Modelling and Multicriteria Optimization of the Web Formation in a Spunbond Processes

Héctor Raymundo Flores Cantú

Vom Fachbereich Mathematik
der Technischen Universität Kaiserslautern
zur Verleihung des Akademischen Grades
„Doktor der Naturwissenschaften“
(Doctor rerum naturalium „Dr. rer. nat.“)
genehmigte Dissertation.

1. Gutachter: PD Dr. habil. Karl-Heinz Küfer
2. Gutachter: Prof. Dr. Andreas Meister

Vollzug der Promotion: 12. Juni 2008

TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

D 386

...por una niña de ojos tristes.

Acknowledgments

I want to express my gratitude to Dr. Karl-Heinz Küfer for giving me the opportunity to work in this interesting project, to the CONACYT (Consejo Nacional de Ciencia y Tecnologia) for the economical support in the form of a scholarship and to the Fraunhofer ITWM for offering me a wonderful workplace. I want to thank specially Michael Schröder, Alexander Scherrer, Thomas Hanne, Rafael Velásquez, Michael Monz, Anthon Winterfeld, Paul Miki Willy, Ivan Serna and all the members of the optimization department of the ITWM for offering me always the required advise and/or criticism. Thanks to all my friends who made our time in Germany fruitful, to God who is, even when I am not and specially to my wife who suffered more than me during this time.

Abstract

This dissertation deals with the optimization of the web formation in a spunbond process for the production of artificial fabrics. A mathematical model of the process is presented. Based on the model two kind of attributes to be optimized are considered, those related with the quality of the fabric and those describing the stability of the production process. The problem falls in the multicriteria and decision making framework. The functions involved on the model of the process are non linear, non convex and non differentiable. A strategy in two steps; exploration and continuation, is proposed to approximate numerically the Pareto frontier and alternative methods are proposed to navigate the set and support the decision making process. The proposed strategy is applied to a particular production process and numerical results are presented.

Contents

Chapter 1

Introduction

1.1 Motivation

Imagine the problem of painting a long wall. It does not seems to be a difficult problem, but only because I have not yet specified the restrictions you need to fulfill. First of all, you have to use a very special kind of brush. When this brush is applied on the wall following a straight trajectory, it does not paint a solid stripe of color but something similar to what is depicted in Figure 1.1.

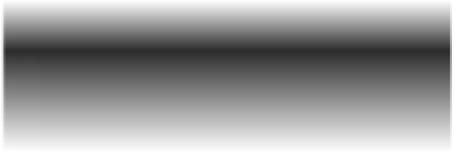


Figure 1.1: Painting a straight line with our brush.

As it is shown, the brush paints different parts of the trajectory with different intensities. In other words, different amounts of painting material are applied at different spots. The brush does not uniformly paint the wall. So if we want a uniform color, each time we apply the brush we need to use different trajectories. We will assume that, with each additional pass of the brush, the intensity of the color will increase directly. This assumption may not be realistic in the case of painting, but for the problem we will consider in the dissertation it is.

To make things a bit more complicated, assume no complete control of the brush trajectory. In our particular problem, the brush rotates and moves forward at the same time, i.e. it follows a cycloid.

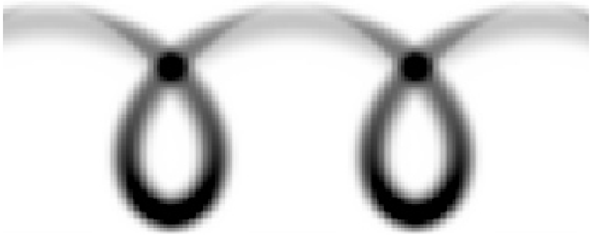


Figure 1.2: Cycloid trajectory of the brush.

1.2 Related Problems

The above described situation reflects the main concepts required in the problem we are considering in this dissertation. It belongs to a family of problems, which in some works is called *Uniform Coverage Problems.*

A typical example of a uniform coverage problem; the one most related to our particular application, is the problem of modeling the paint deposition and finding optimal trajectories to describe the way a robot is used to paint a surface in the automotive industry. For some references about this problem see [9].

Similar models and ideas can be used to obtain insight, and perhaps even interesting solutions, for any kind of process where material is deposited over a surface. An interesting example is the problem of optimal irrigation. In large gardens or farm fields, the watering process is considered a seriously matter. Here the material is water, and the surface is the soil. It is important to maintain a certain amount of humidity in the soil and this humidity level should be maintained between prescribed bounds. Keeping the amount of water in soil below (or above) a critical interval for long time periods will have a negative impact on the development of the plants. An interesting project about optimizing the watering in gardens is described in [44]. In the same direction, the deposition of any kind of substance (seeds, fertilizer) could also be modeled using similar approaches.

The process inspiring this dissertation comes from the production of artificial fabrics; or *nonwovens*, as they are called in the industry [16]. In these processes, a polymer is melted and formed into fibers which are then deposited over a conveyor band. The result of this procedure is a kind of web (figure 1.3), where fibers are placed in different positions and orientations. It is expected that the distribution of fibers should be homogeneous across the fabric. The presence of white or black spots; as is the case in figure 1.3 is not desirable.

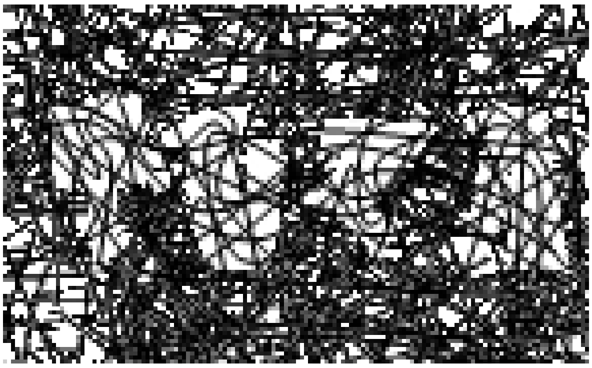


Figure 1.3: Web formed by random position of fibers.

What all the mentioned applications have in common is the requirement of applying the material over a surface uniformly. The reasons are not always the same, but usually a lack of material in a particular region is considered as a sort of defect. On the other hand, a high concentration of material means a waste of resources. This have a direct impact in the costs of the process. Improving the uniformity distribution of fibers in any process becomes then a very important goal.

There is an additional aspect, that I will consider in this dissertation and it is not considered in any of the related applications I found so far. This is the inclusion of an analysis of the stability of the process of applying material over a surface, due to its stochastic nature.

We will see that, even when it is possible to design a process generating almost perfect uniform webs, such process will be very unstable, in the sense that small perturbations in the parameters describing the process will have a large effect in the uniformity of the fabric. Processes are not perfect, they are prompt to errors due to a variety of causes and there is no escape to the fact that uncertainties will happen, even if the most accurate tools are used. The stability (or sensitivity) of the process becomes then a characteristic that should be optimized too. This sensitivity can be understand as the effect of perturbations in the parameters of the process.

These considerations define two distinct goals or objectives to be attained in our process. On the one hand, we want to produce uniform fabrics for a high quality product, but on the other the process should be stable.

Unfortunately these two goals will be in conflict. This means that the design of a process producing highly uniform fabrics will not be the most stable and the most stable process will not produce uniform fabrics.

1.3 Multicriteria and Decision Making

The presence of more than one objective to be optimized brings the problem into a different area. A single process has to be chosen; not the one producing the most uniform fabric and not the most stable, but one where a compromise is attained between those two objectives. A person making such a decision is called a *decision maker*. It should be a person (or group) with experience in the process and the ability to make rational judgments to compare different alternatives.

More and more, computers are being used in the industry and other sectors to assist in the process of taking a final decision. A computational system designed to help in these problems is called *Decision Support System* (DSS). In some kind of problems, when a mathematical model for the considered process is available. The system may make use of such model to assist the decision making activities. A DSS based around a model is called *Model Driven DSS*. One of the aims of this dissertation is to propose strategies useful in the design of such a system.

The goal of these dissertation is not to perform multicriteria optimization for an isolated production process, but to facilitate the process of decision making by developing methods and tools around a decision support system when the functions involved are nonlinear and specially for non differentiable problems. Such methods and tools will be applied to the particular kind of uniform coverage problem related with the production of artificial fabrics.

The realization of those will require an understanding of the theory of multiobjective optimization. Multiobjective optimization deals with the problem of optimizing more than one function at the same time, and is one of the most important mathematical theories applied in the area of decision making. We will see that there exist a subset of the set of all possible states of our process that is of interest to the decision maker. It is called the *Pareto set*. The Pareto set is; roughly speaking, the alternatives that are not worst than any other considering all objectives. Such alternatives are also called "efficient" designs or efficient solutions.

In the case of two criteria, a particular goal is to compute enough efficient designs. Enough in the sense that any other efficient design can be obtained using interpolation between two of the already obtained ones. The decision maker can simply explore this subset and choose the solution he thinks is the best. We will also propose some alternatives to explore the Pareto set, even if the number of required designs is too large.

These goals are not trivial, specially in our particular problem. Similar methods already exist for situations where the functions involved are linear. An example is the develop of tools for decision support and Pareto navigation applied to the problem of radiotherapy planning, see [34].

The strongest difficulty in our application is the lack of convexity. The appearance of a non convex problem complicates the tasks required in multiobjective in many ways. One of the obvious difficulties is the appearance of several local optima. Even if a single objective is considered, obtaining the global optimum of such an optimization problems is, in most of the cases, an impossible task. For these problems heuristics (evolutionary algorithms, scatter search, etc.) are usually the only adequate approach [49]. The consideration of additional objectives make the problem even more challenging.

1.4 Outline

The dissertation is structured in 6 chapters.

In chapter 2 I will give an introduction to the concept of *nonwovens*, its definition according to the experts and some of its uses. Section 2.2 will explain some of the details related with a particular kind of process used to produce nonwovens. The steps involved on such process (spinning, web formation and bonding) will also be described. Section 2.3 will cover some general aspects about quality control and in particular the quality attributes expected in the final product of these processes. We will see that several attributes are desired in the final fabric. In the same section we will consider the stability of the process as one of the required attributes. If a process is unstable, the product generated may not satisfy the specifications imposed. The presence of several objectives to be optimized creates a conflict in the quality. Section 2.4 will make present this conflict as a natural result comparing it with other different classical examples.

Chapter 3 gives a detailed description of a mathematical model of the production process presented previously in chapter 2 together with ways to compute the quality attributes required in the process. Section 3.1 develops a general model of deposition of material for an abstract "brush" depositing material over a surface. It is general enough to be applied in all situations where the addition of material on a point results in a direct increase of the density. The model will be applied to our example spunbond process in section 3.2, where such process is described in detail by separating it in several steps. Section 3.3 presents the alternatives to measure the homogeneity of the fleece using the output of the production process, some different criteria will be proposed to estimate the different kind of defects appearing in the real fabrics. In section 3.4 the two kind of control parameters that will be used for the optimization are explained, the *radial distribution* and the *shift synchronization*. Those are related with the trajectory of the object applying the material and how the material is extruded. Section 3.5 propose a way to estimate the stability or sensitivity of the process using the concept of *condition number*. Two different ways of computing the condition numbers are presented and the care that should be taken when attempting to apply such concepts to estimate a real sensitivity is considered. In section 3.6 the multicriteria problem will be motivated.

Having more than one criterion brings our problem in the context of multicriteria optimization. In chapter 4 such mathematical theory is presented. Section 4.1 covers the most important concepts of the theory, define the concept of efficiency of solutions and the Pareto set. It will present also a way to describe the Pareto set and to transform a multicriteria problem in a family of scalar problems. Section 4.2 explains a method used to obtain successive points in the Pareto set using ideas coming from algebra for the approximation of implicitly defined differential manifolds. They are called *continuation or homotopy methods*. Here I will propose a derivative free continuation method to avoid the requirements of gradients or Hessians usually needed in the application of those methods. For the derivative free method I will need to use scalar methods not based on differentials, but those will be presented in chapter 5. Section 4.3 will give estimations of the interpolation error obtained if the Pareto set is approximated using the continuation methods. If the errors are considered to high we should consider to make a refinement in the approximation. Section 4.4 explains the problems of approximating the Pareto set when the problem is non linear and several local minima are expected. We will propose to use heuristics for the exploration of the space and continuation methods for the local computation of efficient solutions.

Chapter 5 presents a collection of scalar optimization methods that will be used in the optimization of a single criterion. In this case, the homogeneity.

But as I have mentioned, scalar methods are also important to be coupled with the continuation method of chapter 4. Section 5.1 propose a linear program LP to solve the problem of optimizing one of the parameters, the radial distribution. The lineal program can be solved in seconds using a standard solver. Section 5.2 propose two ways of solving the problem of optimizing the shift synchronization. First using a mixed integer program, which will happen to be too slow to be applied in the continuation method. And a different approach using Fourier approximation, which will return good (but not optimal) solutions with no use of numerical methods. Section 5.3 presents a mixed integer program for the problem considering both kind of parameters, but it is also too slow to be used in the multicriteria framework. Section 5.4 will present three heuristics for scalar optimization. The first heuristic is an evolutionary algorithm. It will be used to perform exploration of the space. The next two heuristics (Pattern Search and Nelder Mead) are used for the local search once an exploration was performed. The Pattern search method posses convergence properties. Even when Nelder Mead do not have such properties, it works very well in practice and is very useful when sets of solutions are already computed close to the optimum. Nelder Mead will also inspire a method for interactive navigation of the Pareto set.

The concepts of decision making are presented in chapter 6. The mathematical theory presented in the previous chapters will be used to propose three different ways of navigating the Pareto set for nonlinear problems. Section 6.1 refers to the general concepts of decision making and introduce the concept of a *Decision Support System* which is a computational program used to assist in the process of making a decision. Section 6.2 describes the necessary concepts for allowing navigation of the Pareto set. It presents the widgets for the graphic user interface used to interact with the system. The most direct way of exploring the Pareto set, is to compute a database of solutions, and allow the user to select a point from it. This can be done using visualization widgets. A most sophisticated way, is to allow the user to select a vector of weights and let the system compute the corresponding Pareto point. This is an option, but it is not considered a “friendly one”. A different alternative is to give the user the opportunity to select a goal value for one of the objectives and let the system the task of finding a solution close to the one required by the user. This can be done using any kind of scalar optimization. The last and most appealing is the use of the Nelder Mead algorithm to indirectly optimize a *value function* which is a hidden function that is assumed to exist and reflect the preference of the decision maker among the solutions. The method requires a valuation of a set of solutions from the user, or simply an “acceptance” or “rejection” of solutions. The geometry of the Nelder-Mead algorithm can be used to propose new solutions to substitute the ones rejected by the user.

In the final chapter. The results obtained from all the methods presented in the theoretical chapters are presented. Section 7.1 will describe the details of the numerical simulation of the model presented in chapter 3. I will also propose ways to increase the speed of the evaluation of the criteria to allow faster navigation. Section 7.2 explains the three different sensitivities considered as criteria. Two for each kind of parameter, and one that mixes those two in a single value. Section 7.3 is short, but very important, because it suggest and justify the simplification of the radial distribution to a two parameter family of functions. We will see that such family of functions is more realistic, reduce the computations, improves both the stability and sensitivity of the function that was used in practice and it is almost as good as the optimal function obtained considering a piecewise linear approximation with much more parameters. Section 7.4 presents the results of all algorithms mentioned in chapter 5 related with the optimization of a single criterion. Finally in section 7.5 an instance of a process is considered, and the Pareto frontier is approximated in a way that allows interpolation between the computed solutions. Such result allow the use of the real navigation of the frontier.

Chapter 2

The Production Process

Introduction

The concept of "nonwovens" is for many people a new one, and only few persons know about the actual production of those. Specially about the one we will consider in this project.

The purpose of this chapter is to introduce the concepts concerning the production of nonwovens and the methods of production. I will need to describe with more detail the specific process we will consider (*spunbonding*) and which parts of this we are modeling (*web-formation*). In this chapter I will explain the actual production process. Later we will develop a mathematical model, simulate and optimize the design of this process.

In section 2.1 the basic concepts related to the production of nonwovens are presented, together with the definition of the term "nonwoven". Section 2.2 describes the particular spunbond processes and the stages of such processes; spinning, web formation and bonding. The section 2.3 deals with the concept of "quality" in general and in the context of artificial fabrics. We discuss the quality characteristics or attributes expected in the product (nonwoven fleeces) together with one extra requirement usually wanted on every production processes, stability. Section 2.4 explains the conflicts arising from considering more than a single objective. In our case we consider at least the reduction of defects in the final product and the stability of the process. We will see how the appearance of more objectives affects the optimization. This section will prepare the frame for the rest of the chapters, inspiring the inclusion of multicriteria theory and decision making.

2.1 Nonwovens

The term *nonwoven* is relative new. But in a way, it explains itself. It refers to certain kind of sheet product or artificial fabric that is not like the typical woven one. Nonwovens are usually not based on yarns, but contain filaments of some polymer like polyester or nylon. Paraphrasing, a nonwoven is a fabric that was not woven.

In other contexts the word is hyphenated *(non-woven)*, but in almost all the literature and publications on the topic of nonwovens, it is written as a single word. Nonwovens are a central topic of this work, and I will use it without hyphen.

There are, at least, two recognized definitions for nonwovens which are provided by the two most important associations of nonwovens in the world, the *Association of the Nonwovens Fabrics Industry, INDA* and the *European Disposables and Nonwoven Association, EDANA* [14, 23].

Definition of Nonwoven (INDA): *Nonwovens are a sheet, web or batt of natural and/or man-made fibers or filaments, excluding paper, that have not been converted into yarns, and that are bonded to each other by any of several means.* The bonding methods are: addition of adhesives, thermal fusion of the fibers, fusion of fibers by dissolving and resolidifying the surfaces, creation of physical tangles, stitch usage for the fibers [23].

Definition of Nonwoven (EDANA): *Nonwovens are a manufactured sheet, web or batt of directionally or randomly oriented fibers, bonded by friction, and/or cohesion and/or adhesion, excluding paper or products which are woven, knitted, tufted, stitchbonded incorporating binding yarns or filaments,* *or felted by wetmilling, whether or not additionally needled. The fibers may be of natural or man-made origin. Thay may be staple or continuous or be formed in situ.[14]*

As stated in the definitions, the filaments in nonwovens are bonded using several methods. A simple alternative is to “glue” them using adhesives. In other processes, the fibers are partially fused and thermally bonded. Sometimes chemicals are added to melt the materials. There also exist mechanical ways of doing it, e.g. creating tangles or tuft on the fibers or stitching the filaments. In any case, what they have in common in the absence of a woven process. In this way, the production of nonwovens is usually faster, compared with a typical yarn process. The reader interested on nonwovens may consider [4, 11, 14, 23] for further information and references.

2.2 Spunbond Process

Spunbond is a popular and effective process used to produce nonwovens. In spunbond processes, the polymer is melted and formed in fibers using a special die. The melted material is extruded trough this die, called *spinneret* which generates filaments of polymer not yet solidified. The filaments; or fibers, are then separated and cooled down using air flows and/or electrostatic charges and are then deposited over a conveyor band. As a result, the fibers are transformed into a web in one step combining the spinning and the web formation. The web is then bonded in a further step, using one of the several processes mentioned before. A nice reference for the spunbond process can be found in [4]. Figure 2.1 depicts the main steps of the process.

2.2.1 Spinning and Web Formation

The deposition of the melted fibers can be performed in very different ways, and several patented methods exist to do this. In some cases the fiber trajectories are altered conveniently using systems of deflecting surfaces or mirrors. The image on the left in figure 2.2 shows a method using rotating deflector

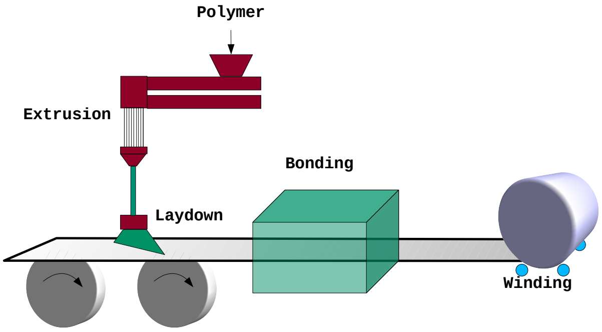


Figure 2.1: Schematic of a Spunbond Process

planes to spread the fibers and deposit them over the moving band. The image on the right illustrates a method where fibers are deposited both in machine direction and in cross direction[1]. We are interested on a method that deflects, in a similar way, a bunch of fibers at the same time in a rotational trajectory, similar to the first system on the left of figure 2.2.

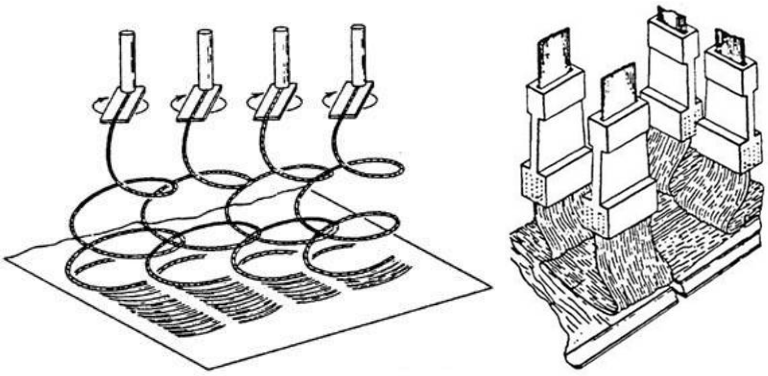


Figure 2.2: Some deflector systems.

2.2.2 Bonding Process

After the web is formed, the fabric is still rather weak to be used and additional treatments have to be performed on the web, in order to increase

[1] Images from *"http://www.apparelsearch.com"* used with permission.

its strength and other desired characteristics. Typically a bonding process is then performed. The bonding can be performed by mechanical needling of the fibers, applying thermal energy to fuse and adhere the fibers or by using chemicals. A different approach is the creation of tangles, stitch bonding, ultrasonic fusing and hydraulic entanglement. In some cases two or more of this processes can be applied on the same fabric.

Unfortunately, if the formed web is very irregular, there is no bonding method able to correct it. In our project we will not study the different bonding methods. Our problem will be restricted to the optimal web formation. But we still need to understand, what is an "optimal web"? To answer this question, we need to consider how the experts judge the quality of nonwovens.

2.3 Quality of the Process

In the manufacturing of products or in services the concept of "quality" plays a central role. There are different ways of defining the concept of "quality"; conformance to requirements, fitness for use, etc [33]. In the design stage it is usual to speak about *off-line quality.* This refers to the selection of optimal parameters in order to obtain a process fulfilling certain required or expected characteristics. We separate these characteristics in two, those inherent to the product and those related with the process in general.

2.3.1 Fleece Quality

Nonwovens are used for hundreds of products in different industries like construction, automotive, agriculture, landscaping, industrial, military, heath care, leisure, clothing, geotextiles, household, among many others. The industry has been growing consistently, year after year, in the last few decades.

Experts measure the quality of the fleece in different ways, according to different criteria, depending on the final use. See [19, 29, 41]. All the desired properties of a web, formed using the previously mentioned process, are related; or at least correlated, with a single aspect or characteristic of the fleece.

Namely, the web should “look alike” at every single spot. The “look alike” property means that, if we take two samples from the web in any position and any orientation, both samples should look similar, as much as possible. The mechanical and visual properties of the fabric depend on this kind of “self-similarity”. In this way, there are two main kinds of defects that can be observed in the web independently of the kind of real application of the fabric.

The first kind of defect is called *cloudiness*, and is described as a high (or low) concentration of fibers in a particular region of the fabric [38, 54]. This concentration creates “dark” or “light” clouds noticeable to the eye. These clouds can have the form of spots, stripes or any other kind of shape. Dark clouds and light clouds have different implications in relation to the quality of the product. For some products light clouds are important for the quality but dark clouds are more related with waste of material. For others, both dark clouds and light clouds have an effect in the quality and performance of the product.

When many parallel fibers stick together in a spot, the fabric shows anisotropies or *ships*, like it is called in the argot. These are the second kind of defects.

The clouds and ships reduce the quality of the fabric and are judged in most cases by the visual inspection of experts. The automatization of this process require the use of interesting mathematical models and image analysis tools, see [38].

A good model should be able to measure as objectively as possible these quality characteristics. Clearly an ideal fabric is uniform in intensity of fibers and in its orientations, what it is not so clear yet is the best way to measure this quality, and how to compare the levels of quality of two webs produced by two different sets of parameters.

2.3.2 Process Stability

If we attempt to measure the quality of our product at different moments, we will find that the values obtained are not the same every time. There is always a natural scattering in the measured values due to variability inherent to the process. Variability is part of any production process and this fact has to be taken into account in any analysis performed. Every process is affected by different factors like methods, parameters, equipment, material, etc. Some of these factors are controllable; in particular the machine parameters, others are either not under control or are almost impossible to keep in control, like the environment conditions.

If small variations in the factors defining the process generate relative high variations in the output of the process, we say that the process is not stable. On the other hand, if the variations in the output are small, the process is considered as stable.

The production process should not be launched until it has been proved to be stable. Therefore it is a high priority to consider measurements of the stability for every process. A process is labeled as "stable" when it runs in a predictable way. Typically the stability is measured by performing a sensitivity or uncertainty analysis comparing the output of different instances of the process.

It is not enough for a process to generate a single sample with the desired quality characteristics mentioned in the previous section. A high quality process has to be able to repeat this characteristics during the entire production time. This is in fact the main problem in quality control of processes [33, 35].

We will see in chapter 3 different mathematical tools, used to approximate numerically the stability of a model. Once we have a mathematical model of the process, we can analyze the effects of variability or perturbations in the machine parameters and study the resulted variation in the output of the process.

2.4 Quality Conflict

Quality attributes are measurements or characteristics used to define the quality of a product or service [33]. In quality control there are two main goals in relation with quality attributes. On the one hand the control of the expected value of the attribute and on the other the reduction of its variance. In many cases these two tasks are in conflict.

There is usually a desired fixed value, and the expected value of our attribute; as a random variable, should be as close as possible to it. This desired value is usually referred as the *specification* for the product. If we were producing a particular kind of geometric piece, the desired values are the requested dimensions for that piece. In our process we are considering the cloudiness as an attribute, and we expect it to be zero in order to have no defects in the fleece.

The second goal in quality control is to reduce the variability of the process. As we saw in the previous section, no process is perfectly accurate and variations exists due to the different kind of elements involved. If the attribute value is considered an instance of a random variable, the variability is modeled with the variance of its probability distribution.

The problem of controlling the expected value and reduce the variance of a variable is not an isolated issue. In portfolio theory; for example, one of the aims is to obtain a high expected value of profit with a low risk value; ideally zero, of a particular investment [52]. Typically the increasing of expected gain has, as a consequence, an increase on the risk, creating a trade-off between both goals. In these sort of situations there is no way to make an optimal choice. The trade-off between the goals force us to make a compromise, a decision.

The design of a high quality process in our application involves also a compromise. On the one hand, we would like to define the process in such a way that the expected cloudiness of the produced fleece is as low as possible. The reduction of the cloudiness is in direct relation with the uniform distribution of fibers over the fleece and has a direct impact in the amount of material required to meet the specifications. In other words, less cloudiness implies material savings and reduction of costs. But such reduction should be maintained during the whole production time. This goal requires a stable process with low variance.

A naive way of improving the expected cloudiness of the fleece is to increase the speed of the process. In fact such increase will reduce the expected cloudiness, but at the same time will make the process more unstable and create high variances in the output. The appearance of conflicting goals in a problem make impossible, in general, to find an optimal solution. As was mentioned before, a decision has to be made. There is a full mathematical theory coping with these kind of problems; the theory of multicriteria programming or multicriteria optimization. Chapter 4 is completely dedicated to multicriteria. In business and management the mathematical tools of multicriteria optimization are used to support the process of decision making. Decision making represents a whole area of research and applications. Chapter 6 is fully dedicated to decision making and decision support systems, inspired on the conflict presented in this chapter.

Chapter 3

Mathematical Model

Introduction

In this chapter a mathematical model for the production process presented in the previous chapter is developed. We will use this model to measure and estimate the quality of the product and the stability of the process.

In section 3.1 a general abstract model of material deposition is presented, that could be applied in other kind of processes. Section 3.2 describe and model the specific production process we are considering. The aim of the section is to obtain a density function to represent the distribution of fibers on the final fabric. The problem of modeling or measuring the quality of the fabric using the proposed density function will be considered in section 3.3, we will see the different kind of defects expected in the fleece and how to measure them. Section 3.4 will briefly explain the parameters of the process that will be used as controls in the optimization. The stability of the process will be considered in section 3.5 using sensitivity analysis. We propose the use of the condition number to estimate the sensitivity, apply the ideas to our model and compare two approaches; an analytical and a stochastic. The last section motivates in few paragraphs the inclusion of the rest of the chapters, the multicriteria theory and the decision making.

3.1 Material Deposition

Consider a segment AB of fixed length moving in a trajectory over $\mathbb{R}^2$. If the length of the segment is constant, the description of the trajectory can be done by specifying the trajectory of one of the points; say A, and the orientation of the segment.

Consider a trajectory in the real plane,

$$\begin{aligned} \gamma : \mathbb{R} &\rightarrow \mathbb{R}^2 \\ t &\mapsto \gamma(t). \end{aligned}$$

If $\gamma(t)$ represents the position of the point A, and $\theta(t)$ the orientation of the segment at time t, with respect to some fixed axis, then the trajectory γ_P of every point P on the segment AB can be obtained from γ and θ as,

$$\gamma_P(t) = \gamma(t) + \frac{|AB|}{|AP|} u(\theta(t)). \tag{3.1}$$

Where u is a unit vector defined as $u(\theta) = (\cos\theta, \sin\theta)$.

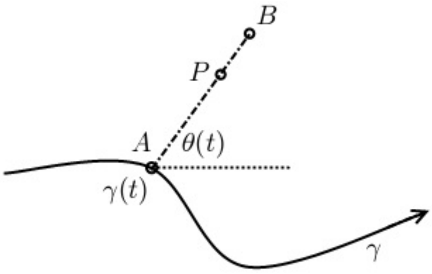


Figure 3.1: Trajectory of a segment AB.

We define a function $\rho : \mathbb{R}^2 \rightarrow \mathbb{R}$ assuming that each time the trajectory of a point P in the interval AB passes over a point $\overline{x} \in \mathbb{R}^2$, the value of $\rho(\overline{x})$ is increased according to a flow function depending on P. The following definition describe that function.

