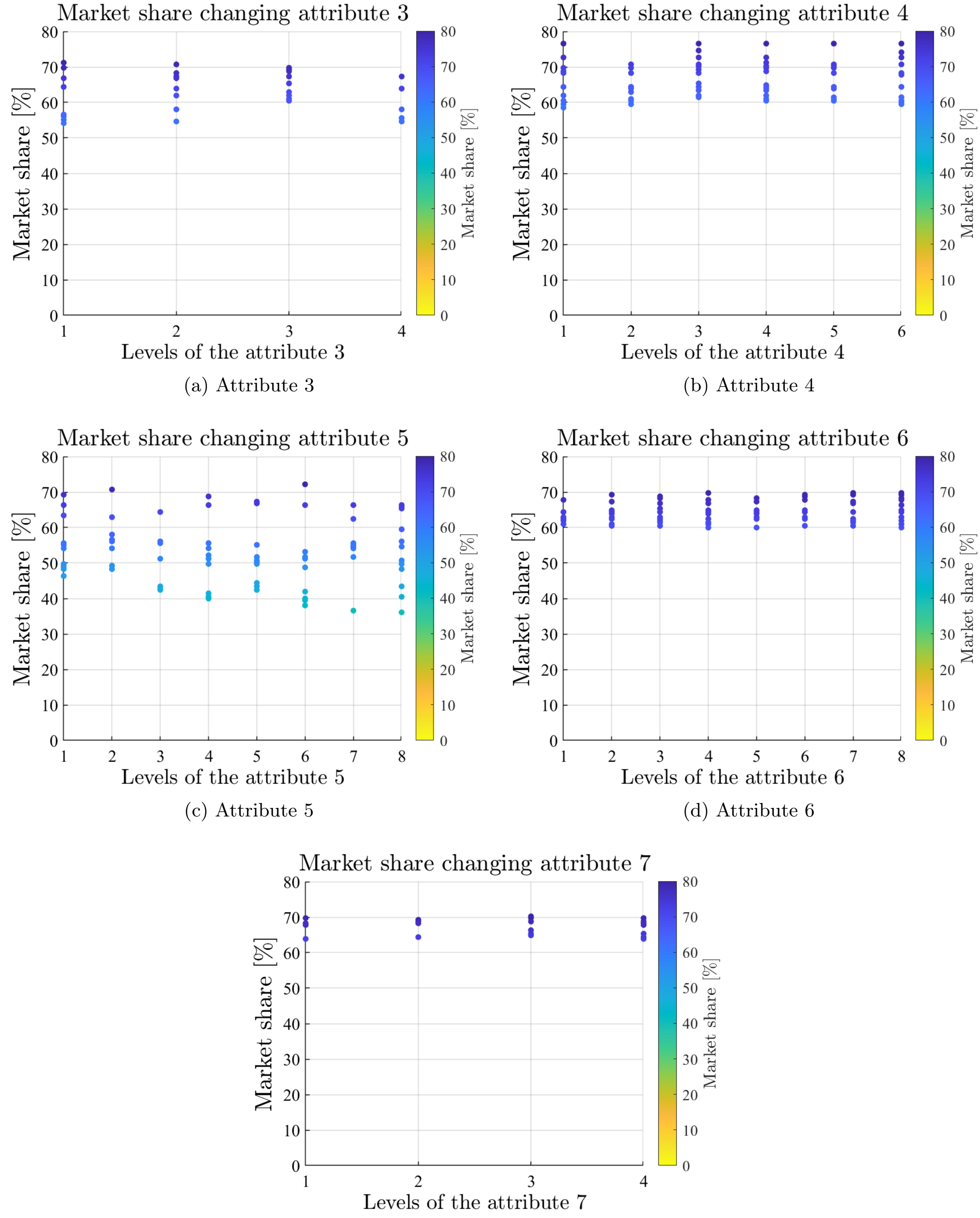


(a) Attribute 1



(b) Attribute 2

Figure 4.5: Market share modifying the levels of attributes 1 and 2.



(a) Attribute 3

(b) Attribute 4

(c) Attribute 5

(d) Attribute 6

(e) Attribute 7

Figure 4.6: Market share modifying the levels of attributes 3-7.

To sum up, after exploring the space, it can be said that there is not any level that is always better than the rest; it always depends on the combination with the others attribute levels. However, it was deduced that changes in attributes two and five can affect more the market share than, for example, modifying attribute seven. As a consequence, it can be expected that in Section 4.3 some of the attributes to be fixed are two and five, while the seventh attribute has more flexibility.

