Uncertainty Handling and Robust Design in Learning-based Optimization
(A technical report)

Sibghat Ullah, ESR 3

January 8, 2019

Declaration

This technical report is submitted to the supervisory board of EU-ITN project ECOLE (Experienced based optimization, learning to optimize) by ESR 3 who is enrolled in Leiden University. This report reviews the fundamentals in the literature on Uncertainty handling and robust design optimization. The purpose of the report is to provide an overview on existing literature in the area. Another important objective of this report is to investigate the compatibility of learning material especially AM3, AM4, AM5 and AM6 with the ESR's project. This report is ESR's summary on the investigations and work done by other researchers as specified in the bibliography and acknowledgment. This report contains 31 pages including tables, equations and figures. An important thing to note is that the ESR's project comprises of three phases namely Robust Design Optimization (see WP 2.3), Big Data Analytics with Robust Representation (see WP 3.2) and Robust and Online Learning (see WP 3.4) as presented in ECOLE proposal. The focus of this report however will be on the first phase, since the second and third phases have to be investigated during ESR's secondment at NEC-laboratories Europe and Honda-research-institute Europe respectively. Furthermore, Section 2 (State-of-the-art) is optional . This is to comply with the word-count limit (2700-3300 words) as described in ECOLE training plan.

Sibghat Ullah
January 8, 2019

Acknowledgements

I would first like to thank my Promoter Professor Thomas Bäck. His guidance, vision and trust meant a great deal to me. I would also like to thank my supervisor Dr. Hao Wang for his valuable advice, continuous overseeing and time throughout the process of writing this report. Lastly, I feel grateful to ESR 4 and 5 for their advice on the report. Thank you all.

Sibghat Ullah
January 8, 2019

Abstract

Uncertainties and noise comprise one of the most challenging area in the optimization literature. They're encountered frequently in real-world optimization problems. Due to different reasons, different types of uncertainties can arise. This report focuses on some of the most promising work in the field of uncertainty handling and robust design optimization. More specifically, We formulate the existing understanding of uncertainty handling and robust design optimization. We state different types of uncertainty modelling techniques which are common in the optimization literature. Next, we will report some basic methodologies to minimize the effects of uncertainties and the approaches to find robust design optima. The relationship of this research project with AM3, AM4,AM5 and AM6 will also be discussed in the final section of the document. Finally, some open research questions and future research line would be proposed.

Contents

Chapter 1

Introduction

In this chapter, I discuss the basics necessary to understand the report. Uncertainty and noise are discussed inter-changeably to describe the same concept i.e. unexpected changes in the optimization throughout the first chapter. In section 1.1 , we discuss black-box optimization. Next, we describe the various sources of uncertainty in the black-box optimization (See figure 1.2). We also move on to draw an informal understanding of uncertainty vs noise and Aleatory vs Epistemic uncertainty types. Finally, a brief summary of methodologies to represent various uncertainty types are discussed in section 1.4. The first chapter of this report can be though of as a rough summary of [1] (Section 1-3) and that of the problem definition.

1.1 Black-box Optimization

Classical view on optimization concerns the best solution among a set of feasible solutions where the best refers to some criterion. Throughout this report, we focus on the scenario of black-box optimization in which we believe to have no or limited knowledge regarding the internal dynamics of the system. An example of a black-box system is provided in figure 1.1.

Here, we're not assuming any particularities on $\mathbf{x}$ and $\mathbf{y}$. Thus, the black-box optimization can be thought of finding the configuration of input setting $\mathbf{x}$ that will yield the best possible value of $\mathbf{y}$. The figure 1.1 does not explain the constraints on $\mathbf{x}$. Furthermore, the real-world optimization problems can have multiple conflicting objective functions to be optimized simultaneously. Hence, a more formal understanding is required to grasp the concept of black-

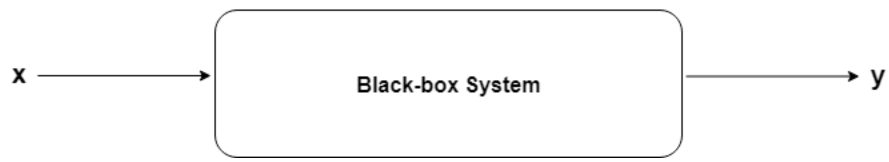

Figure 1.1: An example of a generalized black-box model of a system with some arbitrary input and some arbitrary output.

box optimization.