Definition 1. *Consider a segment AB in a trajectory described by $\gamma(t)$ and $\theta(t)$ in a time interval $t \in [0, T]$, and a function $f : [0, 1] \rightarrow \mathbb{R}$. The* density function $\rho : \mathbb{R}^2 \to \mathbb{R}$ *is defined as,*

$$\rho(\overline{x}) := \sum_{P \curvearrowright \overline{x}} f\left(\frac{|AP|}{|AB|}\right) \cdot \|v^{\perp}\|^{-1}, \overline{x} \in \mathbb{R}^2$$

where $P \curvearrowright \overline{x}$ means that the point P on the moving segment is located over the point $\overline{x}$ in the fixed plane and $v^{\perp}$ is the component of the velocity of P orthogonal to the segment AB.

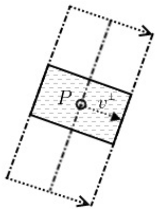


Figure 3.2: Position of P and orthogonal component of velocity.

The value of ρ is proportional to the function f; which is used to model the flow of material through P and inversely proportional to the velocity $v^{\perp}$. I consider an inverse proportion due to the fact that by doubling the speed, we deposit the same amount of material in a region of double area, hence the density will become half of the original value, and the projection, because it is proportional to that area.

This model for material deposition assumes that the orthogonal component of the velocity $v^{\perp}$ never vanishes This is, that the velocity is never parallel to the segment. If the velocity component is zero and the function f is nonzero, the model predicts an infinite density of material. This is, of course, not the case in the real process, because material is not deposited over a point but over a region. For the numerics in chapter 7, we make a partition of the domain and consider an average speed for the simulation. Nevertheless, in practice if the velocity is zero (or close to zero) the density of material will simply go to very high values.

3.2 Deterministic Model of the Process

We will see now a model for a spunbond process based in the material deposition model of the previous section. The set of parameters or controls describing our process can be represented as a vector $\alpha = (\alpha_1, \alpha_2, ..., \alpha_{n_p}) \in \mathbb{X} \subset \mathbb{R}^{n_p}$, where $\mathbb{X}$ is called *Parameter Space* and we consider n_p parameters. We also have n_c criteria quantities (not defined yet) as functions of the parameters. The criteria values are arranged as a vector $\kappa = (\kappa_1, ..., \kappa_{n_c}) \in \mathbb{Y} \subset \mathbb{R}^{n_c}$, where $\mathbb{Y}$ is called the *Criteria Space.* The criteria are meant to measure the desired properties or quality attributes of the process and of the final product. An important aim is to describe each criterion as a function of the parameters.

$$\begin{aligned} F : \mathbb{X} &\rightarrow \mathbb{Y} \\ \alpha &\mapsto \kappa \end{aligned}$$

If we have this function, we will be in a position to apply optimization algorithms to improve the process with respect to those criteria. But the relation between α and κ is far from being a simple one, let alone finding a closed or algebraic expression. Nevertheless, the function F can be decomposed in a natural way into two parts. The first part will map the parameters into a function space using a model or simulation of the production process, the function will be a kind of density representing the concentration of fibers in the fabric. For this aim, we will use the model for material deposition described in the previous section. The second part will quantify the criteria from the output of the simulation and from the model itself having in mind the considerations used by quality experts in the nonwoven industry already mentioned in section 2.3.

The fleece is generated by hundreds of spinnerets arranged in rows. Each spinneret deposits fibers over a moving band in a particular way. The number of rows varies from machine to machine, but usual values range from 6 to 15. We will need first to model the material deposition for a single spinneret, and then use it to describe the density function representing the final fleece.

3.2.1 Fiber Deposition for a Single Spinneret

Each spinneret rotates and deposit fibers over the conveyor band. At the same time the band moves below the spinneret with constant speed. The flow of fibers deposited depends on the distance from the center of rotation. Each spinneret behaves like the moving segment explained in section 3.1, we can describe the behavior then using the same expressions.

Keeping the same notation, we model the spinneret as a segment depositing material as it moves. In our particular process one extreme of the segment; modeling the spinneret, follows a linear trajectory with respect to the band according to

$$\gamma(t) = (v_b t, 0), \tag{3.2}$$

at the same time, the spinneret rotates around that point with constant speed. The orientation of the segment is then,

$$\theta(t) = \omega t, \tag{3.3}$$

where v_b is the speed of the conveyor band, and ω is the rotational speed of the spinneret.

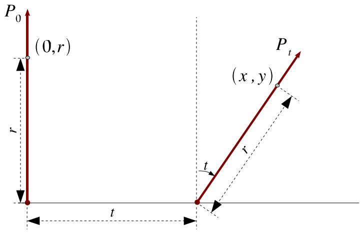


Figure 3.3: Rotating y-axis.

Consider an orthogonal coordinate system based on the band, with the negative x-axis oriented in the direction where the band moves, and the y-axis orthogonal to the band. Assume that at time $t = 0$ the center of the spinneret is located at the origin of that coordinate system and oriented towards the positive y-axis. From 3.1 we can obtain the trajectory of an arbitrary point in the spinneret at time t, see figure 3.3.

The spinneret is in fact fixed, but the effect of moving the band to the left is equivalent to moving the spinneret to the right. We can imagine the trajectories better if we fix our coordinate system (and perspective) to the moving band.

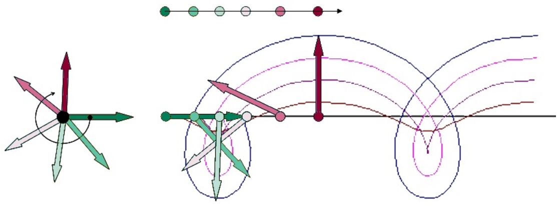


Figure 3.4: Cycloid trajectories are composition of rotational and translational movements.

The composition of linear and rotational movements generates a particular kind of curve trajectories known as *cycloids* [7, p.101]. Figure 3.2.1 depicts some of these possible curves. The parameterizations of cycloids curves are described in the following lemma.

Lemma 1. *The position of a point P in the spinneret at a distance r from the rotation center after time t with respect to the coordinate system based on the band is,*

$$\overline{x}(r,t) = \begin{pmatrix} x(r,t) \\ y(r,t) \end{pmatrix} = \begin{pmatrix} v_b t + r \cdot \sin(\omega t) \\ r \cdot \cos(\omega t) \end{pmatrix}. \tag{3.4}$$

From lemma 1 we obtain an explicit algebraic relation between the position $\overline{x} = (x, y)$ in the band, the distance r of the point in the spinneret passing over $\overline{x}$ and the time t,

$$\begin{aligned} x = & \quad v_b t + r \cdot \sin(\omega t) \\ y = & \quad r \cdot \cos(\omega t). \end{aligned} \tag{3.5}$$

Given a position (x, y) on the band. If we are interested on recovering the position r in the spinneret and the time t when such point passes over (x, y), we need to solve the system 3.5 for r, t.

Canceling the trigonometric functions, we obtain the equation,

$$r = \sqrt{(x - v_b t)^2 + y^2}. \tag{3.6}$$

Which can be used to obtain the position r in the spinneret given the time t. On the other hand, we can cancel r from the system and obtain the equation,

$$v_b t + y \tan(\omega t) - x = 0. \tag{3.7}$$

The time t can be obtained by solving equation 3.7.

This equation is compact but difficult; if not impossible, to solve analytically. If we want to solve it, numerical methods will be required. A plot of equation 3.7 is shown in figure 3.5.

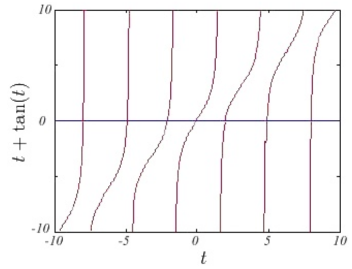


Figure 3.5: Trigomometric equation 3.7 for $x = 0$, $y = 1$, $v_b = 1$ and $\omega = 1$

It looks very similar to the tangent function, with infinitely many translations of a similar branch, and this is true due to the fact that

$$v_b\left(t + \frac{\pi}{\omega}\right) + y \tan\left(\omega(t + \frac{\pi}{\omega})\right) = v_b t + y \tan(\omega t) + \frac{v_b \pi}{\omega}.$$

For this graph the branches are translated in the y direction too. This means, we can solve

$$v_b t + y \tan(\omega t) + k\frac{v_b \pi}{\omega} = 0, k \in \mathbb{Z}$$

using a single branch instead if we want.

The solutions are the intersections of the graph with the $x-$axis, and it is clear that there is one solution for each branch of the function. Then we have an infinite countable set of times t solving 3.7 and the system 3.5. But from 3.6 we have $r \to \infty$ as $|t| \to \infty$ and in practice, the r is bounded by the spinneret radius R_s, and then only finitely many solutions will be needed.

In order to finish the description of the trajectory of the spinneret, we need also the expression for the velocity of the points. Such expression can be easily obtained by partial differentiation of the position vector with respect to time.

Lemma 2. *Under the assumptions of lemma 1 the velocity vector of a point in the spinneret is,*

$$\overline{v}(r,t) = \begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} v_b + r\omega \cdot \cos(\omega t) \\ -r\omega \cdot \sin(\omega t) \end{pmatrix}. \tag{3.8}$$

To complete the analysis of the fiber deposition due to a single spinneret we need to describe the way the spinneret deposits fibers. We assume the flow of fibers depends on the distance from the center of rotation. This dependence is described using a distribution.

Radial Distribution

The spinnerets deposit fibers while rotating. Fibers are deposited at a certain distance from the center of rotation, generating different concentration of material depending on the distance from the spinneret. A function f describes this deposition. It is considered a non-negative smooth function with support on the closed interval [0, 1].

Definition 2. *The* radial distribution *is considered as a non-negative continuous function f with support on the closed interval* $[0,1]$*, i.e.*

$$f : \mathbb{R} \to \mathbb{R}_0^+, x \notin [0,1] \Rightarrow f(x) = 0, \tag{3.9}$$

The value $f(x)$ represents the flow of fibers at a distance xR_S from the center of rotation of the spinneret. Where R_s is the maximal distance where flow exists.

So far, in the tuning of the real process, the radial distribution was considered as a fixed function. One of our proposals for the design of the process is the use of the radial distribution as a further control parameter. Clearly, the realization of a system of deflector mirrors to mimic any arbitrary function, is not a realistic task. We will suggest to consider a simple and robust family of radial distributions in the optimization. A family of functions for which the exact fitting is not crucial.

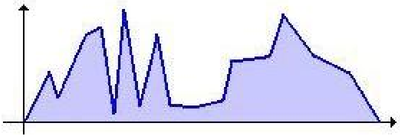


Figure 3.6: A "non-realistic" radial distribution using a discretization.

The considered radial distributions are of the form,

$$p(t) = \begin{cases} 0 & \text{if } 0 \leq t < a \\ \frac{1-cos(\pi \frac{x-a}{b-a})}{2} & \text{if } a \leq t < b \\ cos(\frac{\pi}{2} \frac{x-a}{b-a}) & \text{if } b \leq t \leq 1 \end{cases} \tag{3.10}$$

This family of functions is parameterized using two numbers $a, b \in \mathbb{R}$. The first number a represent the minimum distance where the spinneret delivers flow and b is the location of the peek of the function; the position where the flow is maximum. The function (3.10) is depicted in figure 3.7.

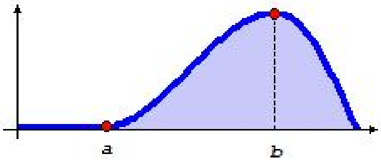


Figure 3.7: Two point profile with trigonometric splines.

Arguments in favor of such change will be presented in chapter 7. But we have at least three reasons for adopting this kind of radial distributions.

As mentioned it is not expected to be able to design systems of deflector planes to mimic every kind of function for the radial distribution, see figure 3.6. Functions like the one depicted in figure 3.7 are more realistic. Another reason is the reduction in the number of parameters required, besides we will see that such functions approximate very well the optimal radial distribution in a more general space. The complexity of the algorithms will be reduced in that way. Finally, such functions will be sufficient to obtain processes with density function almost perfectly homogeneous.

When considering the flow of fibers as $f(x)$, we are assuming an average behavior. The flow of fibers in spunbond process, and in general, is a stochastic process that cannot be avoided. The fibers dynamics are strongly affected by air flows, turbulence, deflecting surfaces and by the interaction of fibers between themselves, see [30]. For the purpose of this thesis, we consider this flow as a random variable whose expected value is described by the radial distribution. Later, we will consider the perturbations on the radial distribution in order to study the stability of the process.

With the trajectory of the spinneret and the radial distribution described we are in position to obtain an expression for the density function generated with a single spinneret.

Lemma 3. *Consider a spinneret trajectory described in lemma 1 and the radial distribution for the spinneret $f : [0,1] \to \mathbb{R}^+$ representing the flow of fibers from definition 2. Then, the density of fibers generated by the spinneret can be obtained as,*

$$\rho_0(\overline{x}) := \sum_{t \in T'} f\left(r/R_s\right) \cdot \|v^{\perp}\|^{-1} \tag{3.11}$$

Where T' is the set of all times when the spinneret passes over $\overline{x}$ and the orthogonal component of the velocity is obtained from the position $\overline{x}$ and the velocity $\overline{v}$ of the corresponding point in the spinneret,

$$v^{\perp} = \overline{v} - \overline{x}\frac{\overline{x} \cdot \overline{v}}{\|\overline{x}\|^2}.$$

The position $\overline{x}$ can be obtained from the position r in the spinneret and the time t according to 3.5 and 3.6. The velocity $\overline{v}$ is obtained from lemma 2. There is a couple of facts we can say about the density function ρ_0 generated by a single spinneret.

Lemma 4. *The density function ρ_0 is periodic in the x component with period*

$$L_x = 2\pi \frac{v_b}{\omega}.$$

Proof. Let (x, y) be a point on the band. Assume that at time t the point r in the spinneret passes over (x, y). From 3.5, at time $t + \frac{2\pi}{\omega}$ the point r in the spinneret is positioned over the point (x', y') on the band according to,

$$x' = \left(t + \frac{2\pi}{\omega}\right) v_b + r \cdot \sin\left(t\omega + \frac{2\pi}{\omega}\omega\right) = x + L_x$$

$$y' = r \cdot \cos\left(t\omega + \frac{2\pi}{\omega}\omega\right) = y$$

In a similar way from lemma 2 the velocities at both times are the same. Considering the definition of ρ_0 from lemma 3, the flow over the two points is always the same and $\rho_0(x, y) = \rho_0(x + L_x, y)$. □

Because the center of rotation of the spinneret runs over the x-axis, and R_s represents the maximum distance where fibers are deposited, we have the following result.

Lemma 5. *The spinneret delivers no material beyond a vertical distance of R_s. In other words, if $R_s < |y|$ then $\rho_0(x, y) = 0$.*

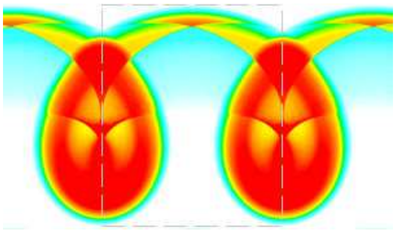


Figure 3.8: Density function for a single spinneret.

Depicted in figure 3.8 is a representation of the density function ρ_0. The color map, sometimes called *lookup table* represents the relative values of a real function. The lookup table goes from white(zero) through cyan(low), green(average), yellow(high) to red(too high) as shown in figure 3.9. I use this lookup table for almost all the visualizations of density functions.

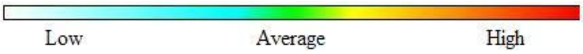


Figure 3.9: Lookup table used for the density functions.

3.2.2 Fiber Deposition for a Row of Spinnerets

As already mentioned, the fleece is produced by the effect of hundreds of spinnerets arranged in rows delivering the fibers over the moving band. If we assume that every spinneret behaves similarly, we can obtain the density function representing the final fleece by adding up translated copies of the density function ρ_0.

Consider a single row of infinitely many spinnerets working perpendicular to the band movement, all starting with the same orientation. If L_y is the separation between the spinnerets in the row, the density function generated is described as follows.

Definition 3. *The* row density *function is defined as,*

$$\rho_1(x,y) = \sum_{i=-\infty}^{+\infty} \rho_0(x, y + iL_y) \tag{3.12}$$

where ρ_0 is the spinneret density.

The row density function has also the following properties, in particular it is periodic in both directions, as stated in the next lemma.

Lemma 6. *The density function ρ_1 for a single row of infinitely many spinnerets, is periodic in both directions. i.e, for L_x, L_y given above,*

$$\rho_1(x,y) = \rho_1(x + L_x, y + L_y)$$

for every $x, y \in \mathbb{R}$.

Proof. The periodicity in the x component follows directly from lemma 4. The periodicity in the y component is obtained by changing the index of the sum to $(i+1)$.

$$\rho_1(x,y) = \sum_{i=-\infty}^{+\infty} \rho_0(x, y+iL_y) = \sum_{i+1=-\infty}^{+\infty} \rho_0(x, y+(i+1)L_y) = \rho_1(x, y+L_y).$$

□

As a result, it is enough to study this function in the rectangle defined by the periods

$$R_\rho := [0, L_x] \times [0, L_y]. \tag{3.13}$$

We may also be interested on proving that the row density is bounded. This may not be clear from the definition, because the function is the result of an infinite sum of functions. Nevertheless, lemma 5 avoids infinite accumulation of values.

Lemma 7. *The density function ρ_1, generated by a row of spinnerets, is bounded.*

Proof. From lemma 5, if $|y+iL_y| > R_s$, then $\rho_0(x, y+iL_y) = 0$. Consider the index set $I := \{i \in \mathbb{Z} : |y+iL_y| < R_s\}$. The set is clearly finite and therefore,

$$\rho_1(x,y) = \sum_{i=-\infty}^{+\infty} \rho_0(x, y+iL_y) = \sum_{i\in I} \rho_0(x, y+iL_y) < \infty$$

for every point (x,y). □

The distribution of fibers generated by a single row of spinnerets is not very uniform. In order to improve the uniformity of the process and fill the clouds generated from a single row, several rows are used in parallel to increase the covering of the band.

3.2.3 Generating the Final Fleece

The final fleece is generated by N_r rows of spinnerets. Each row deposits fibers over the band according to ρ_1, but each of them does it in a different position.

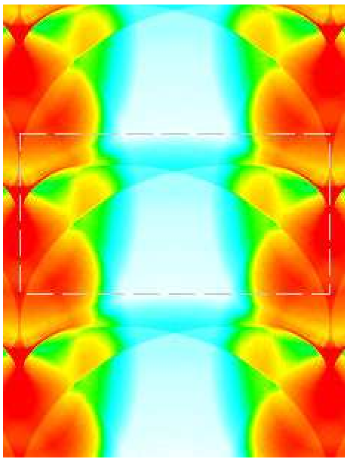


Figure 3.10: Density function for a single row of spinnerets.

The function describing the final density of fibers is then a superposition of N_r translated copies of the function ρ_1.

Definition 4. *The* density of fibers *is defined as a function* $\rho : \mathbb{R}^2 \to \mathbb{R}^+$,

$$\rho(x,y) = \sum_{k=1}^{N_r} \rho_1(x + s_k^x, y + s_k^y) \tag{3.14}$$

where (s_k^x, s_k^y) *represents the corresponding translation of the* $k-$*th row of spinnerets with respect to the first one.*

In our process the translations in the $y-$direction are given by $s_k^y = kL_y/N$, and we will only use the translations s_k^x (or simply s_k) as controls. The definition assumes that all rows of spinnerets are rotating with the same constant speed ω and spinnerets in the same row start with the same orientation. Nevertheless spinnerets on different rows may (and are expected to) have different orientations. These angles, or shifts, can be used to control the translations.

Definition 5. *The orientation of the spinnerets in the row* i*, with respect to row* 0 *is defined as* phase shift *and expressed with* φ_k^{phase}.

If d_k is the distance between the $k-$th row and the first one modulo the

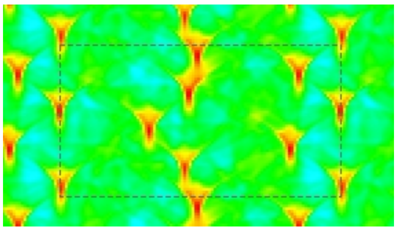


Figure 3.11: Example of a final density function.

period L_x, then it is possible to define the relative orientation of the spinnerets in a row with respect to the moving band.

Definition 6. *The* band shifts *are the relative orientations of the rows with respect to the band and are defined as,*

$$\varphi_k^{band} = \varphi_k^{phase} + \frac{d_k}{L_x} 2\pi. \tag{3.15}$$

The linear translation s_k can be computed from the band shifts

$$s_k = \frac{\varphi_k^{band}}{2\pi} L_x \tag{3.16}$$

The value $\rho_0(x, y)$ represents the density of material (fibers) in the position (x, y) deposited by the matrix of spinnerets. The fleece looks more like a web (see figure 3.12) and the density of fibers in very small regions of the fleece is, of course, zero. In spite of this, I will call this function a *density function*, and can be considered a density in the mathematical model, because the radial distribution p models the deposition of fibers as if it were a continuous material.

An alternative is to think in such a function as a probability or as the expected amount of fibers in a (not too small) region of the fleece. In any case, we will avoid to perform computations in small domains, which is the same approach for any kind of density applied in science.

The density function ρ is our model for the final fleece. In the next section we consider the quantifications of the quality attributes of a fleece mentioned in chapter 2.

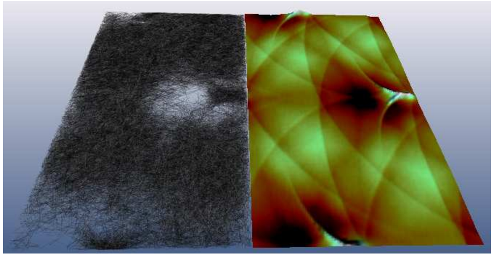


Figure 3.12: Fleece as fibers and as a density function.

In section 2.3 we mentioned the quality attributes required in the production of nonwovens. We have divided such attributes in two. On one side, the expected quality of the fleece. Which depended, as mentioned, on the homogeneity of the distribution of fibers across the band. In particular, we have the "cloudiness" as the most important kind of defect appearing in nonwovens. On the other side, we also need to keep the process in control. A certain degree of stability is required in the process. The next sections will use the model for fiber deposition already described, in order to quantify the quality attributes of homogeneity and stability.

3.3 Fleece Homogeneity

The fleece is modeled as a density function representing the distribution of fibers. This density function approximates the expected amount of material; fibers in our case, at points in the plane. I have mentioned that, experts measure the quality of the fleece by taking samples of the final web, searching for the presence of either dark or light clouds.

3.3.1 Clouds

Relative large regions with high or low concentration of fibers are called "clouds". Usually they consider clouds of more than 3 centimeters as defects. The absence of clouds in the fleece means that the value of the density function is always equal to its average value, i.e. it is a constant function.

As it is easy to imagine, it will be impossible, even in the model, to obtain a constant density function. What we can attempt, is to minimize the deviation of such function from the ideal constant one.

The total flow of fibers passing through the spinneret is consider as constant. The radial distribution reflects only the deviation of the trajectories of the bunch of fibers, but does nothing to increase or reduce the total flow. This means that, at the end, the average amount of fibers can be considered as constant too.

The ideal homogeneous function is then defined as the constant function with value equal to the average of the density function ρ, i.e.

$$\overline{\rho}(x,y) := \frac{1}{|R_\rho|}\int_{R_\rho} \rho(x,y) \equiv \overline{\rho} \in \mathbb{R} \tag{3.17}$$

where R_ρ is the periodic domain 3.13 and $|R_\rho|$ is its area.

If we want to measure the deviation between functions the most direct way of doing it, is to define a norm in the function space where such functions live. And the first idea when looking for a norm for a space of measurable functions is to use an L_p norm.

$$\|\rho - \overline{\rho}\|_p = \left(\int_{R_\rho} |\rho - \overline{\rho}|^p\right)^{1/p}, \quad p > 0, f \in \mathbb{W}. \tag{3.18}$$

Measuring the deviation using an L_p norm give us a global idea of the deviation between two functions. Nevertheless it was shown in [38] that such a number does not describe the quality of a fleece and the deviation from the homogeneity in the way it is proper. A simple reason is that an L_p norm does not distinguish between a big cloud region and many small ones. According to experts, the former is worst.

Defects on the fleece are present when the density of fibers is considerably below the average (or above it) in a relative large region. Regions with low density of fibers create light clouds and regions with a high density of fibers create dark clouds. Both are considered as defects in the fabric.

With this in mind, we may think that the infinity norm, a norm measuring maximum and minimum values, can be more useful.

$$\|f\|_\infty = \max_{\alpha \in \mathbb{W}} |f(\alpha)|. \tag{3.19}$$

The infinity norm does not distinguish between accumulation or lack of fibers.

If we want to measure only how the values of the density function go below the average we can try the following,

$$\|\rho - \overline{\rho}(\alpha)\|_{\infty_-} = \min_{\alpha \in \mathbb{W}} (\rho(\alpha) - \overline{\rho}). \tag{3.20}$$

This is, however, not a norm, but a semi-norm.

Similarly the expression

$$\|\rho - \overline{\rho}(\alpha)\|_{\infty_+} = \max_{\alpha \in \mathbb{W}} (\rho(\alpha) - \overline{\rho}). \tag{3.21}$$

can be used to measure the deviation from above the average.

There is a detailed explanation about how to judge the quality of a fleece with respect to the clouds in [38]. As I mentioned, experts usually consider clouds of sizes of more than 3 cm as defects. But this could depend on the type of product. This means that in order to estimate the quality of the fleece with respect to the clouds, it is better to measure the density of material on regions of the fleece and not on points. We will then transform our density function ρ, in the following way.

$$\rho^*(x,y) = \int_{x-\delta_x}^{x+\delta_x} \int_{y-\delta_y}^{y+\delta_y} \rho(\tau_x, \tau_y) d\tau_x d\tau_y \tag{3.22}$$

If we write $\xi = (x, y)$ and $\tau = (\tau_x, \tau_y)$, the transformed function ρ^* can be described in a compact form,

$$\rho^*(\xi) = \int_{\mathbb{R}^2} \rho(\tau) q(\xi - \tau).$$

Where the function $q : \mathbb{R}^2 \to \mathbb{R}$ is defined conveniently as,

$$q(\xi - \tau) = \begin{cases} 1 & \text{if } |x - \tau_x| < \delta_x \text{ and } |y - \tau_y| < \delta_y \\ 0 & otherwise. \end{cases} \tag{3.23}$$

As we observed, this transformation is nothing but the convolution of ρ with the function q, this is,

$$\rho^* := \rho * q.$$

Having this convoluted function as a more realistic function, bring us other advantages. By choosing an alternative smooth q function, we can obtain a ρ^* function as smooth as required.

Now, the minimum and maximum values of ρ^* are better measurements for the presence of light and dark clouds of radius δ.

Definition 7. *The criterion* light cloudiness κ_{min} *is defined as,*

$$\kappa_{min} = \overline{\rho^*} - \min_{x \in \mathbb{R}^2} \rho^*(x)$$

Definition 8. *The criterion* dark cloudiness κ_{max} *is defined as,*

$$\kappa_{max} = \max_{x \in \mathbb{R}^2} \rho^*(x)$$

Where $\overline{\rho^*}$ is the mean value of ρ^* in the domain, i.e.

$$\overline{\rho^*} = \frac{1}{|R_\rho|} \int_{R_\rho} \rho^*(x, y) dx dy.$$

Clearly, κ_{min} and κ_{max} are to be minimized. The general cloudiness can be defined by simply using the infinity norm for that deviation, or from both the dark and light cloudiness.

Definition 9. *The criterion* cloudiness (or homogeneity) κ_{cloud} *is defined as,*

$$\kappa_{cloud} := \|\rho^* - \overline{\rho^*}\|_\infty = \max\{|\kappa_{min} - \overline{\rho^*}|, |\kappa_{max} - \overline{\rho^*}|\}. \tag{3.24}$$

3.3.2 Stripes

A special kind of defect appears when the clouds extend along full lines. Dark and light stripes are then visible on the fleece. This kind of defect is expected because of the periodic nature of the process.

Stripes are not expected to appear in arbitrary orientations, but only in a finite number of possible angles if we are interested on thicker stripes only.

The proof requires the equidistribution theorem,

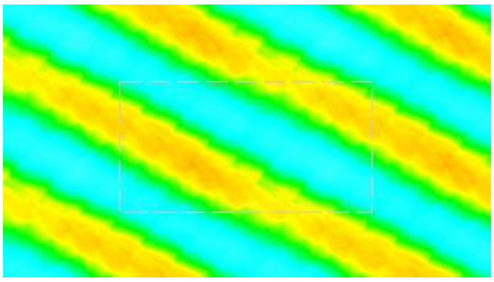


Figure 3.13: Density function with stripes.

Theorem 1 (Bohl 1909, [5]). *The sequence*

$$\{n\alpha \operatorname{mod} 1\}_{n\in\mathbb{N}}$$

is uniformly distributed on the unit interval, when α is an irrational number.

The equidistribution theorem implies directly the following lemma.

Lemma 8. *If the periods of the fleece density are L_x and L_y, the only possible slopes of the stripes are rational multiples of the number L_y/L_x, i.e. if m is the slope of the stripe,*

$$m = \frac{a}{b}\frac{L_y}{L_x}, a, b \in \mathbb{Z}.$$

If m is an irrational multiple of L_y/L_x, the center line of the stripe will cover the full domain densely. Which means that the stripe will cover the entire domain. As a consequence no stripes with slope m irrational are expected.

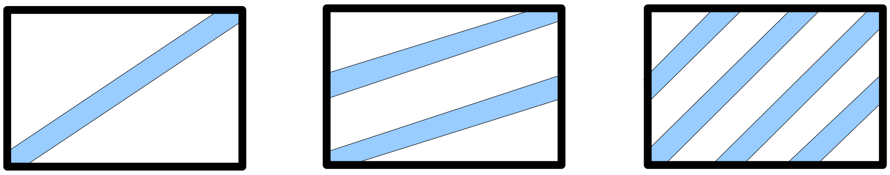


Figure 3.14: Stripe patterns for (a,b)=(1,1), (a,b)=(1,2) and (a,b)=(3,2).