4.2. Fully optimized solution

Once the exploration was concluded, the next step is to find the fully optimized product line solution for the MP3 case. To conduct the optimization, a genetic algorithm via the optimization Toolbox of MATLAB is employed. Particularly, the GA is trying to minimize the negative value of the market share. On the other hand, in order to consider the stochastic nature of the GA three runs of the genetic algorithm code are performed. The results are gathered in Table 4.3.

For a better post-process of the solution, it is important to take into account that the three products can be disorganized and appear in different orders. For example, the design string with products in order A B C would give the same market share as the design string with products with the sequence B C A or C A B. For this reason, all the design strings are rearranged in the way that the first product is the one with the lowest price and the last one is the most expensive of the three products.

Run	**Product 1**	**Product 2**	**Product 3**	**Market share**
1	8 5 2 5 3 6 2	8 3 1 1 2 3 3	4 5 3 3 2 3 3	87.32 %
2	8 5 3 3 2 3 3	8 3 1 1 2 3 3	4 5 3 4 2 4 4	90.24 %
3	8 5 3 4 2 1 3	8 3 2 1 2 3 3	7 8 2 4 2 6 2	88.29 %

Tabla 4.3: Fully optimized solutions.

From Table 4.3, some curious similarities among the solutions can be extracted. Runs 2 and 3 share five of the seven attributes for the first product and six of the seven attributes for the second one (8 5 3 * 2 * 3 | 8 3 * 1 2 3 3). Runs 1 and 2 only share two of the seven attributes in the first product but they have the exact same second product. Regarding product 3, runs 1 and 2 share four of the seven attributes (4 5 3 * 2 * *). From this, it can also be deduced that some of the attributes need to have a specific level in order to achieve a high market share. For example, attribute five is almost always set as level two for every product of every run. Besides, levels five or three are also popular for attribute two. Also, it seems that attribute one must be level eight. In Section 4.4, these statements will be validated or discarded after comparing the fully optimized solution with the best set of solutions obtained in Section 4.3.

Among the three solutions, the one that provides a higher market share is the second one reaching a value of 90 %. This would be the fully optimized solution and it corresponds with the following configuration for the three products:

- Product 1: MP3 with photo, video and a high-resolution camera, web and app, a touchscreen of 3.5 in., 16GB of storage and a silver shell with a custom graphic.
- Product 2: MP3 with photo, video and a high-resolution camera, app, dial, a screen of 1.5 in., 16GB of storage and a silver shell with a custom graphic.
- Product 3: MP3 with photo and video, web and app, a touchscreen of 4.5 in., 16GB of storage and a red shell with a custom pattern and graphic.

4.3. Best set of solutions

The next step is to find the best set of solutions to the product line design problem. In order to achieve it, the optimization problem is modified to find the indexes of the attributes of the string that can be changed without modifying excessively the market share. To conduct the optimization, a genetic algorithm is written in *MATLAB*. This time, what is being minimized is the negative mean of market shares when the variables selected in the input are modified. Again, three runs of the genetic algorithm code must be performed every time to account for the stochastic nature of the GA.

Simulations keeping variable from 1 to 11 attribute levels in the design string are carried out. Increasing in more than 11 the number of parameters flexible does not make sense because the market share would be reduced and the objective of this project is changing some variables of the string while maintaining a high market share. The results of the simulations can be observed in Figure 4.7 where the *y-axis* represents the number of elements fixed from the 21 integer variables that constitute the design string. As can be appreciated, if the number of fixed attributes decreases, the market share is also reduced. However, the curve is quite steep so, with only ten elements defined in the design string it is possible to still get a market share of around 70 %. So, if the product designers want flexibility they can get it by sacrificing slightly the market share. Having only half of the design string defined would imply a reduction of the market share of 20 % with respect to the maximum obtained in Section 4.2. If a higher market share is desired, we could give freedom to only six of the 21 integer variables. This way, a market share greater than 80% can be reached.