A constrained, multi-objective black-box optimization problem can be formulated as optimizing a set of functions **F** within a search space **X** while satisfying a set of constraint functions **G**.

$$\mathbf{F} = (f_1, f_2, ..., f_d) \tag{1.1}$$

$$\mathbf{G} = (g_1, g_2, ..., g_k) \tag{1.2}$$

Each objective function f_i in (1.1) is a mapping from an arbitrary input space $\mathbf{X}$ to a scalar in $\mathbb{R}$. Similarly, each constraint function g_z in (1.2) is a mapping from an arbitrary input space $\mathbf{X}$ to a scalar in $\mathbb{R}$.

Multi-objective optimization (MOO) has been a focus of research in the last decade or so. Many different algorithms have been proposed to solve multi-objective optimization problems with most notable work presented in [2-4]. A latest survey on multi-objective optimization is presented in [5]. However, for the remainder of this report, we will be focusing on single-objective unconstrained optimization problems in which $\mathbf{G} = \emptyset$ and $\mathbf{F}$ has only one element. This is since the existing work largely focuses on the most trivial cases namely the unconstrained single-objective optimization problem. However, it must be stated that the goal of the ESR's project involves extending the conceptual understanding of existing literature to the cases of multi-objective constrained optimization problems.

1.2 Uncertainty types

Too often in the optimization literature, theoretical understanding hardly describes the complete picture of real-world issues. Out of such unexplained issues, we only focus on uncertainty and noise. Here, Uncertainty and noise

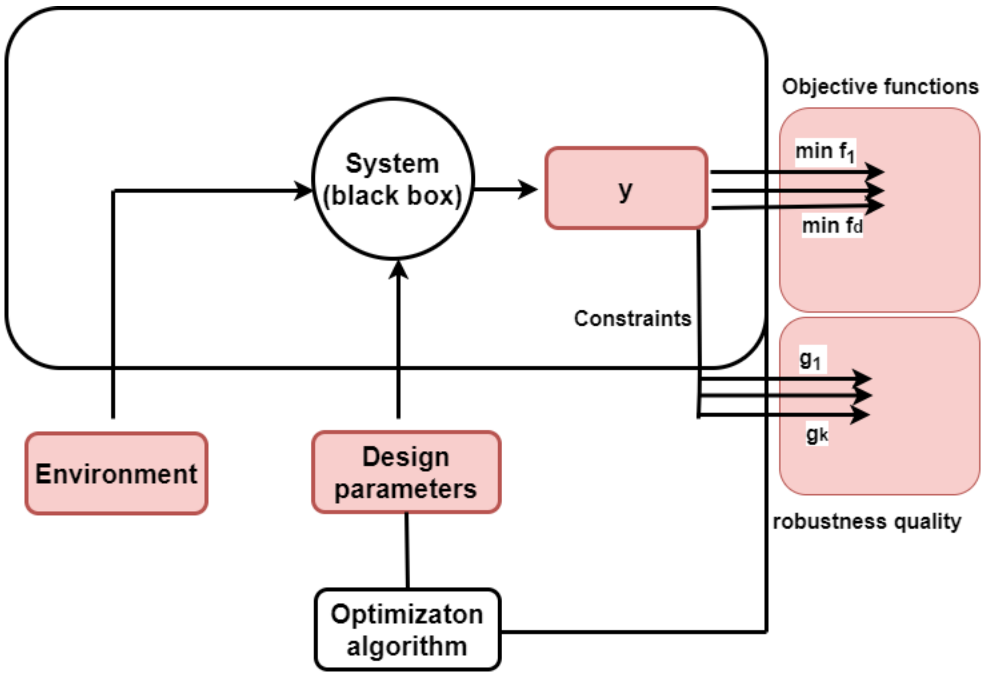

Figure 1.2: An example of a black-box model of a system with color representing areas of the system where noise can arise.

refer to the same concept i.e. unexpected changes in the system. These changes can happen in the input, output or within the system itself. We can refer to these changes as type A, type B, type C, type D and type E uncertainties as taken from [1] and [7]. It is important to understand that type D and E uncertainties are considered to be the same in [7].

- Type A uncertainties refers to the fact that the design variables cannot be controlled with unlimited precision. This can happen due to the economies of manufacturing or the limitation of the state-of-the-art. The effect of the imprecision of the design variables on the output of an arbitrary objective function and a constraint function within a black-box system is formulated in (1.3) and (1.4) respectively.

$$\tilde{f}_i(x) = f_i(x + \delta_x) \quad , i = 1, ..., d \tag{1.3}$$

$$\tilde{g}_z(x) = g_z(x + \delta_x) \quad , z = 1, ..., k \tag{1.4}$$

- Type B uncertainties refers to the fact that the operational (or environmental) conditions fluctuate or are known only to a certain extent. This means that our existing knowledge about the system is limited and as a consequence, we're not informed completely about the search space. In the classical optimization as depicted in figure 1.1, environmental conditions are treated as constants however, in practice that's hardly the case. We thus extend the output dependency to the environmental variables set $\mathbf{C}$ where $\alpha \in \mathbf{C}$ represents an individual variable. The effect of such variables on the output and the constraints is represented in (1.5) and (1.6).