Assume we are interested on stripes of width δ_s or wider only, and there appear N stripes in a period. Then it is clear that $N\delta_s$ should be smaller than the diagonal of the periodic rectangle domain. From this follows the following result.

Lemma 9. *If there exist N stripes of width δ_s in the domain, then*

$$N < \frac{\sqrt{L_x^2 + L_y^2}}{\delta_s}.$$

Which means that only a finite number of rational multiples could become slopes for the stripes.

The minimum and maximum criteria do not give us information about stripes. A possibility to give a measure for these defects is to consider projections of the density function ρ along particular directions.

Let us say we would like to measure the presence of vertical stripes for a particular density function ρ. The function $\rho_y^{ave}(x)$ defined as

$$\rho_y^{ave}(x) := \frac{1}{L_y} \int_0^{L_y} \rho(x, \xi) d\xi,$$

represents the average of ρ along a vertical line passing trough $(x, 0)$. This function has period L_x.

Depicted in the left side of figure 3.15 are this projection function for the density function shown in figure 3.13. As we can see this projected function looks rather homogeneous, and can be interpreted as an absence of vertical stripes. But the picture on the right side shows a similar projection along lines parallel to the diagonal of the rectangle R_ρ. The presence of diagonal stripes is clearly depicted there.

We can interpret from this graphic the effect of the diagonal stripes on figure 3.13.

Definition 10. *The* projected function *using a general slope $m \neq 0$, is defined as,*

$$\rho_m^{ave}(\xi) = \frac{1}{L_x} \int_0^{Lx} \rho(x, \xi + mx) dx$$

A single number can be used to estimate the deviation of such functions with respect to the constant function.

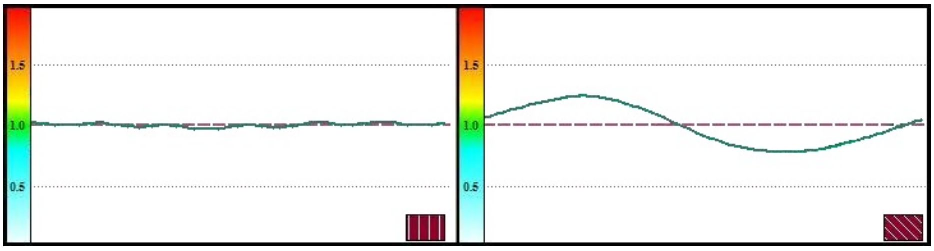


Figure 3.15: Vertical and diagonal projections of density function.

Definition 11. *The criterion measuring the presence of stripes with slope m is,*

$$\kappa_{stripe}^{m} := \|\rho_m^{ave} - \overline{\rho}_m^{ave}\|_2$$

where $\overline{\rho}_m^{ave}$ is the mean value of the function in the interval.

In this case, we use an L_p norm with $p = 2$, but any other value could be used.

3.3.3 Ships

As I mentioned before in section 2.3.1, ships are a type of defect on fleeces where a region in the web is populated with a majority of fibers in a particular direction. Also called, anisotropy, in some contexts. Having homogeneity in the fibers does not necessarily means an absence of ships. The measuring of ships is more complicated than the measure of clouds. In order to have some information about it, we would require to know about the orientation of the fibers in the web.

Anyhow, the way fibers are deposited in our special kind of spunbond process; in particular the rotating system of deflecting mirrors, have as an effect, a natural homogeneous scattering in the directions of the fibers. Which is perhaps one of the principal reasons to use this kind of systems. Even when it is possible to model the fiber orientation, I consider the appearance of ships in the fleece as controlled indirectly by the own nature of the process.

3.4 Control Parameters

For our particular process we will consider only two kinds of parameters. Those describing the fibers deposition for a single spinneret, we call it *radial distribution*, and those used for the synchronization of the rows of spinnerets, the *shifts*.

3.4.1 Radial Distribution Discretization

The spinnerets deposits fibers while rotating. Fibers are deposited at a certain distance from the center of rotation, generating different concentration of material depending on the distance from the spinneret. We already define in section 3.2.1 the radial distribution as a function describing this deposition. It is considered a non-negative smooth function with support on the closed interval $[0, 1]$, and an element of a function space, $p \in \mathcal{C}_0(\mathbb{R}, \mathbb{R})$. For the numerical implementation it will be required to define a discrete space for it.

The straightforward way is to use a piecewise linear approximation. To this end, we use a partition of the unit interval using a finite number of points $x_i \in [0, 1]$, say $x_i = \frac{i}{N_p+1}$ with $i = 0, ..., N_p + 1$ and we consider only radial distributions which are linear combinations of a set of basis functions π_i. The basis function are the popular "hat functions" with support on the interval $[0, 1]$ defined as,

$$\pi_i(x) = \begin{cases} \frac{x-x_{i-1}}{x_i-x_{i-1}} & \text{if } x \in [x_{i-1}, x_i] \\ \frac{x-x_{i+1}}{x_i-x_{i+1}} & \text{if } x \in [x_i, x_{i+1}] \\ 0 & \text{otherwise} \end{cases}$$

Figure 3.16 depicts a typical hat function.

Definition 12. *A* piecewise linear radial distribution *is a function of the form,*

$$p_\beta(x) = \sum_{i=1}^{N_p} \beta_i \pi_i(x).$$

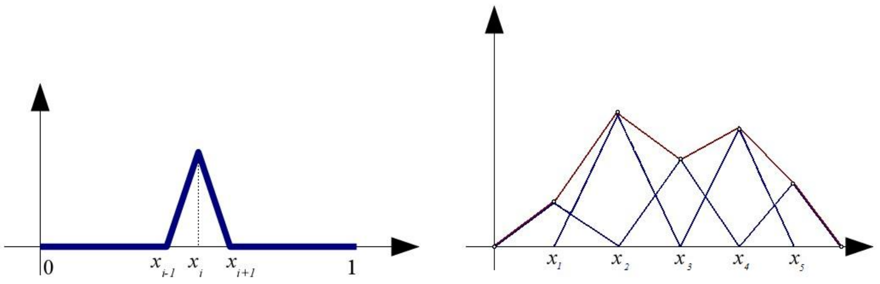


Figure 3.16: A Basis function and a radial distribution formed using five basis functions.

Where N_p is the number of discretization points and we consider only positive coefficients. From the definitions, a piecewise function can be used as a radial distribution. In this form, radial distributions p are formed by linear combinations of the basis functions π_i. The vector of coefficients $\beta = (\beta_1, ...\beta_{N_p})$ characterize each radial distribution. This approach can be used to approximate any Lipschitz continuous function.

3.4.2 Shifts Synchronization

The second kind of parameters are used to describe the synchronization of the rows of spinnerets. This synchronization is measured by the corresponding angles for each row of spinnerets, the band shifts defined in section 3.2.3. If there are N_r rows, the space of shifts could be considered as $\mathbb{R}^{N_r}$ if we allow angles from $-\infty$ to $+\infty$. If we want instead to work modulo 2π, the space is $\mathbb{S}^{N_r}$, thus $\mathbb{S}$ is the set $[0, 2\pi]$, where we identify 0 with 2π. In other words, the unit circle.

3.5 Sensitivity Analysis

Every model is nothing more than an approximation of the reality. When in a model we consider a distance of 10 centimeters or an angle of 35 degrees, we should not forget that in the real process, such distance is never exactly 10 centimeters nor the angle precisely 35 degrees. A serious model should take into account the fact that measurements should not be modeled using numbers until we have proven that the errors are significantly small. Where "significantly small" depends on the particular problem and the "errors" are not necessarily only measurement errors, but also inconsistencies coming from simple the fact that the model is no the real process.

A much realistic approach is to consider measurements as random variables, with a given probability distribution. In many cases, normal distribution is the common choice. But this is usually an assumption and other distributions could be used according to the particular model.

If we consider these quantities as random variables, the outcome or result of the model has to be considered as a random variable too. Being random variables, the values coming from the model or simulation are also not precise, but have a certain variance. The variance in the output of a model is related with the variance of the input parameters, or all the quantities we are considering in it.

In *sensitivity analysis* we attempt to understand the relation between these two variances. The variance of a value reflects the amount of perturbation we expect in it. If relative small perturbations in the parameters of the model creates large perturbations in the output, we say that the process is *sensitive*. If not, the model is usually called *robust*. This analysis will deliver us information about the stability of the process being modeled.

Sensitivity analysis is used mainly for two purposes. To know up to which extent we can trust in our model, and to discover which control parameters have more effect on the desired output. This can, as a consequence, be useful in its optimization. The interested reader may consider [43] for gaining some insight in modeling and sensitivity analysis.

3.5.1 The Condition Number

The quantification of the degree of sensitivity or stability of a model, algorithm or function is done with the so called, *condition number*. It is also used to determine the liability to numerical computation, because it measures the effect of small errors in the calculations, see [13], [24], [55].

Consider a function $f(x) = y$ representing a mathematical problem, where the point $x \in \mathbb{X}$ represents a set of *input parameters* or data and $y \in \mathbb{Y}$ is the *output* or solution of the problem. The condition number allows us to estimate the sensitivity by considering the effects that small perturbations in the input parameters have in the output of the model.

Definition 13. *Let $\mathbb{X}, \mathbb{Y}$ be normed vector spaces. Consider a map between the spaces $f : U \subset \mathbb{X} \to \mathbb{Y}$, U open and*

$$V_\delta(x) := \{x' |\ \|x' - x\|_{\mathbb{X}} < \delta\} \in U$$

for $\delta > 0$ a δ-neighborhood of x. The value

$$\kappa^{\delta}_{abs}(f, x) := \sup_{x' \in V_\delta(x)} \frac{\|f(x') - f(x)\|_{\mathbb{Y}}}{\|x' - x\|_{\mathbb{X}}}$$

is called the absolute condition number *of (f, x) in V_δ. If both x and $f(x)$ don't vanish then*

$$\kappa^{\delta}_{rel}(f, x) := \kappa^{\delta}_{abs}(f, x) \frac{\|x\|_{\mathbb{X}}}{\|f(x)\|_{\mathbb{Y}}}$$

is called the relative condition number *of (f, x) in V_δ.*

This numbers describe the maximum effect of perturbations in the domain of the function, and becomes clearer if we rewrite the expressions,

$$\begin{array}{rcll} \|f(x') - f(x)\| & \leq & \kappa_{abs}\|x' - x\| & \\ \dfrac{\|f(x') - f(x)\|}{\|f(x)\|} & \leq & \kappa_{rel}\frac{\|x'-x\|}{\|x\|} & x' \in V_\delta \end{array}$$

Both condition numbers give us a bound factor relating the expected perturbation in the output $\delta f := \|f(x') - f(x)\|$ given a perturbation in the input $\delta x := \|x' - x\|$.

For problems considering quantities, like amount of material in this case, the absolute condition number may not be the best choice. The effect of the perturbation should be measured as a percentage, not as an absolute quantity.

The relative condition number κ_{rel} is then proper choice in that case. We can get κ_{rel} from κ_{abs} multiplying it by $\frac{||x||}{||f(x)||}$.

For a general non linear function f the actual computation of the condition number could become a difficult task in most cases. Because either we are unable to compute the values of the function in every point of neighborhood V_δ or we do not have means to bound these values with information about f, with only a finite number of evaluations of the function.

As the definition of the condition number reflects, the numbers represents an understanding of the local behavior of the function f. In this sense it is not difficult to see the relationship with the differentiability of the function (if f happens to posses a degree of differentiability).

Analytical Approach

Analytically the condition number of smooth functions is estimated using a Taylor approximation. In this case, the condition number can be estimated using the norm of the gradient of f for a first order approximation and for a second order, the Hessian will be involved.

For a function $f : \mathbb{R}^n \to \mathbb{R}$ the first order Taylor approximation is given by,

$$f(x_0 + \delta x) = f(x_0) + \nabla f(x_0) \cdot \delta x.$$

Theorem 2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function and define the affine function,*

$$g(x) := f(x_0) + \nabla f(x_0) \cdot x.$$

Then the condition number of $(g, 0)$ is given by,

$$\kappa^\delta_{abs}(g, x) = ||\nabla f(x_0)||.$$

For any x and δ.

Proof. The proof is straightforward,

$$\begin{aligned} \kappa^\delta_{abs}(g, x) &= \sup_{x' \in V_\delta(x)} \frac{|g(x') - g(x)|}{||x' - x||} \\ &= \sup_{x' \in V_\delta(x)} \frac{|\nabla f(x_0) \cdot (x' - x)|}{||x' - x||} \end{aligned}$$

We now that $|\nabla f(x_0) \cdot (x' - x)| \leq \|\nabla f(x_0)\| \, \|(x' - x)\|$ with equality if

$$(x' - x) = \alpha \nabla f(x_0)$$

for some $\alpha \in \mathbb{R}$. Such vector can be found in any neighborhood, therefor,

$$\begin{aligned} \kappa_{abs}^{\delta}(g, x) &= \frac{\|\nabla f(x_0)\| \, \|(x' - x)\|}{\|x' - x\|} \\ &= \|\nabla f(x_0)\|. \end{aligned}$$

□

As we see, a simple way of estimating the condition number and the sensitivity is to use this local gradient approximation, given that our function is differentiable.

This approach is, of course, not valid when the first order approximation is not accurate. A typical situation occurs when the point x_0 is an extreme point or close to one. Here the first approximation will return a sensitivity value of zero, but the real sensitivity could be very high, depending on the size of neighborhood being considered. Figure 3.17 depicts a function with gradient zero but with a non-zero sensitivity.

Specially for this kind of points using a second order Taylor approximation can add more information for the sensitivity.

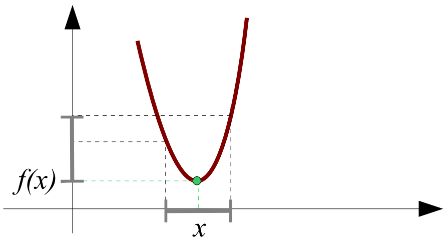


Figure 3.17: Function with zero gradient but non-zero condition number.

Theorem 3. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function and define the quadratic function,*

$$q(\xi) := f(x_0) + \nabla f(x_0) \cdot \xi + \frac{1}{2} \xi^T \nabla^2 f(x_0) \xi.$$

Then the condition number of $(q, 0)$ *is bounded by,*

$$\kappa^{\delta}_{abs}(g, 0) \leq \|\nabla f(x_0)\| + \|\nabla^2 f(x_0)\|\delta.$$

Where $\nabla^2 f$ *is the Hessian matrix, the norm of the gradient is the usual Euclidean norm for vectors and the norm for the Hessian is the matrix norm defined as,*

$$\|A\| = \max\{\|Ax\| : x \in \mathbb{R}^n, \|x\| = 1\}.$$

Proof.

$$\begin{aligned}
\kappa^{\delta}_{abs}(q, 0) &= \sup_{\xi \in V_\delta(0)} \frac{|q(\xi) - q(0)|}{\|\xi\|} \\
&= \sup_{\xi \in V_\delta(0)} \frac{|\nabla f(x_0) \cdot \xi + \frac{1}{2}\xi^T \nabla^2(x_0)\xi|}{\|\xi\|} \\
&\leq \sup_{\xi \in V_\delta(0)} \frac{|\nabla f(x_0) \cdot \xi|}{\|\xi\|} + \sup_{\xi \in V_\delta(0)} \frac{|\frac{1}{2}\xi^T \nabla^2(x_0)\xi|}{\|\xi\|} \\
&\leq \sup_{\xi \in V_\delta(0)} \frac{\|\nabla f(x_0)\| \, \|\xi\|}{\|\xi\|} + \sup_{\xi \in V_\delta(0)} \frac{\|\frac{1}{2}\xi^T \nabla^2(x_0)\| \, \|\xi\|}{\|\xi\|} \\
&= \|\nabla f(x_0)\| + \|\nabla^2 f(x_0)\|\delta.
\end{aligned}$$

□

The bound of theorem 3 is in fact the condition number. The "worst case" can be achieved by choosing a point at a distance of δ from x_0 and in the direction of the gradient.

Stochastic Approach

If we consider the input parameters as random variables, the output of the process becomes also a random variable. The purpose of the sensitivity analysis is to obtain the probability distribution of this output random variable. Or more exactly, to obtain the uncertainty of the output, say, in terms of the variance.

In some cases, if the relationship between input and output is simple enough, the explicit expression for the distribution of the output can be obtained. But most of the times, this distribution is obtained experimentally using Monte Carlo simulations. For this aim, assumptions in relations to the distribution of the input parameters have to be made. Later, we will use these approaches for our particular problem and will return to this points.

3.5.2 Model Sensitivity

If we express the model described in section 3.2 as a function or mathematical problem mapping an input u to an output κ, we will be able to compute the condition number and a measure of the sensitivity for it.

In our model, u represents the set of parameters defining the process; the positions, orientations, and flows used to construct the model of fiber deposition. As an output κ, I consider a single value. The homogeneity (cloudiness) which is the most significant quality function.

The mathematical problem has the following form,

$$\begin{aligned} \mathcal{F}: \quad \mathbb{X} &\to \mathbb{R} \\ u &\mapsto \kappa, \end{aligned}$$

where u encodes all the information related to the parameters of the process.

If we apply now the definition of the condition number to this function we get an expression of the form,

$$\kappa^{\delta}_{abs}(\mathcal{F}, u) := \sup_{u' \in V_\delta(u)} \frac{|\mathcal{F}(u') - \mathcal{F}(u)|}{\|u' - u\|_{\mathbb{X}}}.$$

Here, we are confronted with a delicate issue. The vector u encodes distances, angles and flows from the process, but we need to define a norm $\|u' - u\|_{\mathbb{X}}$ for the design space $\mathbb{X}$. Even when it is easy to define a mathematical norm for this space, the question is if such a norm has a sense in the reality.

The value of the Euclidean norm for the vector $(3, 4)$ is 5, but if the vector represents 3 centimeters and 4 grams, the number 5 has no meaning at all, unless we find a way to compare centimeters and grams.

I will consider then, a condition number for each kind of parameter of the process.

Definition 14. *The sensitivity of $\mathcal{F}$ with respect to the radial distribution $p \in \mathcal{C}_0^\infty(\mathbb{R}, \mathbb{R}^+)$ is given by,*

$$\kappa_{abs}^{\delta}(\mathcal{F}, p) := \sup_{p' \in V_\delta(p)} \frac{|\mathcal{F}(p') - \mathcal{F}(p)|}{\|p' - p\|_\infty}. \tag{3.25}$$

The radial distribution is a continuous function and we use an infinity norm to measure the deviations between functions.

Definition 15. *The sensitivity of $\mathcal{F}$ with respect to the orientation shifts $\sigma \in \mathbb{R}^{N_r}$ is given by,*

$$\kappa_{abs}^{\delta}(\mathcal{F}, \sigma) := \sup_{\sigma' \in V_\delta(\sigma)} \frac{|\mathcal{F}(\sigma') - \mathcal{F}(\sigma)|}{\|\sigma' - \sigma\|}. \tag{3.26}$$

In this case, the shifts are encoded in a vector and we use an standard Euclidean norm for the deviations.

It is also possible to obtain analytical expressions for the condition numbers. The criterion homogeneity measuring the cloudiness was defined as the maximum deviation of ρ with respect to its average value, this is,

$$\|\rho - \overline{\rho}\|_\infty.$$

As a function, the infinity norm is not differentiable. But we can find arbitrary approximations by using a p-norm with large p values.

Lemma 10. *Let f be a bounded function then,*

$$\lim_{p \to \infty} \|f\|_p = \|f\|_\infty.$$

The p-norm tend to be closer to the ∞-norm for function where the maximum is clearly over the average. This suggests that it will be an adequate choice when the density function reflects the presence of clouds.

To test the accuracy of this approximation I run 500 random instances of our simulation and compared the values of some different p-norms with the infinity norm. The correlation for some of them is depicted in figure 3.18. The larger the value, the best is the approximation. But computationally, more than a 10-norm tend to produce numerical errors in the algorithm,

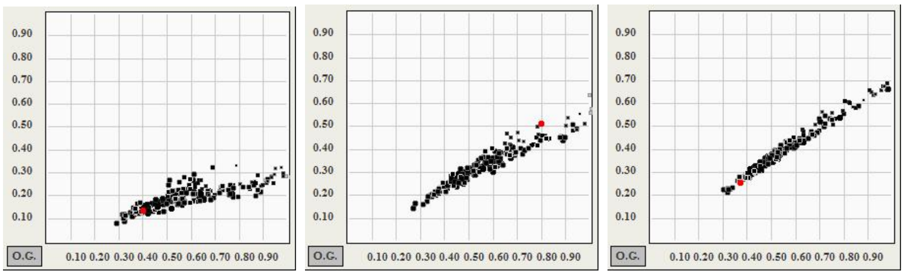


Figure 3.18: Comparison between p-norms and ∞-norm for $p = 2, 10, 20$.

not to mention the complexity of the operations. The errors appear due to the necessity of performing operations with too large or too small numbers. Anyhow, once we consider a differentiable function we have the possibility to estimate the condition number by computing the gradient of our function.

Consider the function, $\mathcal{F}_p : \mathbb{R}^n \to \mathbb{R}$ returning the expression

$$\mathcal{F}_p(u) = \left(\int_\Xi (\rho(\xi) - \overline{\rho})^p d\xi \right)^{\frac{1}{p}},$$

where Ξ is the periodic domain of the density function ρ and $u \in \mathbb{R}^n$ encodes the control parameters for the process. If ρ is smooth, $\mathcal{F}_p$ is differentiable and it approximates the original $\mathcal{F}$ for large values of p.

We should not forget that ρ is also a function of u. The next lemma relates the partial derivative of the smooth criterion function $\mathcal{F}_p$ with the partial derivatives of the density function in a fixed point.

Lemma 11. *The partial derivative of $\mathcal{F}_p$ with respect to the parameter u_j is given by,*

$$\frac{\partial \mathcal{F}_p}{\partial u_j} = \|\delta_u\|_p^{1-p} \int_\Xi \delta_u^{p-1} \frac{\partial}{\partial u_j} \rho(\xi) d\xi \tag{3.27}$$

Where $\delta_u := \rho - \overline{\rho}$.

Proof. The proof follows easily,

$$\begin{aligned}
\frac{\partial \mathcal{F}_p}{\partial u_j} &= \frac{\partial}{\partial u_j}\left(\int_\Xi (\rho(\xi)-\overline{\rho})^p d\xi\right)^{\frac{1}{p}}, \\
&= \frac{1}{p}\left(\int_\Xi (\rho(\xi)-\overline{\rho})^p d\xi\right)^{\frac{1-p}{p}} \frac{\partial}{\partial u_j}\int_\Xi (\rho(\xi)-\overline{\rho})^p d\xi, \\
&= \frac{1}{p}\|\rho-\overline{\rho}\|_p^{1-p}\int_\Xi \frac{\partial}{\partial u_j}(\rho(\xi)-\overline{\rho})^p d\xi, \\
&= \frac{1}{p}\|\delta_u\|_p^{1-p}\int_\Xi p(\rho(\xi)-\overline{\rho})^{p-1}\frac{\partial}{\partial u_j}(\rho(\xi)-\overline{\rho})d\xi, \\
&= \|\delta_u\|_p^{1-p}\int_\Xi \delta_u^{p-1}\frac{\partial}{\partial u_j}\rho(\xi)d\xi.
\end{aligned}$$

□

If we assume we already have the result of the simulation, this is, the function ρ and therefor δ_u. The computation of expression 3.27 requires only the estimation of the values of

$$\frac{\partial \rho(\xi)}{\partial u_j}.$$

This is, the derivative of the total fiber flow over a fixed point ξ with respect to the control parameter u_j. This value will depend on the kind of parameter we consider.

If the parameter u_j represents the translation s_j^x of the shift j and the rest of the parameters are fixed, the density functions ρ_0, ρ_1 are constant, and the expression for the partial derivative can be obtained from,

$$\rho(x,y) = \sum_{k=1}^{N_r} \rho_1(x+s_k^x, y+s_k^y). \tag{3.28}$$

The shift has only effect on the translation in the direction of the $x-$axis, and only in the term j of the sum. This means that,

$$\frac{\partial \rho(x,y)}{\partial u_j} = \frac{\partial}{\partial u_j}\rho_1(x+u_j, y+s_j^y). \tag{3.29}$$

If u_j is a parameter from a discretization of the radial distribution, according to section 3.4.1. We know that the flow $\rho(\xi)$ over $\xi=(x,y)$ is given by the same expression 3.28 where,

$$\rho_0(x,y) = \sum_{i=-\infty}^{+\infty}\sum_j \beta_j \pi_j(r_i/R_s)\cdot \|v_p(r_i,t_i)\|^{-1} \tag{3.30}$$

$$\rho_1(x,y) = \sum_{i=-\infty}^{+\infty} \rho_0(x, y + iL_y) \tag{3.31}$$

As we see, the flow is linear with respect to β_j. Therefore, the derivative is represented as the flow over ξ considering π_j as the radial distribution. In other words,

$$\frac{\partial\rho(x,y)}{\partial u_j} = \rho^{\pi_j}(x,y) \tag{3.32}$$

where $\rho^{\pi_j}(x,y)$ is the density function of a process with the same parameters but taking π_j as the radial distribution.

From all this, we are in the position to obtain the gradient of our function

$$\nabla\mathcal{F} := (\frac{\partial\rho(\xi)}{\partial u_1}, ..., \frac{\partial\rho(\xi)}{\partial u_n}),$$

if we consider the discretization of the radial distribution and the shifts as parameters.

As a corollary of all this analysis we get.

Corollary 1. *If in our model the density function ρ_0 is differentiable, the function $\mathcal{F} : \mathbb{X} \to \mathbb{Y}$ relating the control parameters with the criterion is also differentiable.*

The condition number can be estimated with the norm of the gradient $\|\nabla\mathcal{F}\|$. But there is still something to consider, and it is again the selection of the norm to be used. And the reason is that different components of the gradient vector are related with different kind of parameters. One part with the shifts and other with the flow given by the radial distribution. Such quantities can not be mixed freely.

There are two possibilities. Either we consider two different sensitivities, one with respect to the radial distribution and a second one with respect to the shifts, or we introduce a way to compare "apples with oranges" to mix these to values. For our process I consider the two sensitivities and a "unified" sensitivity by a reverse process.

I consider a perturbation in the shifts equivalent to a perturbation in the radial distribution if both generate a similar perturbation in the output.

This is some sort of averaged sensitivity. But we should not forget that the approach is artificially designed to compare both sensitivities.

3.5.3 Comparing Analytic vs Stochastic Sensitivity

At this point, I will present a comparison of the two approaches for the sensitivity, the analytic and the stochastic. Figure 3.19 shows the correlation between the condition number computed with a Taylor approximation in a discrete domain and the variance from a Monte Carlo simulation. There is no sense in trying to compare such values quantitatively, we can nevertheless compare them qualitatively. They are computed using different domains. The Condition number was estimated using perturbations of 10 degrees in the shifts and a 10% deviation from the mean flow. For the Monte Carlo simulation the shifts where considered a von Mises distribution of variance close to 3 degrees, and the flow was considered a gamma distribution in such a way that three times the variance were the equivalent of the perturbation in the analytical approach. 20 simulation were realized for each point. All this to try to make the number comparable or at least of the same order of magnitude.

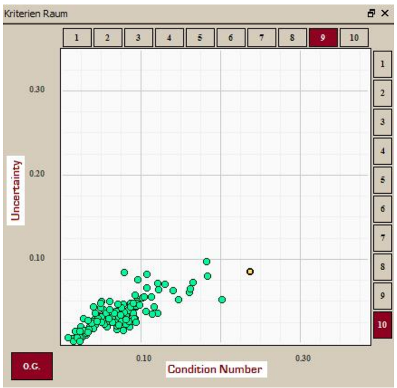


Figure 3.19: Correlation between Stochastic and Analytic sensitivities.

3.6 Optimization of the Process

It was remarked in chapter 2, that there are two important quality attributes to be considered in the production process. The expected level of defects in the fleece and the stability of the process.

The main criterion giving us answers about the expected quality of the final fleece is the cloudiness or homogeneity of the density function describing it. We also have other criteria to measure the specific appearance of light or dark clouds and to detect the presence of stripes in the fleece.

For the stability of the process, we have used the concept of sensitivity of a model and; by describing our model as a function $\mathcal{F} : \mathbb{X} \rightarrow \mathbb{Y}$, we had the possibility of use the condition number to quantify the sensitivity.

As mentioned these criteria or quality attributes for our process happen to be in conflict. Whenever conflicting criteria appeared in an optimization problem, the finding of an "optimal" solution becomes an non trivial task. In the next chapter, we will explore the mathematical theory required to cope with the problem of optimizing a model with conflicting criteria or objectives. This is, the theory of *multicriteria optimization.*

Chapter 4

Optimization with Multiple Criteria

Introduction

In this chapter I will face the interesting problem of optimization with multiple criteria. The theory of multicriteria optimization has a long history and several good publications exist in the area. For the purpose of this project I will explain only the required concepts for my particular application. The interested reader can find much more information on general multicriteria problems, for example in the books of Ehrgott [15] or Miettinen [32] among many others.

In section 4.1 I will introduce some basic concepts and notation related with general multicriteria optimization. The most important is the concept of optimality or efficiency in presence of more than a single criterion. We will see that the solution of multicriteria problems is usually considered a full set, the Pareto set. A relative new approach to approximate successive points in the Pareto set use of continuation methods. Those will be presented in section 4.2. For differentiable problems the Pareto set is a manifold, and techniques from algebra are used to compute points in this manifold [21, 45]. I will also propose an alternative formulation using derivative free methods to avoid the requirement of differentiability in the method. The Pareto set will be approximated using a discrete set of points and interpolation will be used between those points. The estimation of errors in the interpolation will be presented in section 4.3. Finally section 4.4 will treat the problem of approximating and exploring the Pareto set and will motivate the inclusion of the following chapter on decision making.

4.1 Multicriteria Concepts

Consider a function $f : \mathbb{R}^n \to \mathbb{R}^m$, and $R \subset \mathbb{R}^n$. The multicriteria problem is described as,

$$\min_{x \in R} f(x) = (f_1(x), ..., f_m(x)). \tag{4.1}$$

In our problem the set R of constraints or restrictions will be defined by inequalities,

$$R := \{x \in \mathbb{R}^k | g_j(x) \leq 0, j = 1, ..., k\}.$$

Each of the function components $f_i(x)$ represents a goal, criteria or objective function which is to be optimized (in this chapter we assume the functions are to be minimized). We will be looking for a point $x \in R$ such that $f_i(x)$ is minimum for all $i \in \{1, ..., m\}$, but in general there exist no x satisfying,

$$f_i(x) \leq f_i(x'), \forall x' \in R, \forall i \in \{1, ..., m\}.$$

and usually it is required to introduce some special concepts.