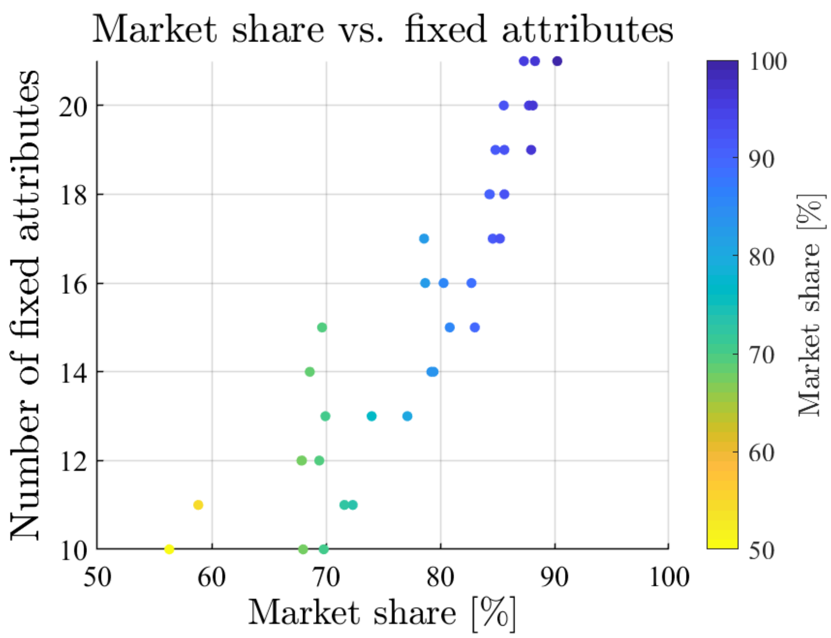


Figure 4.7: Market share depending on the number of fixed attributes.

Taking only the best result of the three runs for every number of fixed attributes allows obtaining the curve shown in Figure 4.8. This curve as was stated before is steep, so the reduction in the market share is not that abrupt as the number of flexible variables increases.

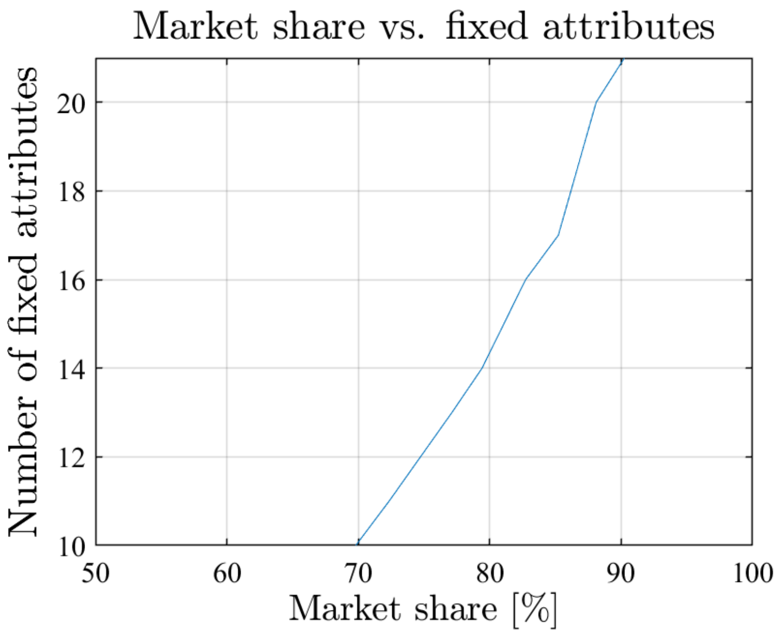


Figure 4.8: Market share depending on the number of fixed attributes for the optimal run.

Once it was deduced that some variables of the design string can be flexible without losing a lot of market share, it is required to acknowledge which of the attributes are the ones that allow more flexibility and which ones need to fix to a specific level in order to not drop the market share. To determine them, the market share of the simulations for 18, 15 and 11 fixed variables are collected in Table 4.4. Not all the simulations have been included because then the data would be too extensive to process properly. Analogously as in the previous section, for a better post-process of the solution, all the design strings in the table are reorganized and their products are ordered from the cheapest to the most expensive of the three products. In addition, level 0 in the design string indicates the variables of the 21 integer variables that are being modified. In other words, the attributes that are allowed to be flexible without deteriorating notably the market share. Finally, it is needed to be said that the market share is the mean of market shares changing ten times the attributes with level 0 randomly.

Fixed variables	**Run**	**Product 1**	**Product 2**	**Product 3**	**Market share**
18	1	2 5 1 1 0 0 0	1 7 3 6 2 3 3	8 5 3 4 2 3 3	85.61 %
	2	4 5 3 3 2 1 3	8 5 1 0 2 3 3	1 0 1 3 2 0 2	84.21 %
	3	2 5 0 1 2 3 0	8 5 3 4 2 3 3	6 3 0 1 2 5 3	84.34 %
15	1	2 5 0 1 2 6 2	0 3 0 0 2 0 3	8 5 3 4 2 0 3	83.02 %
	2	8 5 0 3 2 3 0	0 5 3 4 2 0 3	2 3 1 0 2 3 0	80.83 %
	3	6 5 0 1 2 6 0	4 0 3 4 2 3 0	8 5 3 3 2 0 0	69.66 %
11	1	0 0 2 0 6 6 0	3 7 4 0 0 0 4	0 5 1 0 2 1 0	58.83 %
	2	0 5 1 0 2 0 2	0 0 3 4 0 0 0	6 5 3 0 3 0 4	72.34 %
	3	0 8 0 3 2 0 2	0 5 1 0 2 0 0	5 5 1 0 2 0 0	71.61 %

Tabla 4.4: Set of optimized solutions.