$$\tilde{f}_i(x, \alpha) = f_i(x, \alpha + \delta_\alpha), i = 1, ..., d \tag{1.5}$$

$$\tilde{g}_z(x, \alpha) = g_z(x, \alpha + \delta_\alpha), z = 1, ..., k \tag{1.6}$$

- Type C uncertainties refers to the fact that the output of the real-world system or of the (simulation) model is noisy. Approximation models replace the real-world system within the optimization loop. This type of uncertainty is further elaborated in [6]. Noise in the output can be modeled within formulations of the objective functions as:

$$\tilde{f}_i(x) = f_i(x) + h_{f_i}(x), i = 1, ..., d \tag{1.7}$$

$$\tilde{g}_z(x) = g_z(x) + h_{g_z}(x), z = 1, ..., k \tag{1.8}$$

where $h_{f_i}(x)$ and $h_{g_z}(x)$ are random variables representing the propagation of the uncertainty to the objective and constraint functions.

- Type D uncertainties refers to the vagueness in the preference of objective functions **F** especially when there's a trade-off between the objectives.
- Type E uncertainties refers to the vagueness in the set of constraint functions **G**. This type of uncertainties can be modelled using fuzzy logic [13-15].

In [8], type A and B uncertainties are considered identical and are thought to be related with the concept of **Sensitivity Robustness**. Type D and E are often considered to be the same in the sense that they do not directly affect the output of the black-box system in figure 1.1. Instead, they influence the search space **X**. Type E uncertainty is also related to another important concept known as reliability based robustness. This means that the black-box system must also respect the neighbourhood of the feasible search-space since the constraint boundary is subject to changes. Examples of the reliability based design are discussed at great length in [9-12].

1.3 Uncertainty vs Noise

A clear distinction between uncertainty and noise has not been made in the literature. Thus, we have to rely on an informal understanding of both concepts. The same is also true for Aleatory and Epistemic uncertainties [17]. The most common idea for aleatory uncertainty is that it is fundamentally irreducible, completely random and almost certainly unavoidable. Epistemic uncertainty on the other hand is in principle due to the lack of understanding, knowledge or information on the optimization problem. The effect of this kind of uncertainty can be minimized by representing the optimization problem in another way or with more feedback from the domain expert or with more data. Thus, although it is an integral part of the problem, the designer can solve this type of uncertainty. We can also refer the term uncertainty to the aleatory type and noise to the epistemic type although it must be stated that it is by no means a formal and clear understanding. This view is also popular in engineering applications [16-17].

Another important classification within epistemic uncertainty i.e. noise is if the noise is stationary or non-stationary. Stationary noise is thought to be in the system when its distribution remains the same throughout the time/space domain. Non-stationary noise is when the noise or uncertainty distribution changes with time/space domain. This distinction is borrowed from the existing understanding in Signal-Processing and Time-Series analysis.

1.4 Modelling Uncertainty

Thus far, we've discussed the basics of uncertainty and noise i.e. unexpected changes in different parts of the black-box system as represented in figure (1.2). The next interesting question was to categorize the uncertainty in to different types for a clearer understanding. With that, we also defined an informal distinction between uncertainty and noise and between Aleatory and epistemic uncertainties. Now that we've more understanding about uncertainty and optimization, the next logical question is how to model them. In this section, we'll be briefly discussing this question.

Mathematically, uncertainty can be represented **Possibilistically**, **Deterministically** and **Probabilistically**. This simply means that using Fuzzy logic based on Dempster Shafer Theory of Evidence, we can model some of the above uncertainty types (A-E). This involves representing uncertainty type (A-E) by membership functions whose value ranges from 0-1. Example of such work is presented in [13-15]. Another way of representing uncertainty is Deterministically. This means that we assume the domain in which uncertainties can arise are already known and fixed. However, the corresponding probabilities for such neighbourhood are unknown. Examples of such work are presented in [10] and [18-22]. A third way of modelling uncertainty in above scenario is by building a probabilistic model around the part of the system where uncertainty can arise. Here, we assume to know about the domain of uncertainty i.e. neighbourhood of search space and the probability distribution around that neighbourhood. Thus, we can see the effect of such uncertainties on the optimization problem almost directly. This line of work has been discussed in [1],[7],[12] and [23-25].

While we've informally discussed the idea of how to model uncertainty using three main approaches, the work discussed here is merely oversimplification of much more detailed theoretical frame works. As such, the reader is directed to visit [7] (see Section 3) for a thorough understanding. In the next chapter, we'll summarize the existing literature on uncertainty handling.