The domain space $\mathbb{R}^n$ of f is called *decision space*, an element $x \in \mathbb{R}^n$ is called *decision point*, *decision vector* or simply *decision.* The image space $\mathbb{R}^m$ is called *criteria space*, an element $y = f(x) \in \mathbb{R}^m$ is a *criteria point* (*criteria vector*). In the literature the term *objective* is sometimes preferred instead of criteria. In fact terms *multiobjective* and *multicriteria* are used in almost equal numbers in the community. The set $R \in \mathbb{R}^n$ is called the *feasible set* and points $x \in R$ are called *feasible decisions.* Points $x \notin R$ are called *unfeasible.*

The multicriteria problem seeks for the minimum among a set of vectors. In order to have some sort of "minimum", it is required to define a partial order in the space $\mathbb{R}^m$ where those vectors live. When the components of the vectors represents values of goals to be optimized, a natural way of defining this partial order, is the following. We declare y "better" than y' when each component of y is not worst than the corresponding component of y', and at least one component of y is better than the corresponding component of y'. If this is the case, we say that y dominates y'.

Definition 16. *Given $y, y' \in \mathbb{R}^m$, $y = (y_1, ..., y_m), y' = (y'_1, ..., y'_m)$. We say y* dominates *y', or y'* is dominated by *y, if $y_i \leq y'_i$ for all $i \in \{1, ..., m\}$, and $y_j < y'_j$ for some $j \in \{1, ..., m\}$.*

This is the concept of *Pareto Dominance* [15, 32]. It is common to use the symbol "$\preceq$" to denote dominance, i.e. $y \preceq y'$ means that y dominates y'. In the design space, we speak about efficiency.

Definition 17. *A feasible decision x is called* efficient *or* Pareto optimal *if $\nexists x' \in R$ such that $f(x') \preceq f(x)$. In this case the criteria point $f(x)$ is called* nondominated*.*

The set of all efficient decisions is called *efficient set* and the image of the efficient set is called *nondominated set*. In the literature, the concept of *Pareto set* is sometimes used instead of nondominated set but in other places is used instead of efficient set. I may use the term *Pareto set* or *Pareto frontier* for the nondominated set in criteria space.

Without any additional information, it is not possible to distinguish between two efficient solutions, in the sense that none of them is preferred over the other. With this in mind, we consider each efficient/nondominated point as a solution for problem 4.1. Having several goal functions makes clear that, usually there are more than one single efficient point. An alternative point of view is considering the whole set of efficient points as the solution of the problem. Our task is then to approximate this set. In general, for differentiable problems, the efficient set is a manifold of dimension $m-1$ in $\mathbb{R}^m$, see

[21].

The concepts of dominance and efficiency can also be defined locally.

Definition 18. *A feasible decision x is called* locally efficient *or* locally Pareto optimal *if there is an open set $\mathcal{A} \subset \mathbb{R}^n, x \in \mathcal{A} \cap R$ such that $\nexists x' \in \mathcal{A} \cap R$, $f_i(x') \leq f_i(x)$ for all $i = 1, ..., m$ and $f_j(x') < f_j(x)$ for some $j \in \{1, ..., m\}$. In this case the criteria point $f(x)$ is called* locally nondominated.

4.1.1 Parametrization of the Pareto Set

The concepts of dominance and efficiency can be used to define algorithms and find points in the Pareto set. Nevertheless, if a local control and exploration of the set is required we will need to use a modified approach. One of the typical ways of doing it is the use of aggregated functions.

An *aggregated function* is simply a function ϕ that maps the criteria space $\mathbb{R}^m$ into the reals, i.e. $\phi : \mathbb{R}^m \to \mathbb{R}$. Such a function will be useful, when its minimum corresponds with an efficient point.

A single aggregated function is able to obtain one efficient point. In order to obtain all of them, we need to use several functions. For this, aggregated functions are normally used as a parameterized family of functions. By changing the parameters it is possible to obtain different points in the efficient set. The most common aggregated function is the *weighted sum*,

$$w(\alpha, x) := \sum_{i=1}^{m} \alpha_i f_i(x) = \alpha \cdot f(x) \tag{4.2}$$

which corresponds to the inner product of the criteria vector with a vector $\alpha \in \mathbb{R}^m$ of parameters. This means that the weighted sum returns a scalar value in relation with the distance of the criteria point to a given subspace generated by the equation $\alpha \cdot y = 0$ in $\mathbb{R}^m$. From vector algebra, we know that α is normal to the subspace and that the distance from the point $f(x) \in \mathbb{R}^m$ to the subspace is given by the expression,

$$\frac{\alpha \cdot f(x)}{\|\alpha\|}.$$

By choosing normalized weights vectors the weighted sum represents precisely that distance.

In some situations we will consider the functions

$$w_\alpha(x) := w(\alpha, x)$$

with fixed parameters vector α or the function

$$w^x(\alpha) := w(\alpha, x)$$

for a fixed point x.

Lemma 12. *If $x^* \in R$ is a local minimum for the scalar optimization problem*

$$\min_{x \in R} w_\alpha(x), \tag{4.3}$$

then x^* is a locally efficient point for the multicriteria problem 4.1.

A similar result is also true in the case of global optimality.

Lemma 13. *If $x^* \in R$ is a global solution for the scalar optimization problem*

$$\min_{x \in R} w_\alpha(x),$$

then x^* is an efficient point for the multicriteria problem 4.1.

The results are rather intuitive and the proof by contradiction; see [32], is not difficult. For scalar convex problems, local optimality imply global optimality and the same is true for multicriteria problems.

Theorem 4. *If the multicriteria optimization problem is convex, then every locally Pareto optimal solution is a global solution.*

In lemmas 12 and 13 a sufficient condition is given for local and global Pareto optimality. If problem 4.1 is convex, i.e., the set R is convex and the functions f_i are convex, then any efficient point can be obtained solving a minimization problem

$$\min_{x \in R} w_\alpha(x) \tag{4.4}$$

for some parameters vector α.

If we have a non convex restriction set or expect non convex functions for our problem the weighted sum will not be able to obtain all efficient points.

An aggregated function able to obtain all efficient points even for non convex problems is the weighted Tchebycheff function.

The weighted Tchebycheff problem has the following form,

$$\min_{x \in R} \max_{i=1,\ldots,m} \alpha_i (f_i(x) - y_i)$$

where $\alpha \in \mathbb{R}^m$ and $y = (y_1, \ldots, y_m)$ is an ideal criteria vector, $y_i < f_i(x)$ for all $x \in \mathbb{R}^n$. Geometrically the Tchebycheff function uses the distance of the point $f(x)$ to an ideal point y, nevertheless, using the infinity norm. We can rewrite the Tchebycheff problem in the following way,

$$\min_{x \in R} \|\alpha I f(x)\|_\infty,$$

where I is the $m \times m$ identity matrix.

To simplify the notation we will assume that our criteria functions are all non negative. In this way the zero vector is an ideal vector.

Our *weighted Tchebycheff function* is then,

$$\tau(\alpha, x) = \max_{i=1,\ldots,m} \alpha_i f_i(x). \tag{4.5}$$

And the *weighted Tchebycheff problem* becomes,

$$\min_{x \in R} \tau(\alpha, x). \tag{4.6}$$

As before we denote $\tau_\alpha(x) := \tau(\alpha, x)$ and $\tau^x(\alpha) := \tau(\alpha, x)$.

For this aggregated function we have again similar results,

Lemma 14. *If $x^* \in R$ is a local minimum for the problem 4.6 then x^* is locally efficient point for the multicriteria problem 4.1.*

Lemma 15. *If $x^* \in R$ is a global solution for the problem 4.6 then x^* is an efficient point for the multicriteria problem 4.1.*

But now we have an additional result,

Theorem 5. *Let $x^* \in R$ be an efficient decision point. Then there exists a vector α such that x^* is a solution of the Tchebycheff problem*

$$\min_{x \in R} \tau_\alpha(x).$$

It is important to note that the result is valid also for non convex problems. A proof can be found in [32].

4.1.2 KKT Conditions

Assuming differentiability in f, it is possible to unify the criteria functions and the inequality constraints in a single equation. The idea evolves in the well known Karush-Kuhn-Tucker conditions for Pareto optimality.

Theorem 6. *A necessary condition for a point x_0 to be properly efficient is the existence of vectors $\alpha = (\alpha_1, ..., \alpha_k)$ and $\mu = (\mu_1, ..., \mu_m)$, $\alpha_i > 0, \mu_j > 0$ $i < k, j < m$ such that*

$$\sum_{i=1}^{k} \alpha_i \nabla f_i(x_0) + \sum_{j=1}^{m} \mu_j \nabla g_j(x_0) = 0$$
$$\mu_j g_j(x_0) = 0, \quad \forall j \in \{1, ..., m\}$$

This conditions can be written in a single system of equations by including the vectors α and μ. We will write the system in the following way,

$$\mathcal{F}(x, \alpha, \mu) = 0. \qquad (4.7)$$

Solutions of the system 4.7 are candidates to be Pareto optimal points. It is proved in [21] that the set of all these solutions define a differentiable manifold provided the rank of the Jacobian of $\mathcal{F}$ is $n + m + 1$.

The task of computing successive points in the Pareto set becomes the problem of computing neighbor points in a differentiable manifold. This problem is well known in the algebra community and one of the solution strategies commonly used is similar to the idea of the predictor-corrector methods used to approximate solution trajectories in differential equations. That is the spirit of the so called continuation or homotopy methods. See, for example [1],[21].

4.2 Continuation Methods

For differentiable problems where the KKT conditions are valid, we are left then with a system of equations. A procedure to approximate successive points, solutions of the system is the following,

Continuation Step.

1. Assume you have an efficient design x_α, $\mathcal{F}(x_\alpha, \alpha, \mu_\alpha) = 0$.

2. Find a predictor point x^*_α in the subspace tangent to the efficient set. This could be done with the purpose to improve some of the criteria, and there is a relation with a perturbation in the weights vector $\beta = \alpha + \delta$.

3. Using the initial guess x^*_α perform a corrector step (like Newton algorithm) to find a solution of the form $(x_\beta, \beta, \mu_\beta)$ to the system, i.e. $\mathcal{F}(x_\beta, \beta, \mu_\beta) = 0$.

4. Now we have a new efficient decision x_β.

The continuation step can be applied in succession to obtain neighbor points in the Pareto set. The computation of the tangent subspace corresponds to the first order Taylor approximation for $\mathcal{F}$. I will not write here the details of the continuation method, mainly because I will not use it. Nevertheless, the interested reader can look for example [21] for a complete explanation of the method.

But, why not using it?

To begin with, some of our criteria are not differentiable. Remember we have used in many places the minimum and maximum functions. But even when these functions could be approximated by differentiable ones, the principal reason is the heavy use of derivatives involved in the algorithm. The computation of the tangent subspace requires the gradient of the function but even higher order derivatives will be required, specially if a Newton corrector is to be used.

The computation of expressions for derivatives in our particular model is, if at all possible, very cumbersome. For this reason I will propose to stay with the original description for efficient points given in lemmas 12 and 13 and consider a family of minimization problems, instead of a system of equations.

The results obtained in section 4.1.1 can be used to obtain efficient points. The corresponding continuation method would be,

Derivative Free Continuation Step

1. Consider a weights vector $\alpha \in \mathbb{R}^m$ and assume you have found an efficient decision x_α, solution of the scalar optimization problem

$$\min_{x \in R} \phi_\alpha(x).$$

2. Change the weights vector $\beta = \alpha + \delta$ as desired and solve the new problem for the function $\phi_\beta(x)$

3. Now we have a new efficient design x_β.

Depicted in figure 4.1 are two steps of a continuation method. The gradient depicts a two dimensional function (dark color represents high values). The function changes with the parameters. At each step a minimization problem is solved. Point A is the minimum of the function in the first image, B is the minimum of the second function and C the minimum of the third function. En each step, the points can be obtained with a direct search algorithm starting in the previous obtained point.

If the step is small enough, a local search is able to approximate accurately the solution of the new problems based in the old ones. An extra assumption is that the local minima are isolated local minima. Which means that the value of the function is strictly less than any other point in a neighborhood. Without this assumption the dynamics of such process could become very complicated, for example bifurcations or other behaviors could appear. If the

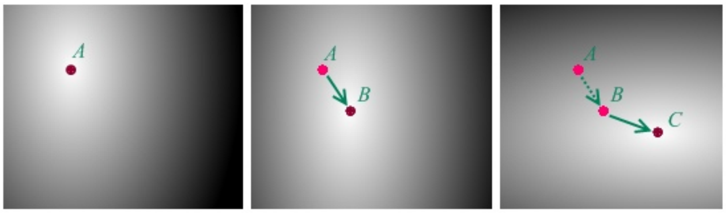


Figure 4.1: Two continuation steps $A \to B \to C$

assumption is false, the continuation method will only obtain one sequence. I do not study the possibility of bifurcations in this dissertation.

There is some freedom in the choose of the aggregated function ϕ_α. For convex problems, it is enough to use the weighted sum w_α but for non convex problems it will be required to use the weighted Tchebycheff function τ_α or other kind of approach instead; like goal programming. In the step 2, there is also no specification on, how the new problem

$$\min_{x \in R} \phi_\beta(x),$$

is to be solved. Derivative free methods to solve this kind of scalar optimization problems will be presented in chapter 5.

Now we analyze the approximation errors based in the information we have related to our function and the size of the step we are considering.

4.3 Discrete Approximation and Interpolation Error

The continuation methods are used to find a sequence of efficient points. We would expect the intermediate efficient points to be approximated by this discrete set. As in any problem where a continuous set is approximated by a discrete one, an important analysis to be performed is the estimation for the interpolation errors between the points in the approximation.

Consider a Pareto efficient point x_α, solution of a problem with weights vector α. Select a new weights vector β where $\beta = \alpha + \delta$. Using the continuation step, assume we obtain a neighbor Pareto point x_β. A natural way of measuring the interpolation error is as follows.

Definition 19. *Given two decisions x_α and x_β in the efficient set. We define the interpolation error between them as,*

$$\max_{0<\lambda<1} \|y_\lambda - f(x_\lambda)\|$$

where x_λ is the solution of

$$\min_{x \in V_\varepsilon(x_\alpha)} \phi_{(1-\lambda)\alpha+\lambda\beta}(x), \lambda \in [0,1],$$

and $y_\lambda := f((1-\lambda)x_\alpha + \lambda x_\beta))$.

We will need the following lemma which relates the interpolation error with the Lipschitz constant.

Lemma 16. *Let $F : D \subset \mathbb{R}^N \to \mathbb{R}^M$ be Lipschitz continuous with constant L. For any two points $x_0, x_1 \in D$ let $\delta = \|x_1 - x_0\|$ then*

$$\|y_\lambda - F(x_\lambda)\| < L\delta,$$

where, for $0 < \lambda < 1$, we have

$$y_\lambda = (1-\lambda)F(x_0) + \lambda F(x_1), \quad x_\lambda = (1-\lambda)x_0 + \lambda x_1.$$

Proof.

$$\begin{aligned}
\|y_\lambda - F(x_\lambda)\| &= \|(1-\lambda)F(x_0) + \lambda F(x_1) - F(x_\lambda)\| \\
&= \|(1-\lambda)F(x_0) - (1-\lambda)F(x_\lambda) + \lambda F(x_1) - \lambda F(x_\lambda)\| \\
&= \|(1-\lambda)(F(x_0) - F(x_\lambda)) + \lambda(F(x_1) - F(x_\lambda))\| \\
&\leq L(1-\lambda)\|x_0 - x_\lambda\| + L\lambda\|x_1 - x_\lambda\| \\
&\leq L(1-\lambda)\|x_0 - x_1\| + L\lambda\|x_1 - x_0\| \\
&= L\delta
\end{aligned}$$

□

Lemma 17. *If f is Lipschitz continuous, the aggregated function*

$$w_\alpha(x) = \alpha \cdot f(x)$$

is Lipschitz continuous.

Proof. The weighted sum $w_\alpha(x)$ is the composition of f with the inner product in $\mathbb{R}^m$. The inner product is more than Lipschitz, it is even differentiable. Because f is Lipschitz, the result follows. □

The same result follows for the Tchebycheff aggregating function.

Lemma 18. *If f is Lipschitz continuous, the aggregated function*

$$\tau_\alpha(x) = \max\{\alpha_i f_i(x)\}$$

is Lipschitz continuous.

Proof. The Tchebycheff function $\tau_\alpha(x)$ is the composition of f with the *max* function. The *max* function is not differentiable, but Lipschitz. In any case the composition is Lipschitz again. □

The following two lemmas follow directly from the Lipschitz continuity of the inner product and the *maximum* function.

Lemma 19. *The function $w^{x_0}(\alpha) := w(\alpha, x_0)$ is Lipschitz continuous.*

Lemma 20. *The function $\tau^{x_0}(\alpha) := \tau(\alpha, x_0)$ is Lipschitz continuous.*

Theorem 7. *Consider the map*

$$\begin{array}{rccc} \pi: & \mathbb{R}^m & \rightarrow & \mathbb{R}^n \\ & \alpha & \mapsto & x_\alpha \end{array} \tag{4.8}$$

where x_α is a strict local minimum of the aggregating function $\phi_\alpha(x)$ for the multicriteria problem 4.1. i.e.,

$$x_\alpha := \operatorname*{argmin}_{x \in R} \phi_\alpha(x).$$

If f is Lipschitz, then π is locally Lipschitz continuous in x_α.

Proof. Consider an ϵ neighborhood of x_α, $\mathcal{B}_\epsilon(x_\alpha) := \{x \in R | \|x - x_\alpha\| < \epsilon\}$ such that

$$\phi_\alpha(x_\alpha) < \phi_\alpha(x), \quad \forall x \in \partial\mathcal{B}_\epsilon(x_\alpha)$$

From lemma 17 or 18 we have,

$$|\phi_\alpha(x_\alpha) - \phi_\alpha(x)| < M_1\|x_\alpha - x\|.$$

and from lemma 19 or 20,

$$|\phi_\alpha(x) - \phi_{\alpha+\delta}(x)| < M_2\|\delta\|,$$

for any $\delta \in \mathbb{R}^m$. The inequality could be written as

$$\phi_\alpha(x) - M_2\|\delta\| < \phi_{\alpha+\delta}(x) < \phi_\alpha(x) + M_2\|\delta\|. \tag{4.9}$$

Consider a vector $\delta \in \mathbb{R}^m$, and $M := \max\{M_1, M_2\}$ such that,

$$2M\|\delta\| < \min_{x \in \partial\mathcal{B}_\epsilon(x_\alpha)} |\phi_\alpha(x) - \phi_\alpha(x_\alpha)| \tag{4.10}$$

From 4.9 and 4.10 we obtain the inequalities,

$$\begin{aligned} \phi_{\alpha+\delta}(x_\alpha) &< \phi_\alpha(x_\alpha) + M\|\delta\| \\ &< \phi_\alpha(x) - M\|\delta\| \\ &< \phi_{\alpha+\delta}(x). \end{aligned} \tag{4.11}$$

The minimum of the function $\phi_{\alpha+\delta}$ is not attained in the boundary of $\mathcal{B}_\epsilon(x_\alpha)$. From this and the last inequality we know that the aggregating function $\phi_{\alpha+\delta}(x)$ attains a local minimum in the interior of $\mathcal{B}_\epsilon(x_\alpha)$. By choosing $M^* = \epsilon/\|\delta\|$ we have finally, $\|x_\alpha - x_{\alpha+\delta}\| < \epsilon = M^*\|\delta\|$. □

The map π and the vector function f are Lipschitz continuous, and so is its composition. Using lemma 16 we can estimate the interpolation error for points obtained from the continuation method.

If the interpolation error for the points obtained in the continuation method is higher than we would tolerate, we should consider the possibility of computing further efficient points between the already obtained. This can be done in a straightforward way by simply reducing the step size δ on the continuation method.

Consider two neighbor efficient points $x_\alpha \neq x_\beta$, $\beta = \alpha + \delta$ for the multicriteria problem 4.1, let $x_\lambda := \lambda x_\alpha + (1 - \lambda) x_\beta$,

We can obtain a new efficient points between x_α and x_β by solving,

$$\min_{x \in R} \phi_{\alpha + \lambda \delta}(x),$$

for each $\lambda \in [0, 1]$.

4.4 Pareto Frontier Approximation and Exploration

The continuation method is able to obtain further points in the Pareto frontier as long as the points belong to a connected component of the frontier. In the general case, the Pareto frontier is not composed of a single component. For convex multicriteria problems over a convex constraint domain R the Pareto frontier and the efficient set in the design space are connected, see [15].

Theorem 8. *For problem 4.1. If R is convex and all functions $f_k, k \in \{1, ..., m\}$ are convex, the Pareto frontier and the efficient set are connected.*

This theorem implies that a single component exist for convex problems. For non convex problems, several disconnected components may exist. Making impossible to jump from one component to the other using local strategies.

Because the continuation method can be used to explore connected components of the frontier, what it is required is to obtain at least, one point in each component. Now we require what is usually offered for any multicriteria optimization method, i.e. a discrete set of non dominated solutions. In most of the literature, multicriteria methods are required to obtain an uniform spread of points in the Pareto frontier. For our algorithm, at least a single point should be obtained per connected component. This requires a good exploration of the design space.

For the problem with two criteria, the strategy is to find the local minima of one of the functions and perform continuation steps, giving more weight to the second function. In this way the full connected components can be approximated. For several criteria, the strategy would be to find non dominated points using several different fixed aggregated functions, and allow a person to choose the direction he want to explore, for the continuation step. I will explore different ways of navigating the Pareto frontier in chapter 6.

In any case, what we need is an efficient method for solving scalar, non linear, non differentiable problems. We will explore different possible methods for scalar optimization in the next chapter.

Chapter 5

Scalar Optimization

Introduction

Up to now we have described a model for the web formation in our spunbond processes and the different criteria to be considered. This chapter will deal with methods for scalar optimization, in other words optimization alternatives where a single from those criteria has to be optimized. We have two reasons to consider such a topic.

First, we have seen that scalar optimization is a required step in the multicriteria context. We need then decide which scalar optimization method to use together with the continuation method in the approximation of the Pareto frontier. The second reason is the importance of obtaining the optimal fleece in terms of the homogeneity criterion. When implementing (in chapter 7) the methods described in this chapter, we will see that almost ideal fleeces can be obtained, at least when we consider only the expected homogeneity and neglect the sensitivity.

A high quality fabric is one where the deposited fibers are homogeneously distributed along the band. In chapter 3 we have seen some possible approaches to quantify this homogeneity. Even when we have several different criteria only to measure the homogeneity of the fleece; or the presence of clouds, the control parameters will allow us to obtain almost perfectly homogeneous density functions. Suggesting a correlation between all these criteria functions. Therefore, in this chapter, I consider only a single criterion at a time. I will present different ways to deal with this scalar optimization problems, compare them and choose the best to use it in the multicriteria framework.

In the spunbond process the production machine is described using several parameters. For the optimization we will consider most of those as fixed, and concentrate in the two kind of parameters we already mentioned. First the radial distribution $p : \mathbb{R} \to \mathbb{R}_0^+$, and second, the vector of shifts $\varphi \in \mathbb{R}^{N_r}$ describing the synchronization of the spinnerets. The cloudiness; as any other criterion, is then a function of this parameters.

In section 5.1 the problem of finding the optimal radial distribution given a set of fixed shifts is presented. Assuming a piecewise linear radial distribution, the problem becomes a classical linear program (LP) and can be solved almost instantaneously using standard solvers. The mechanical modifications required for altering the radial distribution are more sophisticated than those required to alter the synchronization of the spinnerets. In section 5.2 we consider the problem of finding an optimal shift synchronization assuming a fixed radial distribution function. We will see that a mixed integer program (MIP) can be obtained, but that the size of the problems grows beyond the limits and several hours are required to obtain acceptable solutions. In the same section a different idea is considered. The periodic nature of the functions suggest a Fourier approximation. In section 5.3 all the parameters are taken into account and a complete model is described including the radial distribution and the shifts. As in 5.2 we will obtain a mixed integer program. The time required for solving the problem ban the possibility of allowing fast optimization. We will be forced to propose other methods, if we want to speed up the finding of solutions. Finally in section 5.4 I will propose and present several heuristic methods for scalar optimization to cope with the complete model. The attempt of finding a global solution in a non-convex problem, force us to divide the strategy in two. A global exploration search, and a local refinement. For the global exploration, an evolutionary algorithm is proposed. And for the local search, two algorithms were used, a Pattern Search and the Nelder-Mead algorithm.

5.1 Optimal Radial Distribution

Consider a spunbond process as described in section 3.2. With N_r rows of spinnerets and fixed shifts $\{\varphi_1, ..., \varphi_{N_r}\}$. Assume we have to find a radial distribution

$$p : [0,1] \to \mathbb{R}_0^+$$

in order to minimize the presence of clouds (dark or light) in the final web.

The criteria function to be considered in this section is the light cloudiness It is also possible to consider the dark cloudiness or other criteria.

Definition 20. *The objective function is the* light cloudiness *(definition 7),*

$$\mathcal{F}(p, \varphi) := \kappa_{min}.$$

The problem of optimizing the light cloudiness with fixed shifts can be written as,

Problem 1. *Given* $\varphi \in \mathbb{R}^{N_r}$

$$\min_{p \in \mathcal{P}} \mathcal{F}(p, \varphi).$$

Where $\mathcal{P}$ *is a given function set.*

As mentioned in the previous chapter, if we expect to solve such a problem numerically, the first thing to do is to restrict to a set of functions with a finite base. We consider continuous picewise linear functions

$$p_\beta(x) = \sum_{i=1}^{N_p} \beta_i \pi_i(x), \tag{5.1}$$

where π_i are basis functions, the popular "hat functions" with support on the interval $[0,1]$ defined in section 3.4.1. Each radial distribution is then characterized by a vector of coefficients $\beta = (\beta_1, ...\beta_{N_p})$.

Assuming this, the problem is transformed.

Problem 2.

$$\min_{\beta \in \mathbb{R}_+^{N_p}} \mathcal{F}(p_\beta, \varphi).$$

We should not forget that only positive coefficients can be used, because the radial distribution reflects the material deposited, and the material is never removed.

In the previous chapter, the relation between the radial distribution p and the density function ρ was presented. Now, we will see the relation between the vector β and the density functions.

The density function for a single spinneret ρ_0 (3.11) was defined as,

$$\rho_0(x,y) = \sum_{j=-\infty}^{+\infty} p(r_j/R_s)\cdot \|v^\perp(r_j,t_j)\|. \tag{5.2}$$

Using 5.1 we get,

$$\rho_0(x,y) = \sum_{j=-\infty}^{+\infty} \sum_{i=1}^{N_p} \beta_i \pi_i(r_j/R_s)\cdot \|v^\perp(r_j,t_j)\|.$$

Both sums are in fact finite, we can exchange them and take the coefficients out of the first sum. These steps give us a new expression for ρ_0.

$$\rho_0(x,y) = \sum_{i=1}^{N_p} \beta_i \sum_{j=-\infty}^{+\infty} \pi_i(r_j/R_s)\cdot \|v^\perp(r_j,t_j)\|.$$

Or shortly.

$$\rho_0(x,y) = \sum_{i=1}^{N_p} \beta_i \rho_0^{\pi_i}(x,y).$$

Where $\rho_0^{\pi_i}$ is the density function generated by a single spinneret considering a radial distribution π_i.

Analogously, and with similar arguments we can obtain and simplify expressions for ρ_1 and ρ.

From (3.12),

$$\rho_1(x,y) = \sum_{j=-\infty}^{+\infty} \rho_0(x,y+jL_y),$$

$$\rho_1(x,y) = \sum_{j=-\infty}^{+\infty} \sum_{i=1}^{N_p} \beta_i \rho_0^{\pi_i}(x,y+jL_y),$$

$$\rho_1(x,y) = \sum_{i=1}^{N_p} \beta_i \sum_{j=-\infty}^{+\infty} \rho_0^{\pi_i}(x, y + jL_y),$$

$$\rho_1(x,y) = \sum_{i=1}^{N_p} \beta_i \rho_1^{\pi_i}(x,y).$$

And from (3.14),

$$\rho(x,y) = \sum_{k=1}^{N_r} \rho_1(x + s_k^x, y + s_k^y),$$

$$\rho(x,y) = \sum_{k=1}^{N_r} \sum_{i=1}^{N_p} \beta_i \rho_1^{\pi_i}(x + s_k^x, y + s_k^y),$$

$$\rho(x,y) = \sum_{i=1}^{N_p} \beta_i \sum_{k=1}^{N_r} \rho_1^{\pi_i}(x + s_k^x, y + s_k^y),$$

$$\rho(x,y) = \sum_{i=1}^{N_p} \beta_i \rho^{\pi_i}(x,y). \tag{5.3}$$

Therefore, we obtain the following result.

Lemma 21. *Consider a spunbond process with a radial distribution of the form,*

$$p_\beta(x) = \sum_{i=1}^{N_p} \beta_i \pi_i(x).$$

Then, the fleece density function is given by,

$$\rho(x,y) = \sum_{i=1}^{N_p} \beta_i \rho^{\pi_i}(x,y)$$

where ρ^{π_i} is the fleece density generated by considering the basis function π_i as a radial distribution.

As we can see, if the radial distribution p is a linear combination of the basis functions π_i, then the corresponding density function ρ can be written as an analogous linear combination of the density functions ρ^{π_i} generated by considering π_i as radial distribution in the model.

Using this lemma, and by fixing the number of basis functions, it is possible to compute the density function for any radial distribution ρ_β very quickly provided we pre-compute the density functions for a fixed set of basis functions. This will become one of the approaches to allow faster computation and real time navigation.