Since it can be difficult to extract any conclusion from the last table, a graphic representing the frequency of each level for every attribute is presented in Figure 4.9. From the figure, it can be deduced that attribute 1 is pretty flexible with a little predominance for level eight. Attribute two is mostly set in level five though level three is the second most important. Attribute three has predominance for levels one and three. Regarding attribute four, it is more flexible and can be assigned to levels one, three or four. Attribute five is in its majority set to level two. Finally, attributes six and seven have a lot of freedom being dominant the level three in both of them. However, it is not recommended that attribute six has a level two, seven or eight nor that attribute seven has a level one.

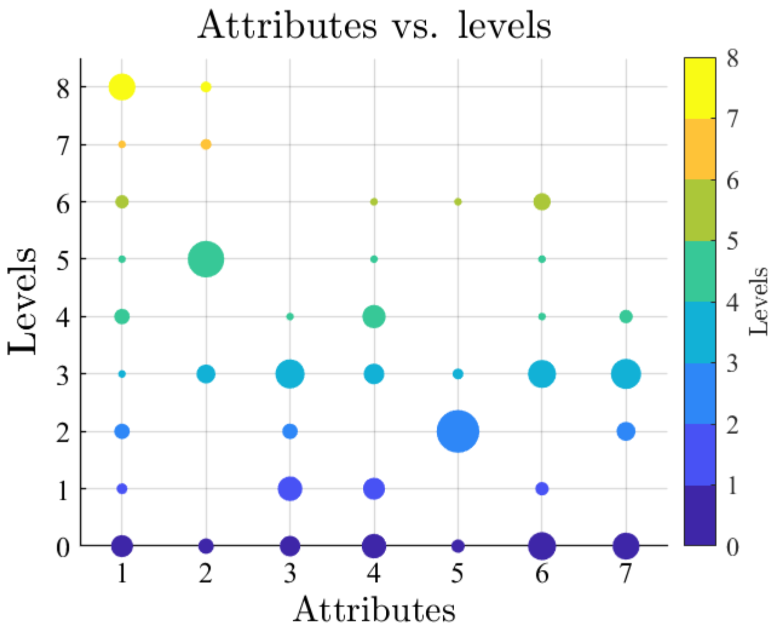


Figure 4.9: Frequency of the levels for each attribute.

Furthermore, for a better understanding of the results another table is elaborated but taking only the best results of the three runs seen in Table 4.4 and the best run of Table 4.3. This way, in Table 4.5 the optimal design strings fixing 21, 18, 15 and 11 variables are compiled. Again the level 0 in the design string indicates the variables that can be modified without deteriorating the market share. Besides, it is important to note that for a better analysis of the results this time the three products are put vertically for every number or fixed attributes instead of being placed horizontally as in prior tables.

Fixed variables	Product	Design string
21	1	8 5 3 3 2 3 3
	2	8 3 1 1 2 3 3
	3	4 5 3 4 2 4 4
18	1	2 5 1 1 0 0 0
	2	1 7 3 6 2 3 3
	3	8 5 3 4 2 3 3
15	1	2 5 0 1 2 6 2
	2	0 3 0 0 2 0 3
	3	8 5 3 4 2 0 3
11	1	0 5 1 0 2 0 2
	2	0 0 3 4 0 0 0
	3	6 5 3 0 3 0 4

Tabla 4.5: Set of optimized solutions for the optimal run.

Analyzing the Table 4.5 can be noticed a couple of patterns. For example, levels eight, four and three are pretty popular for the three first attributes, respectively. Also, most of the time when attribute three is set to level three, attribute four is set to level four. It is also frequent to have levels three for the sixth and seventh attributes when the design string starts with level eight. In addition, it can be corroborated the dominance of level five for attribute two; levels one and three for attribute three; levels four and one for attribute four; level two for attribute 5 and levels three for attribute six and seven.

Taking everything explained above into account (patterns, dominance, variability) the schema that a good solution should have is presented in Table 4.6. This would be delivered to the product designers as a starting point for the debate by the product configuration team. It can be seen how the first product is completely defined for the patterns observed from 4.5. Then, the next two products have more flexibility. The first attribute can take any value because is not a determining factor in the market share. On the contrary, the second attribute is fixed since it is really significant for the market share. As was seen, most of the design strings have two fives and one three for this attribute. The third attribute can take levels one or three and attribute four can take levels one, three or four. Regarding attribute five, it must be clearly fixed to level two. That is due to the fact that the level of this attribute is key for the price of the final product, being level two one of the cheapest options. Attributes six and seven are pretty flexible since the cost of any of their levels is really low comparing it with other attribute levels. Nevertheless, there are some levels more beneficial than others. Particularly, for attribute six, it is not recommended level two, seven or eight and, for attribute seven, level one is not suggested.