5.1.1 LP Formulation for the Radial Distribution Optimization

The density functions are periodic and can be defined in a rectangular domain given by the periods L_x, L_y. Consider a discretization of this domain using N_x points in the $x-$direction and N_y points in the $y-$direction. To compact the notation, let us define the index sets,

$$I = \{1, ..., N_p\}, K = \{1, ..., N_x\}, L = \{1, ..., N_y\}.$$

Relation (5.3) is transformed into,

$$\alpha_{k,l} = \sum_{i \in I} \beta_i \cdot A_{i,k,l}, \quad k \in K, l \in L$$

where $A_{i,k,l}$ can be described as the material deposited due to the basis function π_i over a neighborhood to the point (k, l) in the discretization. In that case $\alpha_{k,l}$ will represent the total flow in the same spot due to the radial distribution p_β.

We consider now the problem of maximizing the minimum value of $\alpha_{k,l}$, i.e. the criteria minimum measuring the presence of light clouds.

$$\max \min_{k \in K, l \in L} \alpha_{k,l}$$

Such expression is equivalent to,

$$\max \lambda, \text{ where } \alpha_{k,l} \geq \lambda, \forall k \in K, \forall l \in L.$$

Now we can define our linear program [17] for the optimization of the radial distribution.

Problem 3. *Given the input data $A_{j,k,l}$ and the real variables λ, $\alpha_{k,l}$ and β_i, for $i \in I, k \in K$ and $l \in L$*

$$\max \quad \lambda \tag{5.4}$$

subject to

$$\alpha_{k,l} \geq \lambda, \qquad \forall k \in K, \forall l \in L \tag{5.5}$$

$$\alpha_{k,l} = \sum_{i \in I} \beta_i \cdot A_{i,k,l}, \qquad \forall k \in K, \forall l \in L \tag{5.6}$$

$$\sum_{i \in I} \beta_i = 1 \tag{5.7}$$

$$\beta_i \geq 0 \qquad \forall i \in I \tag{5.8}$$

Lower values of $\alpha_{k,l}$ are signal of possible light clouds in the fleece. Conditions (5.4, 5.5) imply that λ should be the minimum $\alpha_{k,l}$. Restriction (5.7) represents the normalization of the radial distribution, and (5.8) means that no negative flow of material is allowed. More references on linear programming can be found in [6], [42], [39] among others.

The periods of the domain have dimensions of some hundreds of centimeters. The cloud defects are usually significative if measuring at least between 3 or 5 centimeters. This means that the number of discretization points in the domain; the $\alpha_{k,l}$ are around 10000. On the other hand, the radial distribution needs no more than 10 to 15 discretization points. In this way, the linear problem has a few thousands of variables. Ad-hoc solvers, like ILOG-CPLEX [22] are able to cope with this problem in few seconds.

5.2 Optimal Shifts

As in the previous section, consider a spunbond process with N_r rows of spinnerets and a fixed radial distribution p. We consider now the problem of finding an optimal vector of shifts $\varphi = (\varphi_1, ..., \varphi_{N_r})$ such that our criterion for homogeneity is minimized.

Problem 4. *Given* $p : [0,1] \to \mathbb{R}^+$,

$$\min_{\varphi \in \mathbb{R}^{N_r}} \mathcal{F}(p, \varphi).$$

Each shift is an angle, in this case the search space is $\mathbb{S}^{N_r}$ where $\mathbb{S}$ corresponds to the unit circle. We will need to be careful in the implementation of algorithms when computing with angles. But we will come back to this later.

As seen on section 3.2.3, the vector of shifts define a set of translations of the density function ρ_1. The sum of these translated copies of ρ_1 define the final density function ρ in the following way.

$$\rho(\xi) = \sum_{j=1}^{N_r} \rho_1(\xi + s_j) \tag{5.9}$$

Where $\xi = (x, y)$ is the point in the domain and $s_k = (s_k^x, s_k^y)$ are the translation vectors. Relations (3.15, 3.16) allow us to change between translations and shifts easily. The shifts will be used, most of the times for the optimization algorithms, but the simulations will always require the computation of the translations.

5.2.1 Mixed Integer Program Formulation

In order to obtain a known optimization problem we will make some discretizations. As before, we discretize the domain of the density function using N_x points in $x-$direction and N_y points in $y-$direction and the index sets $K = \{1, ..., N_x\}, L = \{1, ..., N_y\}$.

The next thing to do is to consider only a finite number of possible translations. Typically we can consider only shifts of the form

$$\varphi_j = \sigma^s := \frac{2\pi s}{N_s},\ s \in \mathbb{N}$$

or its corresponding translations. In this way, we restrict the shifts to N_s different values. For $N_s = 4$ the possible values are $\{0°, 90°, 180°, 270°\}$. We define other two index sets for the number of discretized shifts and rows,

$S = \{1, ..., N_s\}, J = \{1, ..., N_r\}$ and the binary variable,

$$\chi_{j,s} = \begin{cases} 1 & \text{if the shift } \sigma^s \text{ is applied in row } j \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

Observe that, in this case, the shift applied in row j can be written as,

$$\varphi_j = \sum_{s \in S} \sigma^s \chi_{j,s}, \forall j = 1, ..., N_r.$$

Relation (5.9) is transformed into,

$$\alpha_{k,l} = \sum_{i \in I} \sum_{j \in J} A_{j,k,l,s} \cdot \chi_{j,s}, \quad k \in K, l \in L,$$

where, in a similar way as before, $A_{j,k,l,s}$ represents the material deposited over the spot (k, l) assuming that the shift of the $j-$th row is σ^s. Remember that the radial distribution is considered fixed.

The problem is again the maximization of the minimum value of $\alpha_{k,l}$.

$$\max \min_{k \in K, l \in L} \alpha_{k,l}$$

Also we obtain the problem.

Problem 5. *Given the input data $A_{j,k,l,s}$ the real variables λ, $\alpha_{k,l}$ and the binary variables $\chi_{j,s}$, with $j \in J, k \in K, l \in L$ and $s \in S$.*

$$\max \quad \lambda \tag{5.11}$$

subject to

$$\alpha_{k,l} \geq \lambda, \qquad \forall k \in K, \forall l \in L \tag{5.12}$$

$$\alpha_{k,l} = \sum_{i \in I} \sum_{j \in J} A_{j,k,l,s} \cdot \chi_{j,s}, \qquad \forall k \in K, \forall l \in L \tag{5.13}$$

$$\sum_{s \in S} \chi_{j,s} = 1, \qquad \forall j \in J \tag{5.14}$$

$$\chi_{j,s} \geq 0, \qquad \forall j \in J, s \in S \tag{5.15}$$

As in the linear program, expressions (5.11,5.12) define λ as the minimum value for $\alpha_{k,l}$. Condition (5.14) reflects the fact that only a single shift value is to be assigned to a single row, and condition (5.15) is required to complete the equivalence with the definition (5.10) of $\chi_{j,s}$.

Due to the appearance of integer and real variables, problem 5 is called *Mixed Integer Linear Program* or MILP for short. Ad-hoc software exists to cope with MILP problems too. But MILP are considerably harder to solve completely compared with LP. Even for a relative coarse discretization, standard software like CPLEX will run for some days before obtaining an optimal solution and gap of less than 10% . This could be acceptable, if considering a scalar optimization problem. But if we are interested on multicriteria optimization, the procedure to obtain a solution has to be repeated several times, and the algorithm becomes not useful in practice. References for MILP are [37], [53].

5.2.2 Fourier Approximation

For a moment we will ignore the actual spunbond process, and will consider only the abstract expression (5.9).

$$\rho(\xi) = \sum_{j=1}^{n} \rho_1(\xi + s_j) \tag{5.16}$$

The abstract problem consist in finding n translated copies of a given periodic function f in order to obtain a desired function h. In general, the restrictions will not permit us to obtain h exactly, so we have to be satisfied with a good approximation of it.

Problem 6. *Given the periodic functions $f, h : \mathbb{R}^2 \to \mathbb{R}$ and $n \in \mathbb{N}$,*

$$\min_{s \in \mathbb{R}^n} \|g - h\|.$$

Where the function $g : \mathbb{R}^2 \to \mathbb{R}$ is defined as,

$$g(\xi) := \sum_{j=1}^{n} f(\xi + s_j).$$

In our application the goal function h is constant, equal to n times the average of the function f. Let $R := [0, L_x] \times [0, L_y]$ be the periodic domain of the functions f, h.

$$h(\xi) \equiv H \in \mathbb{R}, H := n\overline{f} = \frac{n}{R} \int_R f$$

In practice, people in the industry consider two projected functions, like the ones defined in section 3.3.2 to measure the presence of stripes. One projected in the direction of the moving belt, and the second orthogonal to the first. They expect that, optimization with respect to this two directions will result in a high quality fleece for the whole domain.

The projected function $f_x : \mathbb{R} \to \mathbb{R}$ over the $x-$axis is defined as,

$$f_x(x) := \frac{1}{L_y} \int_0^{L_y} f(x, y) dy.$$

Analogously, the projected function f_y over the $y-$axis is defined as,

$$f_y(y) := \frac{1}{L_x} \int_0^{L_x} f(x, y) dx.$$

The one dimensional version of problem (6) is,

Problem 7. *Given the periodic functions $f, h : \mathbb{R} \to \mathbb{R}$ and $n \in \mathbb{N}$,*

$$\min_{s \in \mathbb{R}^n} \|g - h\|.$$

Where the function $g : \mathbb{R} \to \mathbb{R}$ is defined as,

$$g(t) := \sum_{j=1}^{n} f(t + s_j).$$

The periodic nature of the functions suggest an alternative approach. Let us consider the Fourier series expansion of f_x.

$$\begin{aligned} f_x(t) &= \frac{a_0}{2} + \sum_{k=1}^{\infty}[a_k \cos(\omega_j t) + b_k \sin(\omega_k t)] \\ \text{where, for } k \in \mathbb{N}_+ & \\ \omega_k &= k\frac{2\pi}{L_x}, \\ a_k &= \frac{2}{L_x}\int_0^{L_x} f_x(t)\cos(\omega_k t)dt, \\ b_k &= \frac{2}{L_x}\int_0^{L_x} f_x(t)\sin(\omega_k t)dt. \end{aligned}$$

Here ω_k are the harmonics of f_x, a_k are the even Fourier coefficients, and b_k are the odd Fourier coefficients of f_x, see [7], [12], [48]. If we consider the approximation with a single Fourier term, we will obtain easily a good candidate solution of problem (7).

Theorem 9. *In problem (7), if* $h(x) \equiv H = n\overline{f} \in \mathbb{R}$ *and*

$$f(t) = \frac{a_0}{2} + a_1\cos(\omega_1 t) + b_1 \sin(\omega_1 t), \tag{5.17}$$

then

$$\min_{s\in\mathbb{R}^n} \|g - h\| = 0$$

and it is obtained for any set of shifts of the form,

$$s = \left\{\frac{2\pi k}{d}\right\}_{k=1,\dots,n}$$

for any d *divisor of* n *other than* 1, *i.e.* $\frac{n}{d} \in \mathbb{Z}, d \neq 1$.

Proof. For any $t \in \mathbb{R}$ the solutions of the complex equation,

$$z^d - e^{itd} = 0$$

define the set

$$\left\{e^{i(t+\frac{2\pi}{d}j)}\right\}_{j=1,\dots,d}$$

of points in the complex plane. For $d > 1$ the set describes a regular polygon centered in the origin. Therefor the next complex expression vanishes,

$$\sum_{j=1}^{d} e^{i(t+\frac{2\pi}{d}j)} = 0.$$

If n is a multiple of d, the next expression vanishes too,

$$\sum_{j=1}^{n} e^{i(t+\frac{2\pi}{d}j)} = 0.$$

This implies,

$$\sum_{j=1}^{n} \cos(t + \frac{2\pi}{d}j) = 0, \sum_{j=1}^{n} \sin(t + \frac{2\pi}{d}j) = 0.$$

Consequently, if $s_j = \frac{2\pi}{d}j$,

$$g(t) := \sum_{j=1}^{n} f(t + s_j) \equiv n\frac{a_0}{2}.$$

The form (5.17) of f clearly implies that $\frac{a_0}{2}$ is the average of the function f in the domain. Then $g(t) \equiv n\frac{a_0}{2} = H \equiv h(t)$ and the result follows. □

Theorem 10. *Under the conditions of theorem (9). The set of shifts*

$$\{s_j\}_{j=1\ldots n}$$

solve the problem (7) if and only if,

$$\sum_{j=1}^{n} e^{i(t+s_j)} \equiv 0 \tag{5.18}$$

Proof.

$$\begin{aligned} g(t) &:= \sum_{j=1}^{n} f(t + s_j) \\ &= \sum_{j=1}^{n} \frac{a_0}{2} + a_1 \cos(\omega_1(t + s_j)) + b_1 \sin(\omega_1(t + s_j)) \\ &= n\frac{a_0}{2} + a_1 \sum_{j=1}^{n} \cos(\omega_1(t + s_j)) + b_1 \sum_{j=1}^{n} \sin(\omega_1(t + s_j)) \end{aligned}$$

If $g(t) \equiv H$ then

$$a_1 \sum_{j=1}^{n} \cos(\omega_1(t + s_j)) + b_1 \sum_{j=1}^{n} \sin(\omega_1(t + s_j)) = 0 \tag{5.19}$$

differentiating we obtain,

$$\begin{aligned} \omega_1 a_1 \sum_{j=1}^{n} \sin(\omega_1(t+s_j)) - \omega_1 b_1 \sum_{j=1}^{n} \cos(\omega_1(t+s_j)) &= 0 \\ -b_1 \sum_{j=1}^{n} \cos(\omega_1(t+s_j)) + a_1 \sum_{j=1}^{n} \sin(\omega_1(t+s_j)) &= 0 \end{aligned} \tag{5.20}$$

The determinant of the linear system composed by (5.19,5.20) is

$$\begin{vmatrix} a_1 & b_1 \\ -b_1 & a_1 \end{vmatrix} = a_1^2 + b_1^2 \neq 0.$$

Therefor zero is the only solution, i.e.,

$$\begin{aligned} \sum_{j=1}^{n} \cos(\omega_1(t+s_j)) &= 0 \\ \sum_{j=1}^{n} \sin(\omega_1(t+s_j)) &= 0, \end{aligned}$$

From this argument, we conclude that

$$g(t) \equiv H \iff \sum_{j=1}^{n} e^{i(t+s_j)} \equiv 0,$$

as desired. □

As we see, the solutions obtained considering (5.17) are not unique. Any set of shifts satisfying (5.18) is also a solution.

We should not forget that this solutions assume a first order Fourier approximation. The error can be estimated by using the rest of the terms in the series.

$$\sum_{k=2}^{\infty} [a_k \cos(\omega_k t) + b_k \sin(\omega_k t)].$$

If the coefficients of the Fourier approximation of order larger than 1 are relative large. The error in the approximation may destroy the quality of the solutions obtained. Better solutions could be obtained by considering higher order approximations. Nevertheless, this approach give us a family of rather good solutions without even using numerical methods. These solutions can also be used as starting points in heuristics and direct search algorithms while looking for better solutions.

5.3 Full Optimization

Up to this point, we have considered two separated optimization problems. One considering the radial distribution as a control parameter, and the second considering the shifts as control parameters. Now, I will describe the full problem, considering both kind of parameters as variables to be optimized for the process. The production process and the model for it, was already described in chapter 1.

As with the other problems, I will describe a discretized model, and transform the problem into a well know one, a mixed integer program. In a similar way, as done in sections (5.1.1,5.2.1). The reader interested on the details of mixed integer optimization is directed to the books of Wolsey, [37], [53].

5.3.1 Mixed Integer Program Formulation for the Full Problem

As before, we will discretize the periodic domain of the functions using N_x, N_y discretization points in each of the coordinates directions. The domain R is defined as the rectangle $[0, L_x] \times [0, N_y]$. We will consider, a discretized radial distribution using N_p discretization points and define the index sets,

$$I := \{1, ..., N_p\},\ K := \{1, ..., N_x\},\ L := \{1, ..., N_y\}.$$

The discretized radial distribution is then,

$$p(x) := \sum_{i \in I} \beta_i \pi_i(x) \tag{5.21}$$

where π_i are basis functions defined in section (5.1).

We consider N_r rows, and define the index set $J := \{1, ..., N_r\}$. Now we assume only a finite number of possible values for the shifts. Again, we consider only shifts of the form,

$$\varphi_j = \sigma^s := \frac{2\pi s}{N_s}.$$

Then we have only N_s different angles as values for the shifts, and define the last index set as $S = \{1, ..., N_s\}$. The data we will require, will be collected in a single object indexed using all the five sets. In the following way.

The flow of material over the spot (k, l) of the discretization domain due to the row j assuming $\varphi_j = \sigma_s$ and a radial distribution π_i is defined as,

$$A_{i,j,k,l,s}.$$

The total amount of flow in the spot (k, l) can be computed by considering the complete radial distribution and all rows,

$$\alpha_{k,l} = \sum_{i \in I} \sum_{j \in J} \sum_{s \in S} \beta_i \cdot A_{i,j,k,l,s} \cdot \chi_{j,s}.$$

The problem of maximizing the minimum value for $\alpha_{k,l}$, can be expressed as an MIP, in the following way.

Problem 8. *Given the input data $A_{i,j,k,l,s}$ the real variables λ, $\alpha_{k,l}, \beta_i, \delta_{k,l,s}$ and the binary variables $\chi_{j,s}$, with $i \in I, j \in J, k \in K, l \in L$ and $s \in S$.*

$$\max \quad \lambda \tag{5.22}$$

subject to

$$\alpha_{k,l} \geq \lambda, \qquad \forall k \in K, \forall l \in L \tag{5.23}$$

$$\alpha_{k,l} = \sum_{i \in I} \beta_i \cdot \delta_{i,k,l} \qquad \forall k \in K, \forall l \in L \tag{5.24}$$

$$\delta_{i,k,l} = \sum_{j \in J} \sum_{s \in S} A_{i,j,k,l,s} \cdot \chi_{j,s}, \qquad \forall k \in K, \forall l \in L, \forall s \in S \tag{5.25}$$

$$\sum_{i \in I} \beta_i = 1, \tag{5.26}$$

$$\sum_{s \in S} \chi_{j,s} = 1, \qquad \forall j \in J \tag{5.27}$$

$$\beta_i \geq 0, \qquad \forall i \in I \tag{5.28}$$

$$\chi_{j,s} \geq 0, \qquad \forall j \in J, s \in S \tag{5.29}$$

$$\chi_{j,s} \geq 0, \qquad \forall j \in J, s \in S \tag{5.30}$$

Observe that, if either the values of $\chi_{j,s}$ or the values of β_i are fixed the problem becomes linear. Unfortunately, problem (8) becomes non-linear due to (5.24). However, this sort of non-linearity can be removed or, in other words, we can perform a linearization for problem (8).

Let us introduce additional variables $y_{i,j,s}$ such that,

$$y_{i,j,s} = \beta_i \cdot \chi_{j,s}.$$

This additional variables will split the non-linear expression (5.24) and the problem will become.

Problem 9. *Given the input data $A_{i,j,k,l,s}$ the real variables λ, $\alpha_{k,l}, \beta_i, y_{i,j,s}$ and the binary variables $\chi_{j,s}$, with $i \in I, j \in J, k \in K, l \in L$ and $s \in S$.*

$$\max \quad \lambda \tag{5.31}$$

subject to

$$\alpha_{k,l} \geq \lambda, \qquad \forall k \in K, l \in L \tag{5.32}$$

$$\alpha_{k,l} = \sum_{i \in I} \sum_{j \in J} \sum_{s \in S} A_{i,j,k,l,s} \cdot y_{i,j,s} \qquad \forall k \in K, l \in L \tag{5.33}$$

$$\sum_{i \in I} \beta_i = 1, \tag{5.34}$$

$$\sum_{s \in S} \chi_{j,s} = 1, \qquad \forall j \in J \tag{5.35}$$

$$y_{i,j,s} \leq \chi_{j,s} \qquad \forall i \in I, j \in J, s \in S \tag{5.36}$$

$$\beta_i = \sum_{s \in S} y_{i,j,s} \qquad \forall i \in I, s \in S \tag{5.37}$$

$$\beta_i \geq 0, \qquad \forall i \in I \tag{5.38}$$

$$y_{i,j,s} \geq 0, \qquad \forall i \in I, j \in J, s \in S \tag{5.39}$$

$$\chi_{j,s} \geq 0, \qquad \forall j \in J, s \in S \tag{5.40}$$

The problem is to maximize the minimum value of $\alpha_{k,l}$, (5.31,5.32). The equality constraints (5.24,5.25) in problem 8 are transformed in a single constraint (5.33) in problem 9. Constraints (5.34,5.35,5.38,5.40) are the same as in problems 5 and 8. The definition of the additional variables $y_{i,j,s}$ requires constraints (5.36,5.37,5.39).

The problem, as before, is that even for a coarse discretization, we will need to run CPLEX for some days before obtaining an optimal solution and a low gap. And this would not be acceptable particularly if we are interested in multicriteria optimization, where this has to be repeated many times. We will discuss more about the numerics, of all methods on chapter 7.

5.4 Heuristics for Scalar Optimization

If we want to speed up the computation of solutions in our problem, one possibility is to adopt a different perspective. I will consider now the original problem of minimizing one of the criteria functions defined in the previous chapter, and propose some methods for its solution. We consider, as parameters, the radial distribution and a set of N_r shifts corresponding to the number of spinneret rows in the process.

Each criterion is then a function of the radial distribution and the shifts, let us again define such a function as $\mathcal{F}$.

$$\begin{aligned} \mathcal{F}: \quad \mathbb{X} \quad &\to \mathbb{R} \\ (p, \varphi) \quad &\mapsto \kappa \end{aligned}$$

Where the set $\mathbb{X}$ represents the feasible set.

I will not specify which criterion is represented by the function $\mathcal{F}$, because for the heuristics, there is no difference. In the multicriteria approach, we will in fact use several criteria for the same problem.

The optimization problem is again,

Problem 10.

$$\min_{(p,\varphi)\in\mathbb{X}} \mathcal{F}(p, \varphi).$$

The advantage of the model described in chapter 2, is that the simulations can be performed in fractions of a second. This means that we can perform fast evaluations of $\mathcal{F}$. This fact allows us to explore better the search space.

However, the feasible set $\mathbb{X}$ will be a subset of $\mathbb{R}^n$ where typically $n \approx 10$. Besides problem 10 is nonlinear and nonconvex. All this implies the possibility of having several local optimum in a rather high dimensional space. Figure 5.1 depicts a landscape defined in a two dimensional slice of the search space.

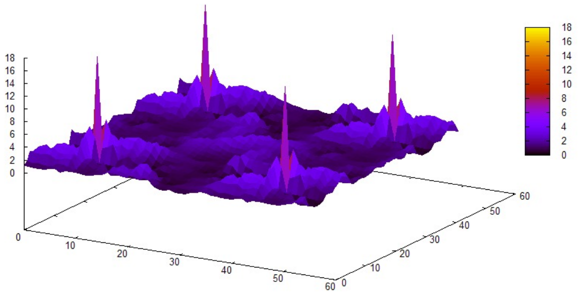


Figure 5.1: Landscape for an objective function of a “slice” of the search space.

In such problems, we will be forced to apply optimization in two steps. In the first one, we should perform a search of the parameter space using some heuristic. For this end, I will propose an evolutionary strategy. After the exploration, a local search has to be applied in order to improve the obtained solutions in a neighborhood. For the local search I use two well known methods, *Pattern Search* and the *Nelder Mead* algorithm.

5.4.1 An Evolutionary Algorithm

An evolutionary algorithm (EA) is a population based metaheuristic optimization algorithm. The mechanisms used in EA are inspired in biological evolution, i.e. reproduction, mutation, recombination, natural selection and survival of the fittest. I will assume here, the reader has some basic knowledge on EA. The EA community has grown fast in the last decades and several excellent publications exists for both, scalar and vector optimization. See for example [8], [20], [49]. The latter is a free online e-book. Figure 5.2 illustrate the steps and cycle of a generic EA.

In EA an initial population P_0 of possible solutions is considered. Then the mechanisms are applied over the population to obtain a new population P_1. The steps are repeated until obtaining a final population P_n. If the mechanism where chosen properly, EA are very efficient in obtaining optimal or very improved solutions as members of the population P_n. Following the argot of the evolution, each population is called *generation.*

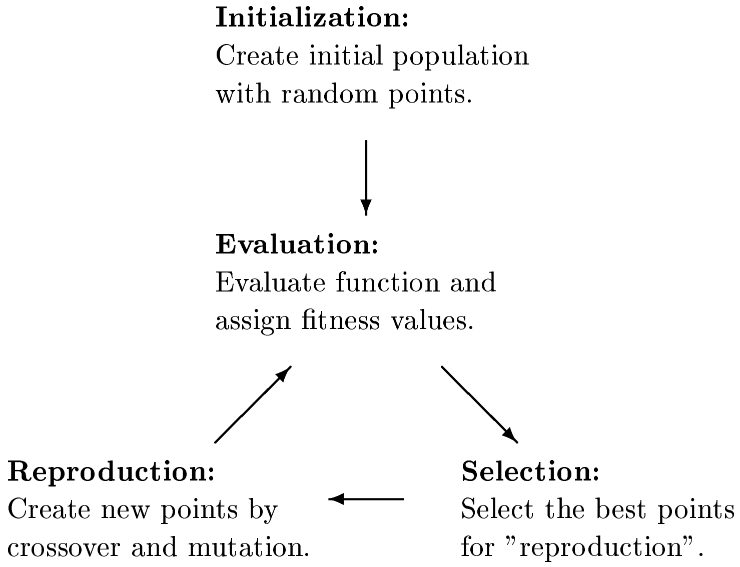


Figure 5.2: Basic cycle of evolutionary algorithms.

Evolutionary algorithms englobe several different techniques according to the way those mechanisms are implemented in a particular problem. To list a few of these related techniques: genetic algorithms, differential evolution, simulated annealing, tabu search, particle swarm optimization, ant colony optimization, state space search and some others. I will now describe each of these steps and how it is applied for our particular problem.

Initialization

For the initialization we need to generate a set of candidate decision points

$$P_0 = \{x_i^0\}_{i=1\ldots n}$$

in the feasible set $\mathbb{X}$. It is generally accepted that such a set should be generated randomly with uniform distribution over the space.

In our problem, each decision point x_i^0 is characterized by a radial distribution p and a vector of shifts φ. We would like to generate the "uniformly".

The uniform random generation of the shifts is not problematic. We choose simply a uniform probability distribution over the interval $[0, 2\pi]$ and sample n points using it.

But for the radial distribution, we have to make a decision. One possibility is to consider the discretization used for the MIP formulations, where a vector, $\beta = (\beta_1, ..., \beta_{N_p})$, $\beta_i \geq 0$, was associated with the radial distribution. If we do this, we can sample values β_i' from a uniform distribution over $[0, 1]$ and apply a normalization,

$$\beta_i = \frac{\beta_i'}{\sum_{i \in I} \beta_i'}.$$

But any parameterization of the function space can be used. In particular we use the one using splines presented in page 29.

Evaluation

Consider a set of solutions,

$$Q_\nu = \{x_i^\nu\}_{i=1 \ldots n_Q}$$

after ν generations. This set will come directly as the initial population, in the first step, but in later generations, the set will be the union of the previous generation and an *offspring* set which will be computed in the reproduction step. In this step, a value of *fitness* is to be assigned to each design point in Q_ν.

In biology the fitness of an organism is the capacity to survive and transmit better its genetic information as compared to competing organisms. In optimization, the fitness should quantify the quality of the solution. The algorithm will keep only solutions with high fitness values, but it is necessary to define it. For scalar optimization, the fitness is mainly (if not totally) related with the value of the criterion function $\mathcal{F}$.

Additional quantities can be included in the evaluation of the fitness to promote the exploration of the space, or to penalize solutions being too close to better ones in the same population.

Functions similar to the following are used,

$$g(x) = \sum_{\xi \in Q_\nu, \xi \neq x} \|x - \xi\|^{-1}.$$

The fitness of a design point x can be defined as,

$$f(x) := \gamma_1 \mathcal{F}(x) + \gamma_2 g(x)$$

for some weights γ_1, γ_2.

It is also possible to use the constraints to penalize the fitness, but I will assume that the population is always composed by feasible points.

In this step, we should also determine how many solutions we will keep in the population. I will simply keep the number of solutions constant and choose those solutions with the highest fitness value. The set of survivors corresponds to the next generation P_ν.

Selection

In the selection step, the algorithm should choose a subset of solutions from the population P_ν for the reproduction. There exist very sophisticated methods for doing this selection. I will apply a *random selection*, where the probability of x of being selected for reproduction is,

$$\frac{f(x)}{\sum_{\xi \in P_\nu} f(\xi)}.$$

Where I will draw n_r points from the n without repetition. What we get afterwards is a new set $R_\nu \subset P_\nu$ of points for reproduction.

Reproduction & Mutation

The algorithm will use all the information gathered to create new points for further steps. There are four different reproduction operations. This operations affect only members of the solution space.

Creation. This operation is used to create a new point, in a similar way as it was done in the initialization step. But the uniform distribution used in the initialization could be substituted by other distributions according to the information recollected during the iterations, for example to avoid the creation of new points close to points already in the population.

Duplication. Sometimes it is convenient to create copies of existing solutions to increase the share of this solution in the population.

Mutation. The mutation is a modification or perturbation of an existing solution. This operation is completely application-dependent, i.e. it has to be defined for each problem. For our problem, first I will choose randomly a small number of components of the design vector. Each one of this values will be mutated either by adding a small perturbation or by replacing it with a new random value.

The set of all the new points obtained by such operations is called *offspring*. The offspring together with the original population are now sent to the evaluation step.

An evolutionary algorithm is usually stopped after a certain number of generations is attained. For our problem, we will not run many generations. We use EA to perform an exploration of the search space, but use other methods for the local improvement of the obtained solutions.

5.4.2 Pattern Search

Pattern Search (PS) is a family of direct search methods applied in the optimization of nonlinear problems. The term "direct search" describes a sequential examination of candidate solutions, comparing each one with the best found so far, together with an strategy to choose the next candidate. In directed searches this strategies do not use information or techniques of classical analysis (gradients, hessian, etc.) either because of lack of differentiability on the functions or because the computations required are too demanding. See [3], [10] and [28] among many others.

Given a candidate point x_i, the PS method perform a search for a next iteration point x_{i+1} from a mesh pattern $\mathcal{P}_i$. The mesh is composed from a finite set of direction vectors, these vectors are better represented with a matrix D using the vectors as columns.

A general pattern search algorithm.