Attributes	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Product 1	8	5	3	4	2	3	3
Product 2	*	3	1/3	1/3/4	2	1/3/4/5/6	2/3/4
Product 3	*	5	1/3	1/3/4	2	1/3/4/5/6	2/3/4

Tabla 4.6: Schema that characterizes a good solution.

The configuration for the three products in the schema that characterizes the qualities of a good solution is:

- Product 1: MP3 with photo, video and a high-resolution camera, web and app, a touchscreen of 4.5 in., 16GB of storage and a silver shell with custom graphic.
- Product 2: MP3 with any kind of photo, video or camera, app, dial or touchscreen of 1.5, 3.5 or 4.5 in., 16GB of storage and a shell of any color but white, blue or custom with a custom pattern or graphic or both.
- Product 3: MP3 with any kind of photo, video or camera, web and app, dial or touchscreen of 1.5, 3.5 or 4.5 in., 16GB of storage and a shell of any color but white, blue or custom with a custom pattern or graphic or both.

4.4. Comparison of both solutions

In the last section of the results, the fully optimized solution calculated in Section 4.2 is going to be compared with the schema that characterizes a good solution obtained in Section 4.3. To facilitate the contrast, both results are presented in Table 4.7.

Attributes	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Product 1 fully	8	5	3	3	2	3	3
Product 1 schema	8	5	3	4	2	3	3
Product 2 fully	8	3	1	1	2	3	3
Product 2 schema	*	3	1/3	1/3/4	2	1/3/4/5/6	2/3/4
Product 3 fully	4	5	3	4	2	4	4
Product 3 schema	*	5	1/3	1/3/4	2	1/3/4/5/6	2/3/4

Tabla 4.7: Summary of the results from prior sections.

It can be seen how the first product is almost the same for both solutions, the only difference would be the level of attribute four (screen size) and the difference is not substantial, only one level of difference. Regarding the second and third products, it can be seen how the most fundamental attributes for the market share, two and five, are the same for both cases. The rest of the attributes have more flexibility in the case of the schema but it can be appreciated how the levels of the fully optimized solutions are some of the options suggested for the schema that characterizes a good solution.

Conclusions

5.1. Conclusions

This project allows by using a genetic algorithm in *MATLAB* to characterize the qualities of a good solution for a market-driven design problem. It explores how the optimization problem can be reformulated so that a set of optimal solutions is returned, rather than a single optimal point. This schema could then be presented to the product configuration team, instead of a fully optimized result that will be changed anyways.

With the intention of accomplishing our objectives, an exploration of the space is done. The conclusions extracted from it are that the space analyzed does not have any anomalies and there is not any level that is always better than the rest. It always depends on the combination with the others attribute levels. However, it was deduced that changes in attributes two and five can affect more the market share than, for example, modifying attributes one, six or seven.

Once the space has been explored, the fully optimized solution is calculated. This solution allows us to get a market share of 90 % and it consists of an MP3 with photo, video and a high-resolution camera, web and app, a touchscreen of 3.5 in., 16GB of storage and a silver shell with a custom graphic for product one; an MP3 with photo, video and a high-resolution camera, app, dial, a screen of 1.5 in., 16GB of storage and a silver shell with a custom graphic for product two; and an MP3 with photo and video, web and app, a touchscreen of 4.5 in., 16GB of storage and a red shell with a custom pattern and graphic for product three.

On the other hand, a general schema that identifies the characteristics of a good solution is obtained. In order to acquire this, several simulations are carried out. From them, it can be realized that as the number of fixed attributes decreases, the market share that can be achieved is lower. So, a compromise need to be found. By all the results fixing a different number of variables, it is also deduced that the more expensive attributes like the storage (attribute five) need to be fixed because it affects notably the market share. In addition, attribute two is also key for the market share so it also needs to be fixed to level five or three. On the contrary, cheaper attributes such as shell color or overlay (attributes six and seven) can be more flexible since they are not fundamental to the market share. Besides, attribute number one is really flexible and it can be at any possible level. The configuration for the three products in the schema that characterizes the qualities of a good solution is an MP3 with photo, video and a high-resolution camera, web and app, a touchscreen of 4.5 in., 16GB of storage and a silver shell with a custom graphic for product one; an MP3 with any kind of photo, video or camera, app, dial or touchscreen of 1.5, 3.5 or 4.5 in., 16GB of storage and a shell of any color but white, blue or custom with a custom pattern or graphic or both for product two and the same for product three but with web and app instead of only app.