The structure of a general pattern search algorithm (GPS) is simple and similar to other methods for nonlinear optimization. For each step corresponds a decision point x_k and a step size s_k. The algorithm then determines if the step size is acceptable and then updates are applied on all the components of the algorithm.

The algorithm starts with an initial point $x_0 \in \mathbb{R}^n$, a given step length parameter $\Delta_0 > 0$ and a mesh pattern $\mathcal{P}_0$ defined as

$$\mathcal{P}_0 = \{x_0 + d_i\}_{i=1 \dots p_0}. \tag{5.41}$$

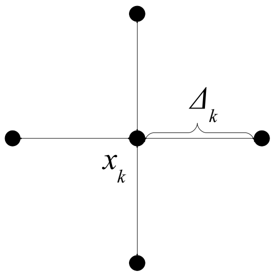


Figure 5.3: Pattern used in our problem.

Figure 5.3 depicts the pattern chosen in our application. There exist, however, many other different patterns. Some of the most typical are illustrated in figure 5.4.

The pattern $\mathcal{P}_0$ is better represented as a matrix $D_0 \in \mathbb{R}^{n \times p_0}$ composed with the vectors d_i as columns. A condition for convergence is that such vectors must span the entire domain positively. If the domain is $\mathbb{R}^n$, this means,

$$\forall x \in \mathbb{R}^n,\ \exists \alpha \in \mathbb{R}^n_+,\ \text{s.t. } , x = D_0 \alpha^T.$$

Such a positive spanning set contains at least a positive basis. A positive basis have between $n+1$ and $2n$ vectors. The GPS algorithm explore neighbor points using the pattern in order to find a point improving the criteria function. If the algorithm finds a better point, the position x_k is updated and the step-length is increased by a factor equal or bigger than 1. If no point in the patter is better that x_k, the step-length is reduced by a factor smaller than 1. The steps continue until the step-size Δ_k is smaller than a given number $\delta > 0$.

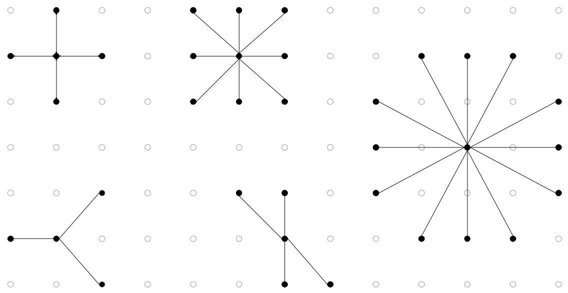


Figure 5.4: Some possible patterns.

Pseudo-code of the method is illustrated below in algorithm 1. The index k defines the step and i the vector component of the pattern. An expression like "$y \leftarrow 1$" means that the value of 1 is assigned to the variable y.

According to [28], [46], the assumption of some smoothing conditions over the function imply global convergence to stationary points in the sense,

$$\liminf_{i \to \infty} \|\nabla f(x_i)\| = 0.$$

Convergence results with weaker conditions (in particular non-smooth and even discontinuous functions) and linear constraints can be found in [3], [27].

```
Data:
Starting point: $x_0 \in \mathbb{R}^n$
Vectors in pattern: $p_0 \in \mathbb{N}$
Step length: $\Delta_0 > 0$
Minimum step length: $\delta > 0$
$i \leftarrow 1,\ k \leftarrow 0,\ j \leftarrow 0$
while $\delta < \Delta_k$ do
    $\xi \leftarrow x_k + \Delta_k d_i$
    $j \leftarrow j + 1$
    if $f(\xi) < f(x_k)$ then
        $k \leftarrow k + 1$
        $\Delta_{k+1} \leftarrow 2\Delta_k$
        $x_{k+1} \leftarrow \xi$
    else
        $i \leftarrow (i + 1) \pmod{p_0}$
        if $j \geq p_0$ then
            $\Delta_{k+1} \leftarrow \Delta_k / 2$
            $j \leftarrow 0$
        end
    end
end
```

Algorithm 1: GPS Algorithm

5.4.3 Nelder-Mead Algorithm

To finish this chapter, I will explain an additional method used for the solution of general non-linear optimization problems. Similar to pattern search, this method does not make use or assume differentiability on the problem.

I include the method for two reasons. As we will see this algorithm use several points to start the local search, in the multicriteria framework, we do have a collection of points close to the problem we want to solve. It is natural then to choose an algorithm which make use of that information. As a second incentive for including Nelder-Mead we will propose later in chapter 6 a method for exploring alternatives in the context of a decision support system.

The convergence properties of the original Nelder-Mead algorithm are much more restricted even for low dimensions, see [26] and [31]. But further corrections to the method make the algorithm very efficient in practice [36].

In the Nelder-Mead algorithm (NMA) a population of points; usually a simplex in an n dimesional space, explore the design space searching for the optimum of a given criteria function. In this sense, the NMA has similarities with the evolutionary algorithms. Even when NMA is sometimes considered a heuristic, the mechanics for exploring the space and finding new candidate points are inspired in the expected behavior of the search space. The Nelder-Mead algorithm has many similarities with the golden section search method used in the minimization of single valued minimization problems. In fact a variant of the original algorithm is described in [36] inspired on the golden section search.

The algorithm improves at each step the worst point in the population by assuming that better points can be obtained if this point is translated in the direction of the "good" points from the population. This translation is usually performed with a reflection with respect to a weighted center of the good points. If the worst point can not be improved in the tries, the simplex is shrunk in the direction of the best point. Then the steps are repeated.

Next I describe a generalized Nelder-Mead algorithm.

Start by choosing a set $S^0 = (x_1, ..., x_{n+1})$ of $n+1$ points, $x_i \in \mathbb{R}^n$.

1. **Sorting:** Sort the points of the set according to the values of the function, i.e. such that, $f(x_1) \leq f(x_2) \leq ... \leq f(x_{n+1})$.

2. **Reflection:** Define,

   $$\tilde{x} := \frac{2}{n(n+1)} \sum_{i=1}^{n} n x_i.$$

   Compute the reflection x_r such that $x_r - \tilde{x} = \tilde{x} - x_{n+1}$.

   If $f(x_r) < f(x_n)$ goto step 3.

   Else goto step 4a if $f(x_r) < f(x_{n+a})$, if not then goto step 4b.

3. **Expansion:** Compute the expansion x_e such that $x_e - x_r = \tilde{x} - x_{n+1}$.

   If $f(x_e) \leq f(x_r) < f(x_1)$ then $x_{n+1} \leftarrow x_e$.

   If only $f(x_r) < f(x_1)$ then $x_{n+1} \leftarrow x_r$.

   Goto step 1.

4. **Contraction:**

   A: Compute the outer contraction x_{oc} such that

   $$x_{oc} - \tilde{x} = \frac{1}{2}(\tilde{x} - x_{n+1}).$$

   If $f(x_{oc}) < f(x_r)$ then $x_{n+1} \leftarrow x_{oc}$ and goto step 1.

   Goto step 5.

   B: Compute the inner contraction x_{ic} such that

   $$\tilde{x} - x_{ic} = \frac{1}{2}(\tilde{x} - x_{n+1})$$

   If $f(x_{ic}) < f(x_{n+1})$ then $x_{n+1} \leftarrow x_{ic}$ and goto step 1.

   Goto step 5.

5. **Shrinking:** Shrink the set according to, $x_i \leftarrow \frac{1}{2}(x_1 + x_i)$, and goto step 1.

In the steps of reflection and expansion, the algorithm attempts to improve the worst points in the direction of the best points. If this fails, is may be that the step was too large. The algorithm performs then smaller steps using contractions. If neither of those points is better, the algorithm shrinks the set, sending all points in the direction of the best point found so far.

Figure 5.5 depicts all possible points that could be generated by a step of the algorithm.

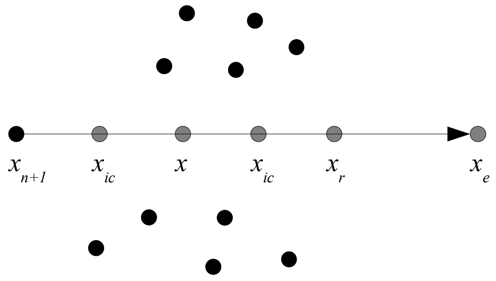


Figure 5.5: Points that can be generated in a NMA step.

If the points of the simplex are carefully chosen the Nelder-Mead algorithm may not converge to a local minimum. It is a standard approach to restart the search with a new simplex around the best point, like in other methods. But [47] suggest the use of two safeguards to assure convergence properties. First the substitution of strict descent by a stronger criterion of fortified descend which somehow forces a certain percentage of improvement, and second by keeping the angles of the simplex away from zero.

Chapter 6

A Decision Support System

Introduction

In the design of a production process it is important to choose the "optimal" parameters in order to achieve high quality characteristics in the final product. When a single number can be used to evaluate the performance of a process, we know there exist a set of parameters optimizing such performance; at least in theory. The goal is then, to find such a set of optimal parameters.

As we have seen, in multicriteria optimization when the model of the process happens to contain more than a single criterion, the very nature of the problem is that no single optimal solutions exists. The best we can do is to define a set of solutions where none of them is totally preferred over the others. Without further information about preferences between efficient solutions there is no way to distinguish among them. Nevertheless; at the end, a single solution or design has to be chosen. Normally a person (or group of persons) judge the possible solutions and make the final decision according to other preferences not considered in the model or optimization.

A Decision Support System (DSS) is an information system in the form of software designed to support the decision making activities. DSS are supposed to improve the process of identification and solution of problems related with decision making. It is usually an interactive software tool intended to help the decision maker recollect, organize and select information.

In this chapter we will see how the mathematical tools presented so far, can be used to create a Decision Support System for the design of the production process described in chapters 2 and 3.

The chapter is divided in two sections. In section 6.1 a general overview of the decision making process is presented together with the concept of Decision Support Systems (DSS) in particular a *Model Driven DSS* where the system is based around a mathematical model. In section 6.2 the controls used in the system are presented and how these can be used in the problem of exploring the Pareto frontier. Also three strategies for the navigation of the Pareto frontier are considered, based in the theory of scalar and multicriteria presented in the previous chapters.

6.1 Decision Making

Decision making is the process of selecting a course of action among different alternatives. It is applied in many different situations in management, engineering and science. See [2], [18], [50]. It is a cognitive process that may; or may not, be rational depending on the person making the decision.

At the end, the decision making process should produce a “choice”. If the process is rational, such choice is taken based on the preferences of the person making the decision. An assumption I use along in this chapter, is that the decision making process is rational. This means that the choice is not taken at random, but that some sort of hidden process happens in the mind of the decision maker. In particular I assume that, if among a set of alternatives the person prefers one over the others, this preference do not change during the decision making process. It is also implied that a certain hidden order exists in the set of alternatives. This assumption is not obvious, because preferences may very well change over time.

When the set of alternatives is small and each of them can be evaluated using any kind of “performance measurement” the decision making process becomes easy. It is simply a matter of choosing the best option among the set of alternatives. But difficulties arise either when the set of alternatives is too large to consider all of them or when there is no specific known performance attached to each alternative.

6.1.1 Model Driven Decision Support Systems

In some applications it is possible for the decision maker to analyze the consequences of a particular choice without implementing it in the real world. This is the case when a surrogate problem can be constructed, for example when a model exists for the real situation.

The simple existence of a model can be a huge advantage to the decision maker. He can get information about the consequences of making a decision without the responsibility of making a definitive choice.

The simulation of the spunbond process presented in chapter 3 is fast enough to allow the system to return the output of the model to the user after selecting a set of control parameters in real time. The control parameters can be altered by the user and the result of the simulation can be observed immediately on the screen.

But skillfully using the model, can lead to much useful possibilities. A *Model Driven Decision Support System* is a computational system or software based around a mathematical model of the decision problem developed to help the decision maker in his task.

Such a system has emphasis in the manipulation of a model, and not too much in the intensive accumulation of data. Even when it is a good strategy to pre-compute sets of alternative decisions based in the criteria functions that can be quantified.

If the decision problem can be described as a multicriteria optimization problem, we make the assumption that only efficient solutions; in the sense of Pareto, are of interest for the decision maker. The central aim in a DSS for this kind of problems is the possibility to approximate and explore the set of efficient solutions in an easy way. Facilitating the task of finding an adequate choice, for the decision maker The mathematical tools presented so far in this dissertation will be a great help in the task of allowing and simplifying the exploration of the Pareto frontier.

The system interacts with the user using a *graphical user interface* (GUI). In a GUI the interaction of the user is done using special graphical elements called *widgets* (sometimes referred as *controls*). Buttons, scroll bars and sliders are examples of widgets in many computer applications. In the next sections I will present the widgets used in the DSS and the strategies adopted to facilitate the decision making process.

6.2 Pareto Frontier Navigation

The start of a session of decision making using a DSS presents the decision maker an efficient decision together with a way to visualize the output of the model and the values of all criteria functions. Some systems may present several alternatives simultaneously.

During the decision making process, the user interacts with the system in different ways and the system suggest each time new alternative solutions based on the actions performed by the decision maker.

6.2.1 Visualization and Control Widgets

Our DSS include two different widgets used to present the criteria values to the user. These widgets can be used to explore a database of solutions and allow navigation of the Pareto frontier.

Fleece Viewer. The density function $\rho : \mathbb{R}^2 \to \mathbb{R}$ can be visualized using a color map in a rectangular region of the screen. The color of each pixel in the rectangle depends on the value of the function. A color map; usually called *lookup table*, is used to specify colors from scalar values. Figure 6.2.1 depicts the fleece viewer widget. The bottom of the widget shows the used lookup table. Green represents the average value of the density function, yellow/red represent high values and cyan/white represent low values.

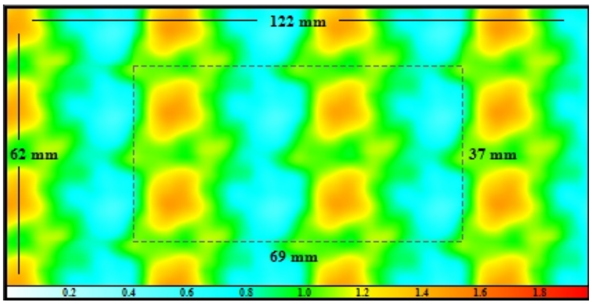


Figure 6.1: Widget to visualize the fleece density function.

The user may perform some actions using this widgets. A mouse click on the image will change the scale of the visualization and a click in the lookup table is used to change the color map. The small black rectangle in the center represents the periodic domain of the function.

Fleece Projections Widget. The projected functions described in section 3.3 for measuring the presence of stripes are depicted using a special widget. The widget essentially plots the function values, together with the maximum and minimum values on the corresponding projection.

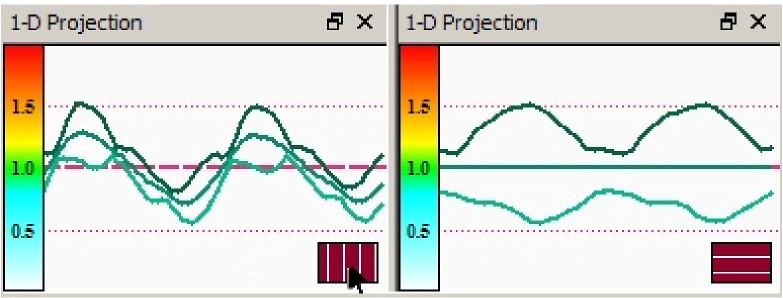
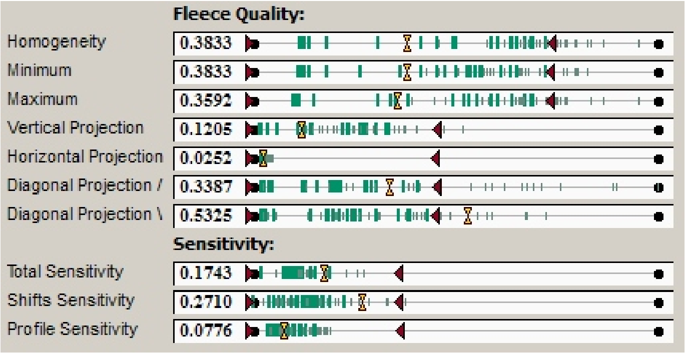


Figure 6.2: Widget to visualize the projected functions.

The user may interact with this widget selecting which projection to visualize by clicking on the red rectangle on the bottom right corner. Figure 6.2.1 presents two projections of the same density function depicted previously on figure 6.2.1.

Criteria Sliders. A typical way of visualizing a high dimensional vector is the use of one dimensional projections. In this case each coordinate of the vector is plotted in a separate line or segment. Some times, this segments are depicted with one of the extremes in a fixed point and the second extreme equally distributed along a circumference. I will prefer to align such segment parallel to each other and use a modified slider to control each component. Figure 6.2.1 depicts the sliders for 10 different criteria functions. The yellow indicators represent a single vector in a 10 dimensional space. The numbers in the left side are the actual components of the vector.



Fleece Quality:

Homogeneity	0.3833
Minimum	0.3833
Maximum	0.3592
Vertical Projection	0.1205
Horizontal Projection	0.0252
Diagonal Projection /	0.3387
Diagonal Projection \	0.5325

Sensitivity:

Total Sensitivity	0.1743
Shifts Sensitivity	0.2710
Profile Sensitivity	0.0776

Figure 6.3: Sliders depicting the values of the criteria functions.

The criteria sliders depicts also constraints for the values for each the criterion. In the figure, the red triangles represent the lower and upper constraints. The background of the sliders presents the values of a database of solutions previously computed. The green ticks represent feasible vectors, in the sense that all criteria values are contained in the constraints specified by the slides. The gray small ticks represent criteria vectors with at least one component outside from the specified constraints. In particular, all ticks to the right of an upper constraint are gray.

The sliders allow the user to pick directly a solution, based on the value of one of the criteria functions. The user may also click in a void region of the slider, suggesting the system the task of finding a solution close to that point. We will see later, how this action will lead to a goal program.

Using the sliders, the user may also change the constraints as he wishes. Such constraints may be included on the optimization problems presented to the system.

Criteria Plane Widget. The best we can do to visualize a high dimensional object, is to present two dimensional projections of it. Even computer 3D graphics are no more than 2D projection on the screen. The *criteria plane widget* will depict a two dimensional projection of the solutions using two of the criteria functions. This widget will directly present the information related with the trade-off between those two criteria functions.

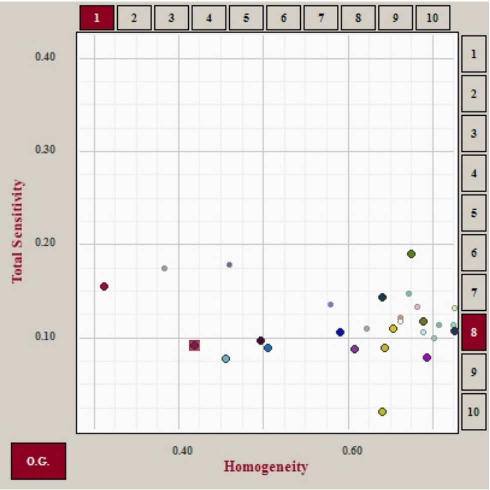


Figure 6.4: Criteria plane widget depicts trade-off between two criteria.

The user may interact with the widget by choosing; on the right side and on the top, the two criteria functions to be used in the projection. Figure 6.2.1 presents a plane widget for 10 criteria functions. The red rectangles on the top and right sides represent the criteria considered in the projection. In that example the homogeneity against the total sensitivity.

The plane widget can also be used, as the sliders, to let the user choose a solution from the database or to propose the value of two criteria in a single click. Such values may also be used later for obtaining new solutions using goal programming.

6.2.2 Selection from a Database

The simplest kind of navigation can be performed by simply allowing the user to explore a database of solutions previously computed using multiobjective optimization algorithms. The user can explore the database using the criteria sliders and/or the criteria plane widget and select decisions based on the criteria values.

If a sufficient dense database of solutions was computed, is such a way that interpolation is possible between neighbor points the system may return an interpolated solution without performing any kind of optimization. This is usually possible for a small number of criteria, in particular we will use this approach for two criteria.

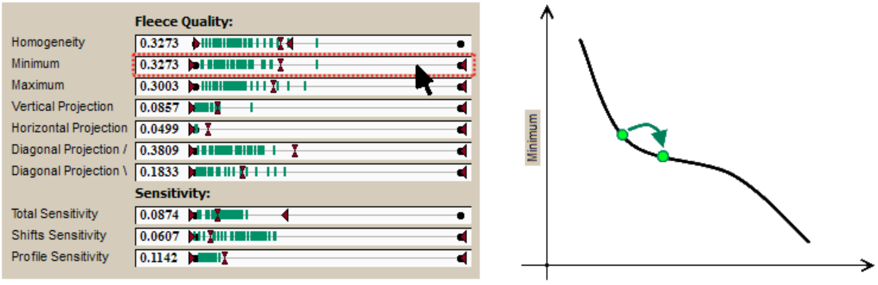
6.2.3 Weights Variation

Variation of the weights vector $(w_1, ..., w_m)$ can be used directly as an alternative to explore the Pareto frontier. Starting from an efficient point the system will ask the decision maker to choose one of the criteria functions which should be improved. A modified weights vector $(w_1, ..., w_i + \delta, ..., w_m)$ is obtained, where w_i is the criteria to be improved and δ may be fixed by the system or an input from the decision maker. The solution of the new aggregated function for the changed set of weights will be then presented to the decision maker for further consideration.

This approach assumes a certain knowledge in the decision maker in respect to the basis of the mathematical model and the multicriteria structure of the problem. In particular the understanding of the effect of the weights in the efficient solution obtained from the aggregated function and how these affect the corresponding point in the Pareto frontier.

In terms of implementation is very simple. It represents a single continuation step, see section 4.2. The only requirement on the system is to solve a new scalar aggregated function.

This kind of exploration is useful when the actual solution is; in some sense, appealing to the decision maker, but he would like to explore the trade-offs



Fleece Quality:

Homogeneity 0.3273

Minimum 0.3273

Maximum 0.3003

Vertical Projection 0.0857

Horizontal Projection 0.0499

Diagonal Projection / 0.3809

Diagonal Projection \ 0.1833

Sensitivity:

Total Sensitivity 0.0874

Shifts Sensitivity 0.0607

Profile Sensitivity 0.1142

Figure 6.5: User selects criterion to be locally improved.

among the criteria. This is in essence a local exploration and clearly it is not suitable for reaching regions of the Pareto frontier far away from the actual solution.

The weights variation can be extended by asking the decision maker to choose a subset of criteria to be improved; and perhaps even coefficients to reflect the importance of each criteria. Although possible, this approach is not considered a friendly alternative for the decision maker. A good DSS is supposed to be simple and easy to use.

6.2.4 Setting Goals

Another natural way of exploring of the Pareto frontier is the possibility to let the decision maker impose a certain "goal vector" in term of the values in the criteria functions and let the system find the parameters to achieve that goal. This approach is sometime referred as *method of global criterion* or *compromise programming*, see [25], [32], [40].

In this kind of optimization problem, a vector $y^* = (y_1, ..., y_m) \in \mathbb{R}^m$ is chosen and the problem is then to find a set of decision variables $x = (x_1, ..., x_n) \in \mathbb{R}^n$ such that, $f(x) = y^*$.

If the values required in the goal vector are too strict, the problem may not have a solution. Usually the problem is from the beginning stated as a minimization of a distance to the goal,

$$\min_{x \in R} \|f(x) - y^*\|,$$

for some norm in $\mathbb{R}^m$.

It is possible to select an arbitrary goal vector. But if an efficient solution is at hand, the decision maker may want to choose a goal vector with improvements in the criteria he is interested, based in the given solution. In this way, the method can be used to allow exploration starting from an efficient solution. Instead of choosing weights, the decision maker is asked for specific values of the criteria functions. There is no need of weights at all. This method can be implemented in a very natural way and keeping at the same time, an intuitive interface for the user as follows.

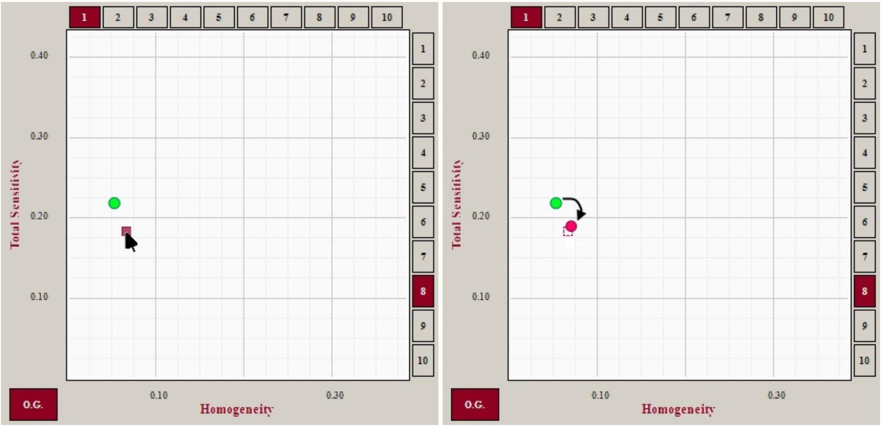


Figure 6.6: User selects a point (value of two criteria) to start a goal optimization.

A proposed solution is presented to the user, perhaps together with a set of efficient solutions in a database previously computed. Using the criteria sliders and plane widget, the user will have an idea of the values that are possible to obtain. If using the plane widget, he can then select two among the criteria and see directly the trade-off between those criteria in the plane widget. The user may now select a goal in that projection plane; specifying in this way, values for the two criteria. The values of those two criteria will be set in a goal vector, and the system will solve the minimization problem by keeping the rest of the criteria fixed with the actual solution. With a criterion slider the user may want to choose the value of a single criterion and the system will solve the goal program keeping the rest of the criteria fixed.

In a more sophisticated approach, the system may ask the user to select additional values for other criteria functions by letting him change the criteria involved to be projected, before starting the optimization. This gives the user more freedom at cost of simplicity.

If there exists an approximation of the Pareto frontier good enough to allow interpolation, the frontier is defined by a finite collection of points vertices of a polyhedra. Some approaches and effective techniques for this situation are described in [34] for interactive multiobjective optimization. In that case there is no need of performing any of the scalar optimization algorithms already presented.

6.2.5 Improving a Value Function

In decision support systems, it is sometimes assumed that the decision maker judges the alternative decisions based in some sort of hidden *value function*, see [32]. It is also assumed that the only way of recovering information about this function is by asking directly the decision maker about a particular solution. This function attempts to model the preferences of the person making the decision.

The main argument against the use of value functions is the inability to uncover the mathematical form of the function, attempting to reconstruct the value function based in the responses of the decision maker. The method I will propose will be inspired on this value function but will not use, or even try to compute specific values of the function. We will only ask the decision maker to compare a set of decisions.

These assumptions allow us to give a simple way of exploring the set of solutions without even directly informing the user about the values of the criteria. In this case, exploration can be performed simple by asking the decision maker to decide on preferences among a small set of alternatives. For example, the specification of the best and the worst alternatives among the proposed solutions.

It is implicitly assumed that a corresponding vector of weights is associated with the best design the decision maker would make. Our task will be to obtain that set of weights based only in the judgments provided by the user.

We consider the abstract map

$$\begin{aligned} \mathcal{P} : \mathbb{R}^m &\to \mathbb{R} \\ w &\mapsto v \end{aligned}$$

where v is the valuation given by the decision maker to the design x obtained as a solution of the aggregated function τ_w using the weights vector w.

The computation of such a function; if possible, requires two steps. First a scalar optimization problem has to be solved considering a weights vector w. After that, the solution is presented to the decision maker for judgment.

In practice, the decision maker may not need to give a specific value to the decision. As I have mentioned, we can perform optimization by only comparing among solutions. I will propose here a family of strategies based on this hidden valuation function, and taking inspiration from different versions of the Nelder-Mead algorithm presented in 5.4.3. Nelder Mead seems an appealing alternative because it does not require specific evaluation of the function to be minimized.

The most direct way to apply the Nelder-Mead algorithm is to present the user a set of $m + 1$ solutions and ask him to order the set according to his judgment, from the best option to the worst. This can be done indirectly by asking the user to give a note, say between 0 “completely unacceptable” and 10 “excellent choice”.

The version of the algorithm presented in section 5.4.3 can be directly applied to improve the worst solution. The system will then propose a new solution to the user obtained using a Nelder-Mead step, i.e. improving the worst solution searching in the direction of the good solutions in the space of weights. It may be worth to mention that the search is not performed in the parameter space. A perturbation of the parameters will take the solution away from the Pareto frontier and may create non efficient decisions.

If the task of sorting the proposed solutions in order of preference seems too demanding for the decision maker. A simpler alternative is to judge only

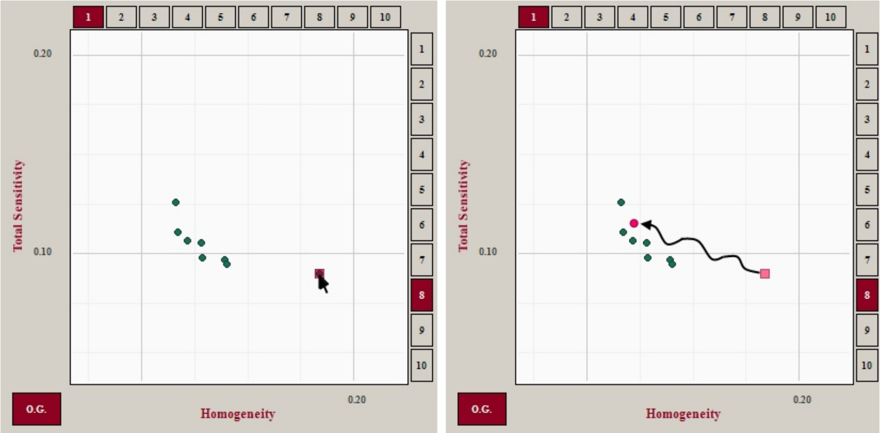


Figure 6.7: User selects the worst point and system improves it.

the solutions as "acceptable" or "not acceptable". At least one solution in each class. This kind of valuation can be used as an order for the Nelder Mead algorithm by artificially imposing a value of 1 to the accepted solutions and a value of 0 to the rejected ones. The steps of "reflection", "expansion", "contraction" can now be applied as usual to the rejected solutions.

At each step, the proposed solutions will become better from the perspective of the user. The step may be repeated until the decision maker is satisfied with one of the presented solutions.