Finally, the fully optimized solution is compared with the schema that characterizes a good solution. It can be seen how they share almost the same first product and how for the two last products the levels of the fully optimized solutions are some of the options suggested for the schema that characterizes a good solution.

Bibliography

[1] Kim B Clark, W Bruce Chew, Takahiro Fujimoto, John Meyer, and FM Scherer. Product development in the world auto industry. *Brookings Papers on economic activity*, 1987(3):729–781, 1987.

[2] Durward K Sobek and Allen C Ward. Principles from toyota's set-based concurrent engineering process. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 97607, page V004T04A035. American Society of Mechanical Engineers, 1996.

[3] Boris Toche, Robert Pellerin, and Clement Fortin. Set-based design: a review and new directions. *Design Science*, 6:e18, 2020.

[4] Khalil Alhandawi. Optimization-driven set-based design for dynamic design requirements. 2021.

[5] Glenn Ballard. Positive vs negative iteration in design. In *Proceedings eighth Annual Conference of the International Group for lean construction, IGLC-6, Brighton, UK*, pages 17–19, 2000.

[6] Dunwei Gong, Jing Sun, and Zhuang Miao. A set-based genetic algorithm for interval many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, 22(1):47–60, 2016.

[7] Shari Hannapel and Nickolas Vlahopoulos. Implementation of set-based design in multidisciplinary design optimization. *Structural and multidisciplinary optimization*, 50:101–112, 2014.

[8] Richard J Malak Jr, Jason Matthew Aughenbaugh, and Christiaan JJ Paredis. Multiattribute utility analysis in set-based conceptual design. *Computer-Aided Design*, 41(3):214–227, 2009.

[9] Yoon-Eui Nahm and Haruo Ishikawa. Representing and aggregating engineering quantities with preference structure for set-based concurrent engineering. *Concurrent Engineering*, 13(2):123–133, 2005.

[10] Jiachuan Wang and Janis Terpenny. Interactive evolutionary solution synthesis in fuzzy set-based preliminary engineering design. *Journal of intelligent manufacturing*, 14:153–167, 2003.

[11] Seyedali Mirjalili and Seyedali Mirjalili. Genetic algorithm. *Evolutionary Algorithms and Neural Networks: Theory and Applications*, pages 43–55, 2019.

[12] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80:8091–8126, 2021.

[13] Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4:65–85, 1994.

[14] IM El-Desoky, MA El-Shorbagy, SM Nasr, ZM Hendawy, and AA Mousa. A hybrid genetic algorithm for job shop scheduling problems. *Int. J. Adv. Eng. Technol. Comput. Sci*, 3(1):6–17, 2016.

[15] Lubna Zaghlul Bashir. Find optimal solution for eggcrate function based on objective function. *World Scientific News*, (10):53–72, 2015.

[16] Ming-Shen Jian, Ta Yuan Chou, Kun-Sian Sie, and Long-Yeu Chung. Adaptive life-cycle and viability based paramecium-imitated evolutionary algorithm. *WSEAS Trans. on Computers*, 8(8):1358–1367, 2009.

[17] Garrett Foster and Scott Ferguson. Enhanced targeted initial populations for multiobjective product line optimization. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 55881, page V03AT03A032. American Society of Mechanical Engineers, 2013.

[18] Charles Audet and Warren Hare. Derivative-free and blackbox optimization. 2017.

[19] Royal Decree 486/1997, of April 14th 1997, which establishes the minimum safety and health provisions in the workplaces. . *Official State Bulletin.* Ministry of Labor and Social Affairs, Government of Spain, Madrid, April 23rd 1997, No. 97.

[20] Royal Decree 488/1997, of April 14th 1997, on minimum safety and health provisions for work with equipment that includes display screens. *Official State Bulletin.* Ministry of Labor and Social Affairs, Government of Spain, Madrid, April 23rd 1997, No. 97.

Part II

BIDDING SPECIFICATIONS

Bidding specifications

The development of this Master's thesis has been carried out at the Department of Mechanical and Aerospace Engineering of the North Carolina State University (NCSU) in association with the Higher Technical School of Design Engineering of the Polytechnic University of Valencia (UPV). More precisely, the thesis' author is a student at UPV doing a year of exchange at NCSU. However, due to the fact that the student did not need any of the university facilities for carrying out the study, all the work was accomplished at the student's house. For obvious reasons, the working conditions present in that space do not correspond to those that should exist in the buildings belonging to any of the universities.