Chapter 7

Numerical Results

Introduction

In the previous chapters I have explained the production process, the mathematical model and the proposed strategies for solving the problem of designing an optimal process with respect to the mentioned quality measurements. But no numerical result has been presented so far. The purpose of this chapter is to concentrate all the numerical aspects of the theory presented before, the numerical assumptions and some other implementation issues.

The results of the central problem of the dissertation are presented in this chapter. We presents the results of the methods to approximate the Pareto frontier for a highly non linear multiobjective optimization problem, computing enough points in the frontier to allow further exploration of the set using linear interpolation.

The outline of the chapter is closely related to the structure of the previous chapters. In section 7.1 I present the aspects related with the direct simulation of the process, the results obtained and some proposed ideas to make the simulation faster. Section 7.2 presents the numerical results related with the sensitivity. The special way of parameterizing the radial distribution using splines, already proposed in section 3.4.1, is justified in section 7.3. In section 7.4, the results of the implementations of all scalar optimization strategies proposed in section 5 are shown. To finish in section 7.5 with results related to the problem of approximating and tracking the Pareto frontier in such a way that interpolation can be performed between the computed points.

7.1 Simulation of the Process

In section 3.2 I have presented a mathematical model of the web formation process. Here I will present a more detailed description of the simulation and implementation of this model.

The model of our spunbond process is totally described by a set of parameters. These parameters represent measurements, distances and positions of the components of the machine used for the production. As I mentioned, we consider two kinds of control parameters (shifts synchronization and radial distribution). Some of the reasons for this choice will be explained in the following sections.

But even when we only use these as control parameters there are also other fixed parameters needed in the description of the production process. The following is a list of the fixed parameters. These parameters determine a given machine or process.

- **(v_b) Band Speed.** Below the rotating spinnerets, the conveyor band is moving with a given speed. Even when the speed can be mechanically altered, these changes would have an effect in previous and further steps of the global process. Therefore, we better consider the speed of the band as a constant parameter.

- **(N_r) Number of Rows.** The spinnerets are arranged in rows. Different kind of processes or machines include particular number of rows. Again, we do not consider optimization with respect to the number of rows, but take the number as fixed. We can, nevertheless, compare the levels of quality obtained by varying this number. In general, more rows improve the levels of quality.

- (L_y) **Distance of Spinnerets in a Row.** In every row, the spinnerets are assumed to be equally distributed, and L_y is the distance between any of them. This distance is optimized beforehand in the real process by considering the projected function perpendicular to the band movement mentioned in section 3.3.2, see [16].

- (d_k) **Distance from Row-1 to Row-k.** The rows are placed parallel to each other and perpendicular to the band velocity. The rows are fixed mechanically in place, for this reason the distances between the rows are taken as constants during all the process. Besides, changes in these distances have the same effect as changes in the shifts synchronization.

- (R_s) **Radius of the Radial Distribution.** This is the maximum distance where the spinneret deposits fibers. As the distance in the spinnerets the radius is also optimized for the perpendicular projection of the fleece.

- (ω) **Rotational speed of the spinnerets.** We assume further, that all spinnerets are rotating with the same speed. As we mentioned in the model, we also assume that the spinnerets in a row are synchronized always in the same direction. But different rows have different relative orientations given by the shifts. This is the only parameter that could be used as a control, but it is not used. We will see later that increasing the rotational speed, improves the expected homogeneity of the fleece but at the same time makes the process much more sensitive.

Our control parameters are, as already explained the following.

- (p) **Radial distribution.** The radial distribution is a function $p : [0, 1] \to \mathbb{R}_+$. For the computations we use a finite number of points to approximate it.

- (ϕ_i^{phase}) **Phase Shifts.** The phase shift i represents the angle of the spinnerets on the row i with respect to the angle of those of row 1 at a given moment.

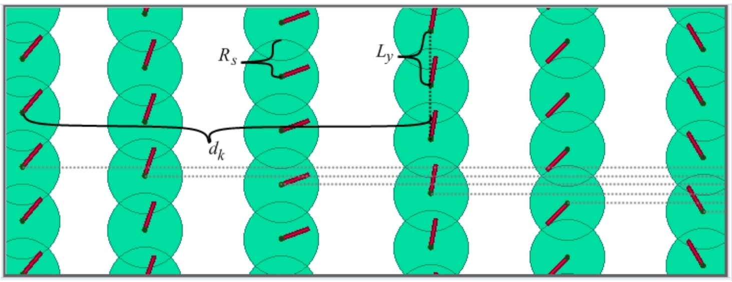


Figure 7.1: Measurements and positions of the spinneret rows.

Additionally, we consider the band shifts as the angles of the spinnerets relative to the moving belt. If we consider a spot in the band, moving with the band, and we measure the angles of the spinnerets on the rows when passing over this spot. The band shifts represent the difference between the angle of the row i and the angle for the first row.

The relationship between these shifts is easily obtained. The time required to move a point in the band from the first row to the row i is given by $T = d_i/v_b$. In this time, the angle of the row i increases $T\omega = d_i/v_b\omega$. If L_x is the period in the direction of the band. Then the time a spinneret require to complete a revolution $2\pi/\omega$ is the same as the time (L_x/v_b) the band needs to move a distance L_x. From this, the relation between the band shifts and the phase shifts can be obtained,

$$\phi_i^{band} = \phi_i^{phase} + \frac{d_i}{L_x} 2\pi. \tag{7.1}$$

The output of the simulation is the density function ρ, representing the amount of material deposited over a particular spot in the band. As we saw, this is a function $\rho : \mathbb{R}^2 \to \mathbb{R}$ periodic in both directions, with periods L_x, L_y.

7.1.1 Direct Simulation of the Process

We consider a discretization of the periodic domain of the density function, by taking N_x points in the x-direction and N_y points in the y-direction. The discretization creates a partition of the periodic domain in rectangular cells. The simulation should be able to estimate the amount of material deposited over each of these cells.

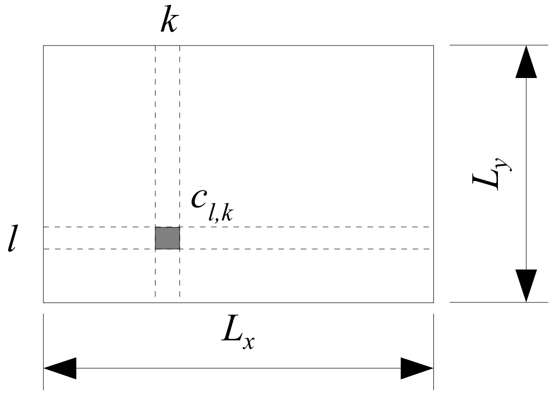


Figure 7.2: Cell in the periodic domain of the fleece.

If the rotational speed of the spinneret is ω. The time to complete a full revolution is $T = \frac{2\pi}{\omega}$. It is possible to simulate the deposition of fibers due to one spinneret using a discretization in time.

Consider time increments of δ_t as a fraction of the total time T. This is, the time points $\{0, \delta_t, 2\delta_t, ..., T\}$, and consider a discrete set of points $\{r_p\}_{p=0,...,N_p}$ in the spinneret at a distance $r_p = \frac{p}{N_p}$ from the rotation center. Equations 3.5 in section 3.2.1 return the position of a point over the belt after time t. According to this equations the position is given by,

$$\overline{x}(r,t) = \begin{pmatrix} x(r,t) \\ y(r,t) \end{pmatrix} = \begin{pmatrix} v_b t + r \cdot sin(\omega t) \\ r \cdot cos(t) \end{pmatrix}. \tag{7.2}$$

If in that moment the point is located over the cell $c_{l,k}$ of the domain. The flow of material in the cell increases according to the value $p(r)$ of the radial distribution. For the simulation we consider time steps δt, and in each step we run the discretized points in the spinneret to find out over which cell it deposits fibers. The simulation of the deposition of material can be performed with the next algorithm.

We should be careful with the chosen time steps and radial points. If the time step is too large, we may skip cells and miss the effect of the spinneret on those cells. The time step should be short enough to cover all cells.

```
Data:
Time step: $\delta t$
Radius step: $\delta r$
for $t=0$ to $T$ do
    for $r=0$ to $1$ do
        $r = r + \delta r$.
        Compute position $x = v_b t + r \cdot sin(\omega t), y = r \cdot cos(\omega t)$.
        Find cell $c$ containing $(x, y)$.
        Increase flow in $c$, i.e., $c_{l,k} = c_{l,k} + p(r)$.
    end
    $t = t + \delta t$.
end
```

Algorithm 2: Algorithm for the simulation of the process.

In a time interval δ_t, a point in the band moves $\delta_t v_b$ and a point at a distance r in the spinneret moves a distance of $\delta_t \omega r$ along the arc of a circle of radius r. This combination of movements means that any point in the spinneret cannot move more than $R_s \delta_t \omega + \delta_t v_b$ during that time interval.

If we choose δ_t such that

$$R_s \delta_t \omega + \delta_t v_b < \text{Size of cell} = |c| = \min \left\{ \frac{L_x}{N_x}, \frac{L_y}{N_y} \right\},$$

and $\delta r < |c|$ then we are sure that no cell will be skipped in the simulation. And, of course, finer steps will improve the results of the simulation.

The required number of time steps can be then estimated with,

$$\frac{T}{\delta t} = \frac{T(\omega R_s + v_b)}{|c|},$$

and the minimum number of points in the radial distribution is

$$\frac{R_s}{\delta r} = \frac{R_s}{|c|}.$$

Given that, the number of iterations in algorithm 2 should be at least

$$\frac{T}{\delta t} \frac{R_s}{\delta r} = \frac{T(\omega R_s + v_b)}{|c|} \frac{R_s}{|c|} = \frac{2\pi R_s (R_s \omega + v_b)}{\omega |c|^2}. \tag{7.3}$$

Which means that the expected times for the simulation grows quadratically with respect to reductions in the size of the cells.

Figure 7.3 depicts the movement of a fixed point over the cells in a time step. If the step is too long, the point may skip a cell which should be avoided.

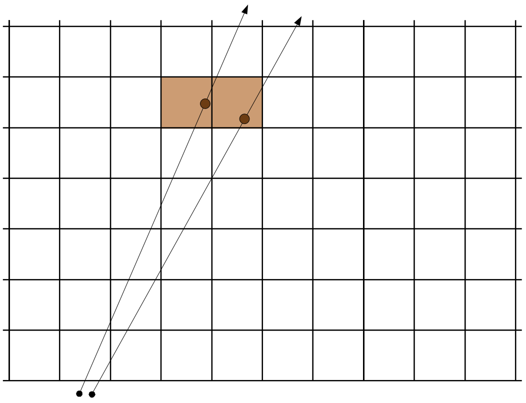


Figure 7.3: A fixed point in the spinneret deposits flow over two neighbor cells.

As a result of the simulation, we got a matrix with entries $c_{l,k}$, representing the fleece generated by a single spinneret. The final fleece can be obtained by superposition using the relations 3.12, 3.14 from section 3.2.

In the applications we considered the dimensions of the periodic domain ranges between 100 and 300 millimeters. In this case considering cells of size 10 or 5 millimeters is small enough; if not too small, in order to obtain credible results. It is important to notice that the refinement in the size of the cell, does not necessarily improve the accuracy of the simulation with respect to the real process. We should not forget that we are modeling a deposition of fibers. And it is clearly absurd to attempt to model the amount of fibers in a cell of size of 1 millimeter or less.

All the fleece images presented are computed with these parameters. The time for the simulation is as expected rather short; less than a second. This fact permits us to perform several simulations fast, apply optimization techniques and even to prepare a software tool where the outcome of the model can be displayed in real time.

7.1.2 Faster Evaluations

If we choose around 100 × 100 points in the periodic domain, the running time for a single simulation is just a fraction of 1 seconds. Even when this time is enough to perform several optimization algorithms, we would like to allow the user to perform real time evaluations of the criteria functions. For this end, we should keep the running time for a single simulation (or at least evaluation of the criteria functions) as fast as we can. This is also important to allow the fast evaluation of the sensitivity.

Once an instance of the process is simulated if we want to perform a change in one of the shifts, the density functions ρ_0, ρ_1 remains unchanged, and it is only required to recompute the final density function ρ by adding up copies of ρ_1. Considering the discretization, this computations are nothing else than a sum of N translations of the matrix obtained in the last section. Being N the number of rows. These operations can be performed in milliseconds for the required discretization points.

The calculations are, nevertheless, complicated if we require a change in the radial distribution. For this change, a new computation of all density functions is required. Here we would need to perform a full simulation each time we change the radial distribution. But there is a way to go around this problem.

Consider the discrete approximation of the radial distribution using N_r base functions $\{\pi_i\}_{i\in I}$.

$$p(x) \approx \sum_{i\in I} \beta_i \pi_i(x)$$

The functions π_i are the "hat functions" described in section 5.1.

The corresponding density function ρ for the radial distribution p can be obtained by summing up the density functions obtained by assuming π_i to be the radial distribution in the process. We compute the density functions for each of the functions $\{\pi_i\}_{i\in I}$, call them $\{p_i^{\pi}\}_{i\in I}$ beforehand. From the definition of the density function, it is clear that the final density function can be obtained from these pre-computed functions.

$$\rho = \sum_{i \in I} \rho^{\pi_i}$$

Considering this, a change in the radial distribution will require a sum N_r matrices, where N_r is the number of discrete points in the radial distribution. For our process we need around 15 discretization points. This operations can be performed fast enough to improve the real time simulation.

7.1.3 Fixed Parameters

In all simulations of this chapter the following parameters, based in those of a real process, are considered as fixed.

- Band speed, $v_b = 80$.
- Distance of spinnerets, $L_y = 110$ mm.
- Distance between rows, $d_k = 900k$ mm.
- Radius of radial distribution, $R_s = 130$ mm.
- Rotational Speed, $\omega = 400$ rpm.

As we already know, the shifts and the radial distribution are in some instances considered fixed, and in other cases taken as design variables to be optimized. We will use 6 shifts in all the instances.

7.2 Computing the Sensitivity

In this section I will present a numerical analysis of the sensitivity or uncertainty of the process. For the numerical experiment I will consider the machine parameters from section 7.1.3, and the following,

- Band shifts, $\phi_i^{band} = 120i$.
- Radial distribution, $p : [0,1] \to \mathbb{R}_+$, as depicted in figure 7.4.

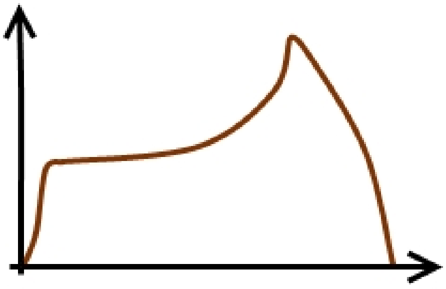


Figure 7.4: Typical radial distribution for the radial distribution.

7.2.1 Radial Distribution Perturbations

The radial distribution is approximated using a piecewise linear function using 20 discretization points, as described in section 5.1,

$$p(x) = \sum_{i=1}^{20} \beta_i \pi_i(x). \tag{7.4}$$

In this way the radial distribution is identified with the vector $\beta = (\beta_1, ..., \beta_{20}) \in \mathbb{R}^{20}$. Each β_i represents the flow of fibers trough a particular point in the spinneret. This flow depends on a stochastic process inherent to the fibers dynamics. In order to estimate the uncertainty of the process, we consider the flows as random variables. As mentioned in section 3.5.1 the flow is non-negative and will be modeled using a uniform distribution in the interval $[0.9\beta_i, 1.1\beta_i]$, i.e. a variance of 10% on the flow. From these assumptions, the parameters of the probability distribution can be obtained.

Assuming such random variables as parameters for our process, it is possible to estimate the shape of the probability distribution for the criterion function. In this case, the homogeneity. Depicted in figure 7.5 is the results of 1000 simulations assuming perturbations only in the radial distribution according to the random variables defined before. The resulting values of homogeneity are classified in 15 intervals running from 0.35 to 0.50. The red line at 0.47 marks the value of the homogeneity for the original simulation, this is, with no perturbation.

I will check the hypothesis that some regions of the radial distribution

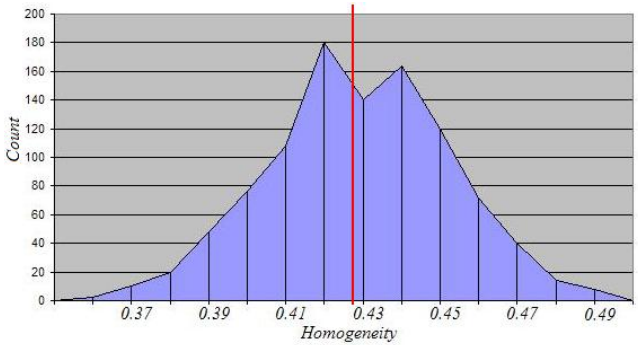


Figure 7.5: Probability distribution for the homogeneity.

have a stronger impact in the sensitivity of the function. To do this, a similar number of simulations are performed but only considering perturbations in a specific subinterval of the radial distribution.

We divide the interval [0, 1] in three subintervals and consider only perturbations in each one of these intervals. The results of the simulations of the 1000 perturbations are depicted in figure 7.6.

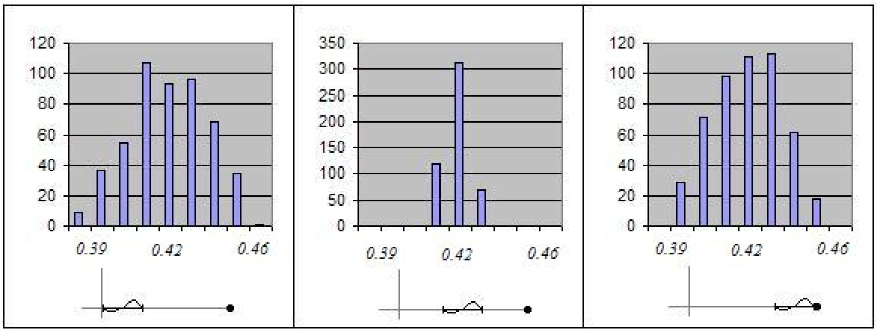


Figure 7.6: Deviations considering only perturbations in 3 subintervals.

It is clear that the first subinterval, the one describing the flow close to the center of rotation, has a stronger impact in the sensitivity of the process with a standard deviation of 0.0169. The middle interval has the less variation with standard deviation of 0.0049. The last interval has also a higher deviation, 0.0148 but not as large as the first interval. These figures suggest also a desire to reduce or cancel the flow of material close to the center of rotation of the radial distribution. Figures may also suggest the reduction of flow in the third interval, but in this case, the flow is required to obtain good values of homogeneity. As will be seen later in this chapter. This simulations also suggest the splines parameterization of the radial distribution that will be presented in page 131.

The choose of a function of that form improves, in many of the cases, the homogeneity and the sensitivity without even performing any optimization. Figure 7.7 compare the results of the perturbations using the old radial distribution from figure 7.4 and a function approximated using splines (page 131). The improvement in both, sensitivity and homogeneity is clear.

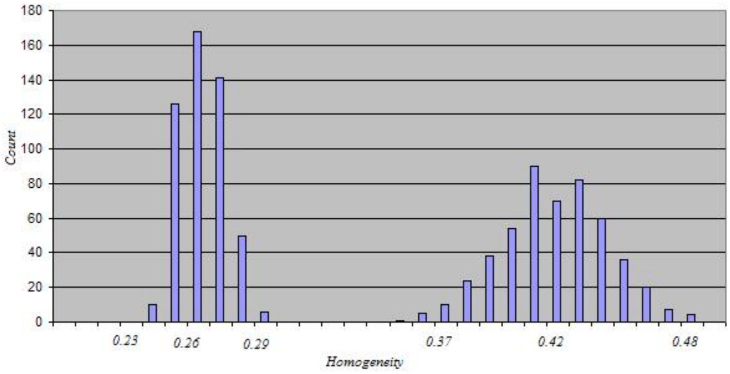


Figure 7.7: Results of perturbations of suggested function (left) and old function (right).

7.2.2 Shifts and Rotational Speed Perturbations

The estimation of the sensitivity of the process with respect to perturbations in the shifts can also be done by performing a Monte Carlo simulation as it was performed for the radial distribution. In this case the shifts are considered as uniform random variables defined in the interval $[\sigma_j - 10°, \sigma_j + 10°]$, where σ_j is the phase shift for the row j.

Nevertheless, there is an additional factor to consider before computing the sensitivity.

The translations considered in the definition of the final density function are directly related with the band shifts, and the band shifts are related with the phase shifts according to 7.1.

If we assume perturbations, say, due to mechanical imperfections, we need to consider perturbations in the phase shifts. But the corresponding effect in the final product depends on the band shifts, which are function of the phase shifts according to 7.1. There are two additional parameters in 7.1, the distance between the rows d_i and the period L_x. Even if we consider the distance d_i as fixed, the period L_x depends strongly in the relation between the rotational speed ω and the speed of the band v_b,

$$L_x = 2\pi \frac{v_b}{\omega}.$$

Perturbations in any of those speeds will have an effect in the band shifts, specially perturbations in ω.

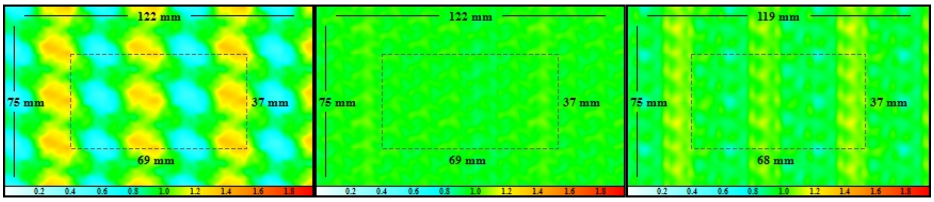


Figure 7.8: Effect of rotational speed and shifts perturbation.

Figure 7.8 depicts in the center the original fleece. On the right the worst output fleece obtained using perturbations of 10° in the shifts, assuming constant rotational speed. On the left the output of the simulation increasing the rotational speed in 1.026% (from 390 to 395 r.p.m.). Clearly, the effects of perturbations in the rotational speed are more significant than those in the phase shifts.

If we consider a perfect model, increasing the rotational speed will improve the homogeneity of the fleece. But the assumption of a perfect model is not a serious option. For those reasons, the shifts sensitivity will consider also a perturbation of ±1% in the rotational speed.

7.2.3 Total Sensitivity

As I mentioned in section 3.5 there is no natural way of considering a single total sensitivity for the model. Adding or computing an average of the shifts sensitivity and the radial distribution sensitivity is like adding apples and oranges, they have different unities.

An alternative is to consider an expected total sensitivity computed by fixing the expected perturbations both in shifts and radial distribution. For the total sensitivity I consider perturbations of 5% in the flow predicted by the radial distribution, perturbations of 5° in the phase shifts and of 1% on the radial distribution. The estimation of the sensitivity should be recomputed if different perturbations are to be considered.

7.3 Radial Distribution Parameterization

In this section I will compare two different parameterizations for the radial distribution. I have already mentioned in page 29, some reasons to consider a parameterized function using only two control points. Such functions are more realistic and require less parameters. This reduction of parameters will reduce the dimension of the optimization problems.

I have considered two different sets of functions during the experiments. The first set contain functions of the form,

$$p^{linear}(t) = \begin{cases} 0 & \text{if } 0 \leq t < \alpha_0 \\ \frac{2(t-\alpha_0)}{(1-\alpha_0)(\alpha_1'-\alpha_0)} & \text{if } \alpha_0 \leq t < \alpha_1' \\ \frac{2(t-1)}{(1-\alpha_0)(\alpha_1'-1)} & \text{if } \alpha_1' \leq t \leq 1 \end{cases} \tag{7.5}$$

for some $0 < \alpha_0 < \alpha_1' < 1$.

These are piecewise linear functions composed of only three segments and described using only two numbers and can be represented as a point $(\alpha_0, \alpha_1') \in \mathbb{R}^2$.

To avoid the restriction $\alpha_0 < \alpha_1'$, we can transform the variable α_1' using the relation $\alpha_1 = \frac{\alpha_1'-\alpha_0}{1-\alpha_0}$, and in this case the point (α_0, α_1) is to be contained in the square $I^2 := [0,1]^2$.

The second set was already described in page 29 and are functions of the form,

$$p^{splines}(t) = \begin{cases} 0 & \text{if } 0 \leq t < a \\ \frac{1-cos(\pi \frac{x-a}{b-a})}{2} & \text{if } a \leq t < b \\ cos(\frac{\pi}{2}\frac{x-a}{b-a}) & \text{if } b \leq t \leq 1 \end{cases} \tag{7.6}$$

This set of functions is also parameterized using two numbers $a, b \in \mathbb{R}$.

In both sets the first number represent the minimum argument where the function is non-zero and the second number locates the peek of the function.

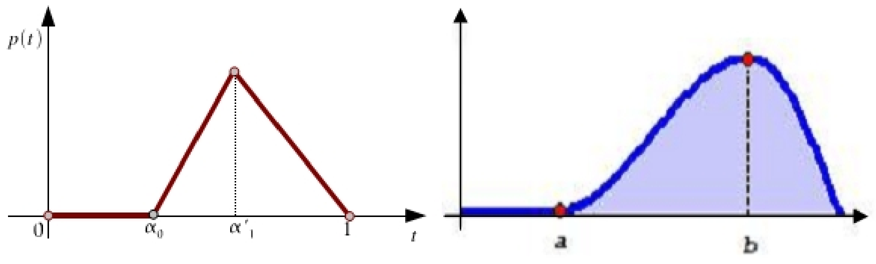


Figure 7.9: Simplified two point radial distributions (piecewise linear and splines).

One element of either set is depicted in figure 7.3. On the left the piecewise linear function and on the right the splines function.

Simulations for 100 random points was considered, in each case considering both kind of radial distributions. Figure 7.10 compares the sensitivity and homogeneity values for those two kind of radial distributions. The plots shows a positive correlation where the piecewise linear radial distribution has a tendency for larger values, i.e. worst sensitivity and homogeneity.

These results suggest that the set of functions defined using splines deliver better results both in terms of fleece uniformity and process stability. We will see in the next section that such functions are almost as good in terms of homogeneity as the original discretization of the radial distribution using 20 parameters considered in the LP formulation on section 5.1.

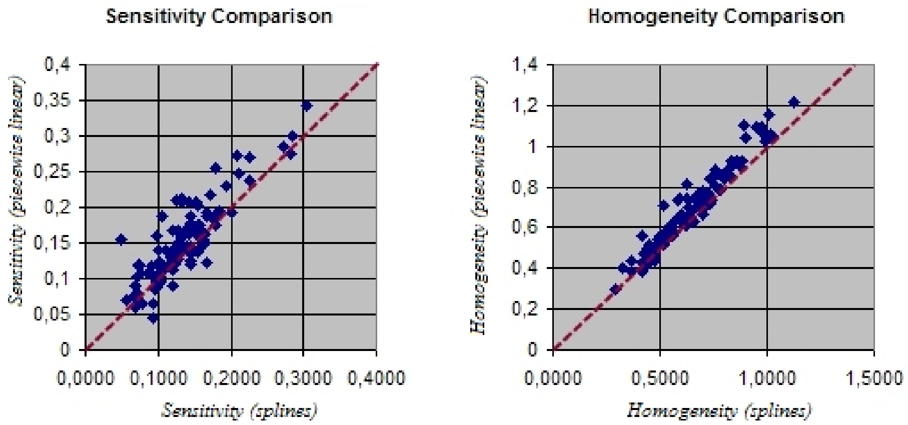


Figure 7.10: Comparing sensitivity and homogeneity for radial distribution parameterizations.

7.4 Optimization of the Fleece Homogeneity

Having the possibility to run numerical simulations for our production process allow us to perform optimization strategies in the analysis. I have explained, in chapter 5 different approaches to the problem of choosing optimal parameters for the machine in order to obtain a fleece with the best quality measurements. In that chapter we considered a single criterion to measure the quality, namely, the homogeneity.

In this section I will present the numerical results of the various scalar optimization problems considered in chapter 5. Due to the variety of the considered problems I do not include several instances for every single kind of problem, specially if similar results where obtained in every instance.

7.4.1 Fixed Shifts

In section 5.1 we model the problem of finding an optimal radial distribution for the process, by keeping fixed shifts, as a linear program.

Problem 11. *Given the input data $A_{i,k,l}$ and the real variables λ, $\alpha_{k,l}$ and*

β_i, *for* $i \in I, k \in K$ *and* $l \in L$

$$\max \quad \lambda \tag{7.7}$$

subject to

$$\alpha_{k,l} \geq \lambda, \qquad \forall k \in K, \forall l \in L \tag{7.8}$$

$$\alpha_{k,l} = \sum_{i \in I} \beta_i \cdot A_{i,k,l}, \qquad \forall k \in K, \forall l \in L \tag{7.9}$$

$$\sum_{i \in I} \beta_i = 1 \tag{7.10}$$

$$\beta_i \geq 0 \qquad \forall i \in I \tag{7.11}$$

As an instance of this problem, I consider a discretization of the domain, using 30 points in each direction and 14 points for the radial distribution. Consider the fixed parameters defined in section 7.1.3 and six rows with shift values $\{0°, 120°, 240°, 0°, 120°, 240°\}$.

The input data $A_{i,k,l}$ represents the flow deposited over a cell (k, l) considering the basis function π_i as the radial distribution. Obtaining the input data, represents running the simulation as many times as the number of basis functions used for the discretization of the radial distribution.

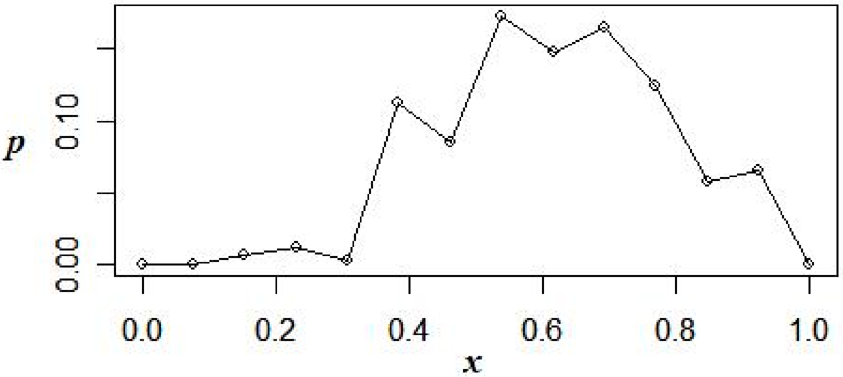


Figure 7.11: Optimal profile obtained from problem 11.

As stated before, this linear problem is solvable by CPLEX. Figure 7.11 shows the optimal radial distribution for the instance. As depicted, the radial distribution suggest a desirable absence of fiber flow close to the center of rotation. This fact is also supported by further experiments and expected from the model, for the reason that points close to the center will move with slow speed, increasing the corresponding flow and generating spots with high density values.

A glance in the shape of this function; in particular the non-smoothness, also suggest the complications of really designing such a radial distribution. We do not expect to design a mechanical system of deflector mirrors to mimic such a function. It is therefore a good reason to suggest the restriction in the family of radial distribution considered for the optimization to those defined in page 29. Smooth functions with no flow close to the zero.

The differences in the output considering the radial distribution with splines and the radial distribution obtained as solution of the LP problem is not distinguished by looking at the fleece output. But the improvement can be observed in the homogeneity criterion. For the optimized splines function the homogeneity value is 0.0772. Which represents a maximal deviation from the average in normalized mass (quantity of fibers). Against a homogeneity value of 0.0442 for the optimal function depicted in figure 7.11. In both cases the fleece is almost uniform, figure 7.12 depicts both outputs.

The lookup table used in figure 7.12 uses grey intensities using gamma correction [51] to emulate a realistic visualization of the fleece. The original color lookup table used in the rest of the visualizations will look completely uniform in this examples.

The output of these two simulations can not be distinguished at sight. The relative difference in the homogeneity is less than 0.5%. This is an additional reason to restrict the search to the family of functions generated by expression 3.10. The functions are more realistic and the obtained results are as good as those obtained with the more general functions.

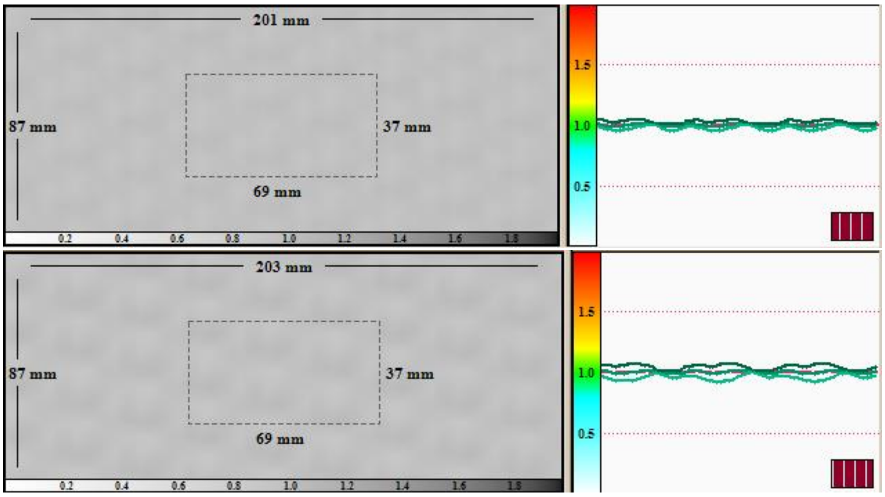


Figure 7.12: Fleeces and projection generated by the optimal radial distribution from figure 7.11 (up) and the best function using splines (down). The first is slightly better.

7.4.2 Fixed Radial Distribution

The problem of optimizing the shifts considering a fixed radial distribution was considered in section 5.2. The problem was modeled as a mixed integer program in the following way.

Problem 12. *Given the input data $A_{j,k,l,s}$ the real variables λ, $\alpha_{k,l}$ and the binary variables $\chi_{j,s}$, with $j \in J, k \in K, l \in L$ and $s \in S$.*

$$\max \quad \lambda \tag{7.12}$$

subject to

$$\alpha_{k,l} \geq \lambda, \qquad \forall k \in K, \forall l \in L \tag{7.13}$$

$$\alpha_{k,l} = \sum_{i \in I} \sum_{j \in J} A_{j,k,l,s} \cdot \chi_{j,s}, \qquad \forall k \in K, \forall l \in L \tag{7.14}$$

$$\sum_{s \in S} \chi_{j,s} = 1, \qquad \forall j \in J \tag{7.15}$$

$$\chi_{j,s} \geq 0, \qquad \forall j \in J, s \in S \tag{7.16}$$

Where the data $A_{j,k,l,s}$ represents the flow of fibers deposited over the spot (k, l) in the fleece discretization assuming that the shift in the j-th row is $\sigma^s := \frac{2\pi s}{N_s}$ and N_s is the number of shifts.

In this case we consider only a discretized subset of possible values as shifts. I consider an instance for 6 shifts and a discretization of 15 degrees for the shifts, i.e. only shifts multiples of 15 are considered as angles.

The problem was given to CPLEX to be solved. After 4 days of computations, CPLEX obtained a gap of 7%. The solution was close to the optimal in the considered discretization sub-domain, but any run of the pattern search method was able to cope with the full continuous domain and obtain better solutions in seconds; in the continuous domain, of course.

Figure 7.13 depicts in red, the best criteria value obtained by CPLEX. The blue interval represents the gap and the green horizontal line, is the best known solution for the instance on the continuous domain.

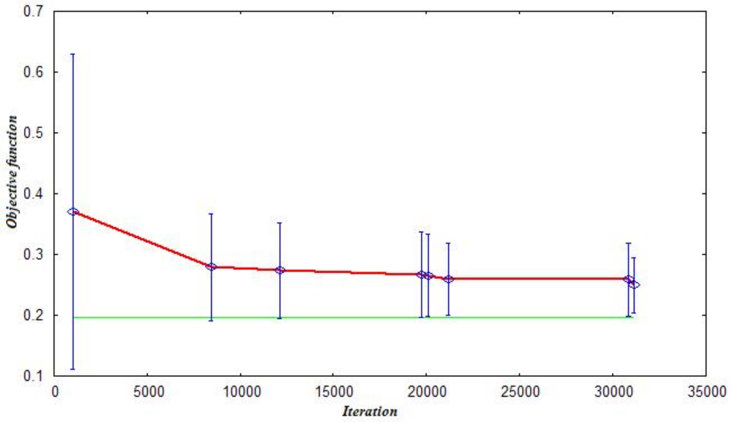


Figure 7.13: Iteration of the MIP using CPLEX and best known solution.

CPLEX consider only the discretized domain and the value of the criterion is not expected to be as good as the solution considering the continuous domain. I will describe the methods for the continuous domain in short. The analysis performed using CPLEX, if not useful for obtaining solutions, give us information if we consider the gap interval.

There is no better solution in the discretized set outside of the gap. The best known solution in the picture was obtained using an heuristic and no global optimality is known for it, in theory, arbitrary better solutions could exist in the continuous domain. But the gap from the MIP analysis tell us that, at least in the discretization, there is no better solution.

7.4.3 Numerical Results of the Pattern Search and Nelder-Mead Algorithms

Section 5.3.1 presented an MIP formulation for the problem of optimizing both shifts and radial distribution simultaneously. This problem is at least as costly as the MIP presented in the previous section. The numerical results are equivalent and I will not present any instance in this section. Instead I present here the results of the general scalar optimization methods, the Pattern Search and the Nelder Mead algorithms.

Consider an instance using 6 shifts the family of radial distribution functions using splines and two control points. In total we have 8 control parameters. The criterion function to be optimized is the homogeneity.

The Pattern Search method was described in section 5.4.2. For the numerical implementation, I use a typical pattern $\mathcal{P}$ defined using the component directions.

$$\mathcal{P} := \{e_i * \eta_i\}_{i=1,\dots,8}$$

Where e_i are the vector in the standard basis of $\mathbb{R}^n$. The coefficients η_i are required. We should avoid using the same step size for every direction, because some parameters are shifts and other are control points for the radial distribution. The coefficients are 1 for the shifts components and 0.3%; as an approximation of 1/360, for the radial distribution control points.

The algorithm recompute the step size according to the performance, and in principle any step can be chosen. I consider an initial size of the pattern of 30 which corresponds to steps of 30 degrees in the shift components and 9% in the control points. If a successful step is performed the Pattern Search method suggest either to increase this step or keep is. Based on numerical experiments I decided to keep the step bounded in 30 degrees, because too large steps are irrelevant in a periodic domain. The size is reduced by a factor of 0.5 in case no improvement is obtained.

The stopping criterion is the reduction of the pattern size to less than 0.5. At this size the angle steps are less than 1/2 of degree and there is no way to be that accurate in the mechanical implementation of shifts in the machine.

Figure 7.14 shows the convergence for 8 random starting points, plotting time (milliseconds) against criterion value (homogeneity). As depicted, in every instance the algorithm converges in less than 3 minutes.

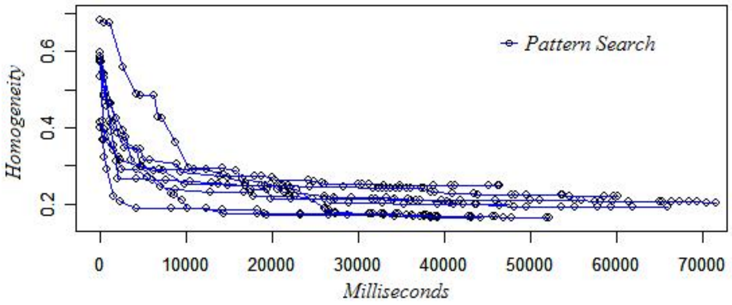


Figure 7.14: Improvement of criterion using Pattern Search.

The same 8 random points are used to perform the Nelder-Mead algorithm. The algorithm requires a simplex as starting data. Using steps of size 30 degrees for the shifts and 9% for the radial distribution points a simplex is first generated. This to allow a comparison with the Pattern Search.

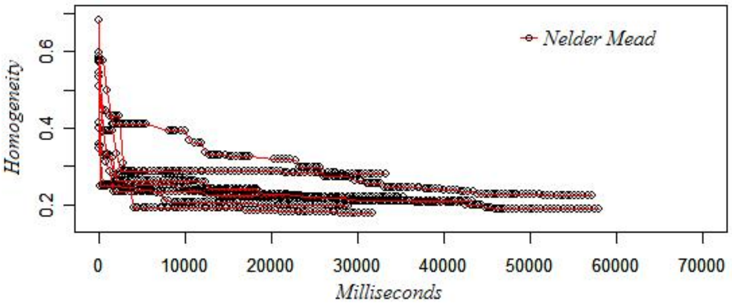


Figure 7.15: Improvement of criterion using Nelder Mead Algorithm.

As mentioned in section 5.4.3, the Nelder-Mead algorithm improves in steps the worst point in the simplex. Depicted in figure 7.15 are the convergence of the 8 random starting points.

In both examples, Pattern Search and Nelder-Mead, the scales are comparable. The tendency is that Nelder-Mead starts improving faster, but Pattern Search usually find slightly better solutions. Figure 7.16 shows a graphic of the time step against the average criterion among the 8 solutions obtained at that time.

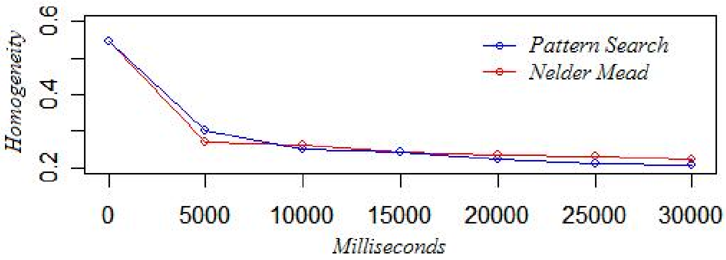


Figure 7.16: Average convergence of Pattern Search vs Nelder-Mead algorithms.

It is difficult to conclude something conclusive from that picture. Interesting is the solutions to where this both approaches converge. From these 8 different starting points, only in one case the Pattern Search and the Nelder-Mead converged to a similar solution. In all other cases distinct solutions where found. This is a signal that the landscape of our function possesses several minima.

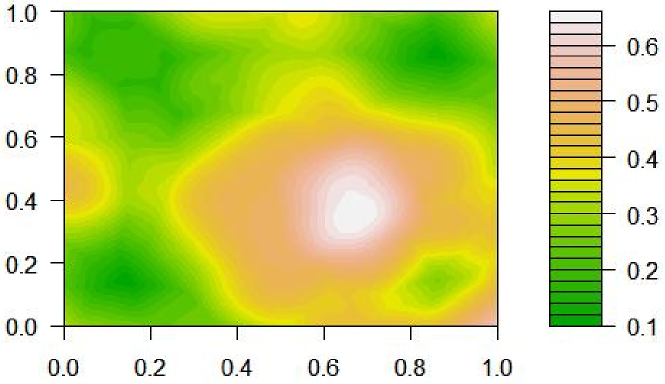


Figure 7.17: Landscape from a two dimensional subspace of the domain.

Figure 7.17 shows the landscape for a two dimensional cut of the domain space. At least 4 regions with local minima are recognizable. But this is only a two dimensional cut of the search space. The plot also suggests that the valleys of minima regions are rather flat.

The convergence properties of both methods guarantees, at most, convergence to a local minimum. Although usually considered a drawback, in the case of searching for Pareto points using a continuation method, this property is important. It was clarified that any local minimum could be used to start tracking the Pareto frontier.

But if we are just trying to find the optimal solution with respect to a single criteria. It is possible to use the idea of the "Basin Hopping" method. Here the strategy is to restart the algorithm and try to jump outside of the attracting basing of the local minimum. This is typically performed by moving from the local minimum and restarting the algorithm. If the new solution obtained is the same in several tries, this suggest that a bigger step should be considered.

7.5 Tracking the Pareto Frontier

For general non linear multicriteria problems, the set of efficient designs is expected to be composed of several disconnected sets. In this section I will present the numerical results of the methods applied to the problem of detecting and tracking patches in the Pareto frontier.

The continuation method starts from a local minima from one of the criterion functions. Local minima can be found using the methods mentioned in the previous section for scalar optimization. In order to obtain further points in the frontier, we need to compose a new function, an aggregating function, which takes into account the rest of the criteria.

For the numerical example, I use a Tchebycheff aggregating function defined as,

$$\|\alpha I f(x)\|_{\infty}.$$

Where f is the vector function whose components are the criteria, α is a vector of weights and I is the identity matrix. In a bicriteria problem, the

Tchebycheff aggregating function is,

$$\min\{\alpha_1 f_1(x), \alpha_2 f_2(x)\}.$$

I consider the problem instance described in the previous sections. Considering again the homogeneity as a criterion, but adding as a second criterion the sensitivity.

The first task is to approximate local minima, considering only a single criterion. In our example this is the homogeneity. Because any local minima can potentially generate a non-dominated path in the Pareto frontier, we are interested on computing as many local minima as we can. For this aim, we consider a population sample of uniform random points in the domain and apply a direct search algorithm, in this case, the Nelder-Mead algorithm. The algorithm will then return local extrema for the considered criterion function. In the previous section I presented the convergence of one application of Nelder-Mead for minimizing the homogeneity.

Because we are interested on capturing several local minima, I consider a simplex of size 5° with respect to the shifts and 1% with respect to the radial distribution points. The simplex diameter is smaller than the one used in the previous section. Starting with this smaller simplex I expect to locate more local minima. Larger simplex will tend to jump above local minima, specially if the attracting basin of the local minimum is small.

If the algorithm converges to a point x_0, the algorithm will be restarted from x_0. If x_0 was in fact a local minima. All successive runs should converge to x_0. Each of these steps will take more time, compared with the convergence shown in the previous section. But as long as the time is bounded in minutes, this pre-computation of solutions is acceptable. Figure 7.18 presents a projection of the local minima obtained. This figure shows us also the high amount of local minima present in the search space. It does not necessarily means that most of those belong to a non dominated patch, but they potentially do.

For the continuation method, I consider the aggregated function

$$\max\{\alpha_1 f_1(x), \alpha_2 f_2(x)\},$$

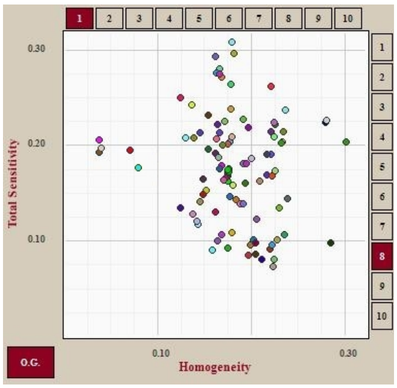


Figure 7.18: Local minima with respect to homogeneity against criterion sensibility.

and parametrize the weights vector as

$$\alpha(t) = (\alpha_1(t), \alpha_2(t)) = (\cos(t), \sin(0)).$$

Figure 7.19 depicts the points obtained from a continuation algorithm, starting from two different local minima. The parameter t of the vector of weights vary from 0° to 90° in steps of 5°.

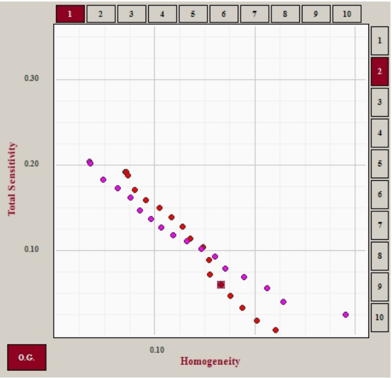


Figure 7.19: Intersection of two locally non-dominated patches.

As expected, from all local minima in figure 7.18, most of the generated locally non-dominated patches where not global non-dominated and did not belong to the Pareto frontier. Starting from 100 local minima, only three patches with non dominated points where obtained for the problem instance being considered. Some of the "best" patches are depicted in figure 7.20.

Points belonging to the same patch are neighbors in the search space and its points depicted with the same color.

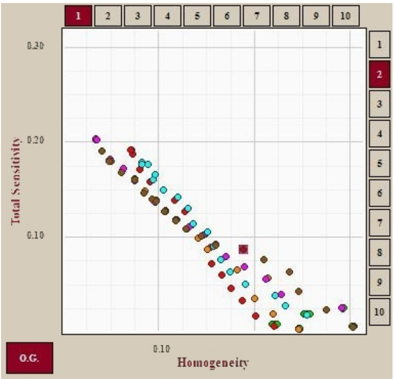


Figure 7.20: Best locally non-dominated subsets obtained from 100 local minima.

Figure 7.21 depicts the image of 5 interpolation points between two points and the approximated frontier using a refined discretization of the parameterization. If such interpolation error is considered too high, further points should be computed and included in the discretization. But it is an expert the one who should decide it.

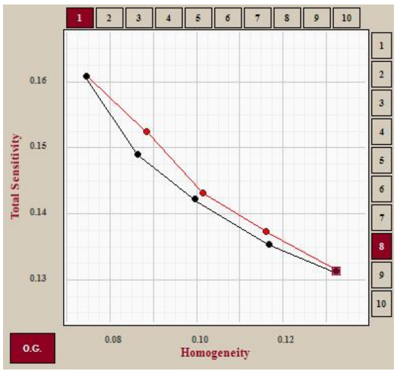


Figure 7.21: A segment of obtained Pareto points (in black) and image of intepolated solutions (in red).

The black points are the images $\{f(x_1), ..., f(x_5)\}$ of five neighbor efficient points obtained using continuation steps. The red points are the images of linear interpolated points between the two extreme points x_1 and x_5. This is, the red points are $\{f(x_1), f(\frac{3}{4}x_1 + \frac{1}{4}x_1), f(\frac{1}{2}x_1 + \frac{1}{2}x_5), f(\frac{1}{4}x_1 + \frac{3}{4}x_5), f(x_5)\}$.

As mentioned before the approximation of the full Pareto frontier for a multicriteria problem with many criterion functions becomes a difficult task. Even the simply storing of points in a file. High memory requirements would prohibit such approximation. In this case, I propose an interactive navigation, where the user starts in a non-dominated point and proposes a direction of improvement. But this step is simply one of the steps in the continuation method presented in the last section and in the next pages. Other proposed strategies in case of high dimensions were already proposed in chapter 6.

This results completes the implementation of the methods proposed in the dissertation. It remains only to resume the conclusions obtained from all this work.

Chapter 8

Conclusions

In a single paragraph the conclusion of this project is that it is indeed possible to develop effective computational tools in the form of a decision support system to assist in the decision making process for non convex and non differentiable problems. As I mentioned in the introduction, such tools exist in the case of linear or convex problems. But several challenges appear when the assumptions of convexity are not available.

We can also conclude that it is possible to approximate the Pareto frontier for non convex multicriteria problems using a discrete set of solutions in a way that permits further interpolation for other points in the frontier.

The inspiration for the dissertation was the design of a spunbond process for the production of artificial fabrics. As it is always the case when a real non typical problem has to be solved, several mathematical topics were required to perform the needed analysis. We developed a general model for the problem of depositing material over a surface. In particular a detailed description of the spunbond process was explained. Some different ways to measure the existence of defects on the fabric were proposed, and to measure in general the non homogeneity of the covering. Special attention was given to study the stability and sensitivity of the production process. We have considered this attribute very important and consider it as an additional criteria to be optimized in the process design.

The appearance of more criteria in the optimization of the process, brought our problem to the framework of multicriteria optimization and decision making. In relation to multiple criteria, we have seen a way to compute successive solutions in the Pareto frontier for non linear problems using a new proposed derivative free method. For this aim, we required direct search methods for scalar problems. We proposed the pattern search and the Nelder-Mead algorithm.

We recognized the necessity of developing tools for assisting the decision making process in the form of a decision support system. Such system was developed using the constructed model and the proposed multicriteria strategies. Some ways of navigating the set of efficient solutions were presented. First a direct exploration of a set of previously computed solutions from a database. As an alternative, the user may choose weight vectors and let the system solve the corresponding aggregated scalar problem. This was possible but considered “not friendly for the user”. The setting of goals in the criteria values, was also proposed as a navigation possibility. A last option was to let the user accept or reject solutions from a small set, and let the system use the mechanics of the Nelder-Mead algorithm to propose new solutions to the user.

The aims of the dissertation were completed. Nevertheless; as is always the case, new and interesting directions can be explored based in the analysis developed. In the modeling of general spunbond processes it may be interesting to consider further criteria, like the anisotropies mentioned in chapter 2. We have proposed a special family of functions to be considered as radial distributions, such functions are realistic, but we left the challenge of design a system of mirrors to mimic such functions. If such design is difficult, it may be required to choose a different family of functions. We also feel that there are more to research in the area of designing tools for decision support systems. This topic was only briefly covered in this project, but it has certainly a huge potential.

Bibliography

[1] Eugene Allgower and Kurt Georg. *Numerical Continuation Methods.* Springer-Verlag, first edition, 1990.

[2] David Anderson. *An Introduction to Management Science: Qualitative Approaches to Decision Making.* Thomson/South-Western, 2000.

[3] Charles Audet and Jr. J. E. Dennis. Analysis of generalized pattern searches. *SIAM Journal on Optimization*, 13(3):889–903, 2002.

[4] Gajanan S. Bhat and Sanjiv R. Malkan. Extruded continuous filament nonwovens: Advances in scientific aspects. *Journal of Applied Polymer Science*, 83:572–585, 2002.

[5] P. Bohl. Über ein in der theorie der säkutaren strungen vorkommendes problem. *Journal reine angew. Math*, (135):189–283, 1909.

[6] Karl-Heinz Borgwardt. *Optimierung Operations Research Spieltheorie: Mathematische Grundlagen.* Birkhäuser, 2001.

[7] I. N. Bronstein, K. A. Samendjajew, G. Musiol, and H. Muehlig. *Taschenbuch der Mathematik.* Verlag Harri Deutsch, fifth edition, 2001.

[8] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[9] David C. Conner, Aaron Greenfield, and Prasad Atkar. Paint deposition modeling for trajectory planning on automotive surfaces. *IEEE Transactions on Automation Science and Engineering*, 2(4):381–392, 2005.

[10] A. L. Custódio and L. N. Vicente. Using sampling and simplex derivatives in pattern search methods. *SIAM Journal on Optimization*, 18(2):537–555, 2007.

[11] Atul Dahiya, M.G. Kamath, and Raghavendra R. Hegde. Nonwovens science and technology ii. http://www.engr.utk.edu/mse/pages/Textiles/. Materials Science Homepage. University of Tennessee.

[12] Anton Deitmar. *A First Course in Harmonic Analysis.* Springer, second edition, 2005.

[13] Peter Deuflhard. *Numerische Mathematik Bd. 1, Eine algorithmisch orientierte Einfhrung.* Walter de Gruyter, third edition, 2002.

[14] European disposables and nonwovens association. http://www.edana.org.

[15] Matthias Ehrgott. *Multicriteria Optimization.* Springer, second edition, 2005.

[16] R. Fessler and D. Hietel. Simulation und verbesserung der vlieslegung, machbarkeitsstudie. Technical report, Fraunhofer ITWM, 2002. Im Auftrag von Johns Manville Europe GmbH.

[17] Saul I. Gass. *Linear Programming: Methods and Applications.* McGraw-Hill, fifth edition, 1982.

[18] Rudolf Grünig and Richard Kün. *Successful Decision Making: A Systematic Approach to Complex Problems.* Springer, 2005.

[19] Peter Hackh. Quality control of artificial fabrics. Research Report 45, Arbeitsgruppe Technomathematik, University of Kaiserslautern, 1990.

[20] Thomas Hanne. Applying multiobjective evolutionary algorithms in industrial projects. Berichte des ITWM 105, Fraunhofer ITWM, 2006.

[21] Claus Hillermeier. *Nonlinear Multiobjective Optimization: a generalized homotopy approach*, volume 135 of *International Series of Numerical Mathematics*. Birkhaeuser, 2001.

[22] ILOG, Inc. Solver cplex. http://www.ilog.com/products/oplstudio/.

[23] Association of the nonwoven fabrics industry. http://www.edana.org.

[24] Eugene Isaacson and Herbert Bishop Keller. *Analyse numerischer Verfahren*. Harri Deutsch, Frankfurt am Main, 1973.

[25] P. Korhonen. Multiple objective programming support. Technical report, International Institute for Applied Systems Analysis, 1998.

[26] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the Nelder-Mead simplex algorithm in low dimensions. *SIAM Journal on Optimization*, 9:112–147, 1998.

[27] Robert Michael Lewis and Virginia Torczon. Pattern search methods for linearly constrained minimization. *SIAM Journal on Optimization*, 10(3):917–941, 2000.

[28] Robert Michael Lewis, Virginia Torczon, and Michael W. Trosset. Why pattern search works. Technical Report TR-98-57, 1998.

[29] Dmitry Luzhansky. Quality control in manufacturing of electrospun nanofiber composites. *International Nonwovens Journal*, 2003.

[30] N. Marheineke and R. Wegener. Fiber dynamics in turbulent flows: General modeling framework. *SIAM Journal on Applied Mathematics*, 66(5):1703–1726, 2006.

[31] K. I. M. McKinnon. Convergence of the Nelder-Mead simplex method to a non-stationary point. *SIAM Journal on Optimization*, 9:148–158, 1999.

[32] Kaisa Miettinen. *Nonlinear Multiobjective Optimization.* Kluwer Academic Publishers, fourth edition, 2004.

[33] Amitava Mitra. *Fundamentals of Quality Control and Improvement.* Prentice Hall, second edition, 1998.

[34] Michael Monz. *Pareto Navigation - Interactive Multiobjective Optimisation and its Application in Radiotherapy Planning.* PhD thesis, Fachbereich Mathematik der Universität Kaiserslautern, 2006.

[35] John Mortimer, editor. *Statistical Process Control.* IFS Publications / Springer-Verlag, 1988.

[36] Larry Nazareth and Paul Tseng. Gilding the lily: A variant of the Nelder-Mead algorithm based on golden section search. *Comput. Optim. Appl.*, 22(1):133–144, 2002.

[37] George Nemhauser and Laurence Wolsey. *Integer and Combinatorial Optimization.* Wiley, 1988.

[38] Helmut Neunzert and Abul Hasan Siddiqi. *Topics in Industrial Mathematics*, chapter How to Judge the Quality of a Nonwoven Fabric, pages 27–42. Applied Optimization. Kluwer Academic Publishers, first edition, 2000.

[39] Panos Pardalos and Mauricio Resende. *Handbook of Applied Optimization.* Oxford, first edition, 2002.

[40] K.L. Poh, M. A. Quaddus, and K. L. Chin. Molp-pc: An integrated decision support environment for multiple objective linear optimization, 1982.

[41] William H. Pound. Real world uniformity measurement in nonwoven coverstock. *International Nonwovens Journal*, 10(1):35–39, 2001.

[42] A. Ravindran, D. Phillips, and J. Solberg. *Operations research: Principles and practice.* John Wiley & Sons, second edition, 1977.

[43] Andrea Saltelli. The critique of modelling and sensitivity analysis in the scientific discourse. an overview of good practices, 2006.

[44] Ole Scheller. Strategies for an optimal irrigation of a garden. Diplomarbeit, Technische Universität Kaiserslautern, 2005.

[45] Oliver Schütze, Alessandro Dell'Aere, and Michael Dellnitz. On continuation methods for the numerical treatment of multi-objective optimization problems. In *Practical Approaches to Multi-Objective Optimization*, number 04461 in Dagstuhl Seminar Proceedings, 2005.

[46] Virginia Torczon. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 7(1):1–25, 1997.

[47] P. Tseng. Fortified-descent simplicial search method: A general approach. *SIAM Journal on Optimization*, 10(1):269–288, 2000.

[48] Andreas Vretblad. *Fourier Analysis and its Applications*. Graduate Texts in Mathematics. Springer, 2003.

[49] Thomas Weise. Global optimization algorithms (theory and application). E-Book, August 2007 http://www.it-weise.de.

[50] Wikipedia. Decision making — wikipedia, the free encyclopedia, 2007. [Online; accessed 4-December-2007].

[51] Wikipedia. Gamma correction — wikipedia, the free encyclopedia, 2007. [Online; accessed 4-December-2007].

[52] Wikipedia. Modern portfolio theory — wikipedia, the free encyclopedia, 2007. [Online; accessed 5-December-2007].

[53] Laurence Wolsey. *Integer Programming*. Wiley, 1998.

[54] Kaiping Zeng. Covariance and spectral density for the characterization of cloudiness of nonwovens. Technical report, Technische Universität Kaiserslautern, 1999.

[55] T. Zolezzi. Condition number theorems in optimization. *SIAM Journal in Optimization*, 14(2), 2003.