The minimum health and safety conditions that must be met in these workplaces are indicated in Royal Decree 486/1997, dated April 14th 1997, in accordance with Article 6 of Law 31/1995, dated November 8th 1995, on the Prevention of Occupational Risks. [19] To these minimum provisions must be added those relating to work with equipment that includes display screens, as is the case. These are included in Royal Decree 488/1997, of April 14th 1997, also in accordance with Article 6 of Law 31/1995, of November 8th 1995, on Occupational Risk Prevention. [20]

In the following Sections 1.1 y 1.2, the conditions concerning the working environment and the computer resources are explained, respectively. Unless otherwise specified, all the annexes, articles and items mentioned belong to the Royal Decrees specified above. It should be added that only those relevant to the type of work performed are included.

1.1. Working environment conditions

Firstly, the minimum conditions to be applied in a general work environment are briefly explained. According to Appendix I of Royal Decree 486/1997, the buildings where work is carried out must be structurally safe and their dimensions must allow workers to perform their tasks without risk to their health and safety, in acceptable ergonomic conditions. Floors must be fixed, stable and not slippery, without irregularities or dangerous slopes. In addition, windows, ventilation and lighting devices, doors, ramps and fixed and service stairs must be adequate and safe, complying with the specifications indicated for each of these elements. With regard to circulation routes, both those located outside and inside must be easy to use and safe for pedestrians or vehicles circulating along them and for personnel working in their vicinity. Likewise, evacuation routes and exits must guarantee maximum safety conditions, allowing for the rapid evacuation of workers. On the other hand, workplaces must comply with the regulations on fire protection and electrical installations. Finally, it should be added that workers with reduced mobility must be able to use all the building's facilities, which must be adapted accordingly.

Appendix II of the same document contains the regulations regarding the conditions of order, cleanliness and maintenance of the workplace. These indicate that this space must be cleaned periodically in order to maintain adequate hygienic conditions and that passageways must be free of obstacles so they can be used without difficulty. In addition, the facilities must be regularly maintained so their operating conditions always meet the project specifications, and any deficiencies that may affect the safety and health of workers must be promptly remedied.

The next appendix, Appendix III, establishes the environmental conditions of workplaces. Exposure to these conditions must not pose a risk to the safety and health of workers and, as far as possible, must not constitute a source of discomfort or nuisance. The temperature should therefore be between 17 and 27 °C (62.6 and 80.6 °F) and the relative humidity between 30 and 70 %.

Appendix IV stipulates the lighting of workplaces. Whenever possible, lighting should be natural, supplemented by artificial lighting when the former alone does not guarantee adequate visibility conditions. In such cases, general artificial lighting should be used in preference, supplemented by localized lighting when, in specific areas, high levels of illumination are required. On the other hand, the distribution of lighting levels must be as uniform as possible, trying to maintain adequate levels and contrasts of luminance, avoiding abrupt variations and glare.

Appendix V governs sanitary facilities. In this regard, workplaces must have drinking water in sufficient quantity and be easily accessible. They must also have toilets with washbasins in the vicinity.

Finally, Appendix VI determines the first aid material necessary in the event of an accident, which must be adequate, in terms of quantity and characteristics, to the number of workers, the risks to which they are exposed and the facilities for accessing the nearest medical assistance center.

Attention is now focused on Royal Decree 488/1997, which regulates the minimum health and safety requirements for working with display screen equipment. Mention of these conditions is essential, as display screens have played a fundamental role in the development of this project. Therefore, the workplace must be adapted in order to avoid risks for the user in front of the computer, in particular the possible risks for eyesight and physical and mental workload problems, as well as the possible added or combined effects of these. Some of the recommendations to be adopted to avoid these problems that damage the employee's health are listed below.

As for the equipment itself, its use must not be a source of risk for workers. To this end, the screen must be adjustable and tiltable at will, with well-defined and stable characters and images. Likewise, its brightness must be easily adjustable and there must be no reflections or flashes that could disturb the user. The keyboard should be tiltable and independent of the screen, facilitating user comfort and avoiding arm and hand fatigue. In addition, there must be enough space to place the arms in front of it. The keys must be legible and sufficiently prominent, with a layout that tends to facilitate their use. The table should be low reflective with sufficient dimensions to allow the placement of equipment and material, as well as a comfortable position for the worker. The seat should enable the user to move freely and provide a comfortable posture. Its height should be adjustable and the backrest reclining.

With regard to the environment, the workstation must be large enough to provide changes in position. General lighting and special lighting should ensure adequate levels of illumination and appropriate luminance ratios between the display and its surroundings. In addition, glare and disturbing reflections on the screen or other parts of the equipment must be avoided, therefore, windows must be equipped with devices that allow the incoming light to be regulated. The noise produced by the equipment installed in the workstation should be taken into account when designing the workstation, especially to not disturb the worker's attention. The equipment installed in the workplace must not produce additional heat that could cause discomfort to workers. In addition, electromagnetic radiation outside the visible spectrum must be reduced to insignificant levels to ensure the safety and health protection of workers.

Finally, as far as computer-to-person interfacing is concerned, the program must be suitable for the target task and easy to use, being adapted to the user's level of knowledge and experience.

1.2. IT resources conditions

In order to carry out the calculations for this project, it has been necessary to use modern computing resources. These can be divided into hardware and software.

Regarding the hardware, the physical equipment necessary to carry out the project must have considerable computing power and memory given the demands of the simulations performed. Therefore, the hardware used has been the student's laptop computer; for the design, simulation and data analysis, computational calculations and writing tasks. The characteristics of this equipment are shown in Table 1.1.

With regard to the software required to develop the project, the software used is shown in Table 1.2, including the purpose or function of each one.

Technical specifications	
Model	HP 250 G5 Notebook PC
Processor	Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz (4 CPUs)
RAM	8192MB
Storage	238GB + Samsung Portable SSD T5 SATA 500GB
Graphic Card	Intel(R) HD Graphics 520
Operating system	Windows 10 Pro 64 bits

Tabla 1.1: Technical specifications of the student's laptop.

Software	**Function**
MATLAB R2020a	Simulations, data post-processing and graphical plotting
Wolfram Mathematica 10.4	Various calculations
Microsoft Office 2019	Data processing and preparation of presentations
Overleaf LaTeX	Writing and text editing

Tabla 1.2: Software required for the overall development of the project.

Part III

BUDGET

Budget

The purpose of this section is to collect and break down an estimate of the cost involved in the development of this thesis, which makes up the budget of the project. This budget considers the human cost required, as well as the cost of the use of computer licenses and the acquisition of the hardware.

The monetary unit used is considered to be the dollar [$], in accordance with the United States of America, and the time unit is the hour [h]. In addition, a 5.25% corresponding to the North Carolina individual income tax rate is added to the final value.

1.1. Labor cost

The labor cost includes both the student who completes his or her graduate studies and is carrying out the project, as well as the professor who is tutoring him or her. It is considered that the professor receives a salary commensurate with his work, while the student receives an internship salary. The time dedicated by the tutor includes hours of meetings, consulting of doubts, review of the work and management of the project's formalities. On the other hand, the hours spent by the student correspond to the 450 hours that the MAE 586 Spring 2023 Project Work In Mechanical Engineering must include, where the 13.5 ECTS (405 h) of the Master's Degree Final Project are already included.

The above costs are detailed in the table 1.1. It provides a breakdown of the hours worked and the unit cost per hour for the author and the tutor, including the total human cost of the project.

Concept	**Time [h]**	**Unit cost [$/h]**	**Cost [$]**
Author	450	10.00	4500
Tutor	70	28.50	1995
Total			**6495**

Tabla 1.1: Detail of the human cost of the project.

1.2. Computational cost

The computational cost includes both the acquisition cost of the equipment required for the project (hardware) and the cost of software licenses used during the development of the work (software).

The equipment used to perform the calculations includes the student's laptop computer, whose specifications are included in the section 1.2 of the bidding specifications. The market value of such equipment is shown in Table 1.2.

The software used includes Wolfram Mathematica, MATLAB, Microsoft Office and Overleaf LaTeX; also mentioned above in the section 1.2 of the bidding specifications. All of them can be acquired and used free of charge thanks to the student license provided by the NCSU and/or UPV or because they are freely available, as is the case of Overleaf. The total cost of acquiring the licenses of these programs during the period of development of the work is shown in Table 1.2.

Concept	Cost [$]
MATLAB	0
Wolfram Mathematica	0
Microsoft Office	0
Overleaf LaTeX	0
AC Laptop	616
Total	**616**

Tabla 1.2: Detail of the computational cost of the project.

1.3. Total cost of the project

Adding the total labor and computational costs and the corresponding 5.25% of tax, the total estimated value of the project amounts to:

SEVEN THOUSAND FOUR HUNDRED EIGHTY FOUR DOLLARS AND THIRTY THREE CENTS

(7484.33 $)

For a more detailed breakdown, please refer to Table 1.3.

Concept	Cost [$]
Labor Cost	6495
Computational Cost	616
Total gross	7111
Tax (5.25%)	373.33
Total	**7484.33**

Tabla 1.3: Detail of total project cost.