Chapter 2

State of the Art

In this chapter, We discuss the state of the art in Robust Optimization for the scenarios limited to unconstrained single-objective real parameter optimization. The work is again, a brief summary of [1] (Section 5-8) and [7]. Section 2.1 describes the goal of noisy objective functions. The next section reports some theoretical and empirical investigations for the optimization of noisy objective functions. Finally, another important case i.e. finding robust optima is discussed.

2.1 Optimization and goals of noisy objective functions

A very common case in the robust optimization corresponds to the optimization of a function which produces a noisy output. This is referred to as the optimization of noisy objective functions in [1] (Section 5). Such functions can be represented as in (2.1).

$$\tilde{f}(x) = f(x) + h(x) \tag{2.1}$$

,

In this relationship, the objective function is represented by some additive noise $h(x)$ in combination with a noise free computation of $f(x)$. The uncertainty is unbiased if the expectation $\mathbf{E}[h(x)] = 0$ for all $x \in R^n$.

The goals of the optimization for a scenario discussed in (2.1) vary considerably. One of the common choices is the optimization of $f(x)$ without considering the effects of uncertainty $h(x)$ [26-27]. This might seem naive but investigations in [1] prove that it is logical in a variety of situations. For example, if we're discussing a slight change in measurement in which the measurement error is not a universally occurring phenomenon. This is since the practical goal of robust optimization does not always disagree with the goal of classical optimization. We can denote the goal of optimization with a robust function aka as the effective function in literature. Thus, the above mentioned scenario can be represented in (2.2).

$$f_R(x) = f(x) \tag{2.2}$$

An alternative robust function can be the optimization of the expected value of objective function. This is formalized in (2.3). This robust formalization is considerably important if the noise or the uncertainty is an intrinsic part of the optimization.

$$f_R(x) = \mathbf{E}[\tilde{f}(x)] \tag{2.3}$$

An interesting alternative described in [28] is to consider it a bi-objective optimization problem with the optimization of expectation in (2.3) and minimization of the variance. Thus, the resulting problem has two objective functions and both are to be optimized simultaneously. This is formalized in relationship in (2.4-2.5). An interesting future research line is to investigate this at length.

$$f_R^1(x) = \mathbf{E}[\tilde{f}(x)] \tag{2.4}$$

$$f_R^2(x) = \mathbf{V}_{ar}[\tilde{f}(x)] \tag{2.5}$$

Several other robust functions have been defined in the literature however we omit those here. The reader is steered to [1] (Section 5.1) and [7] (Section 3.2) for further details. We move on to the next section to describe the basic techniques for the optimization of objective functions with uncertain outputs.

2.2 Basic Noise handling techniques

It has been reported in several studies [1] and [29-32] that Evolutionary Algorithms in general are fairly robust against noisy output. This is intuitive since a population based solution is better than a single solution in many cases. Even surprisingly, random noise can have great effect on the generalization of an algorithm in Machine Learning where it sometimes take the name of **Artificial Feature Noising** [33]. Hence, for the sake of this report, we assume that Evolutionary Algorithms in general are already superior to other methodologies to handle basic uncertainty and that uncertainty is not always bad as initially considered.

One of the most common techniques to reduce the effect of noise i.e. systematic uncertainty for the goal represented in (2.3) is Resampling aka Explicit Averaging. This simply refers to approximate the expectation in (2.3) with sample mean. We can formalize our understanding of Resampling in (2.6).

$$\hat{f}_R(x) = \frac{1}{L}\sum_{j=1}^{L}\tilde{f}(x) \tag{2.6}$$

According to the law of large numbers, the sample mean $\frac{1}{L}\sum_{j=1}^{L}\tilde{f}(x)$ in (2.6) is an unbiased estimator of expectation $\mathbf{E}[\tilde{f}(x)]$ in (2.3). This means (2.6) will converge to (2.3) and as a consequence will reduce the noise. An obvious downside of this approach is the computational effort by a factor of L.

An alternative approach in Evolutionary Algorithms is to increase the population size aka Implicit averaging. It has been reported in [34-35] that the so called selection step in Genetic Algorithms will not be affected by the noise for infinite population size. For Evolution Strategies, this is more complicated as reported in [1] and [26] and does not always result in noise reduction. As taken from [26] , increasing μ in $(\mu, \lambda) - ES$ will result in lower convergence speed. The results reported in [1] (Section 5.3.2) comparing $(\mu/\mu_1, \lambda) - ES$ with CMA$-ES$ [36] in the presence of noise also concludes that increasing μ (by fixing λ) or vice versa decreases the convergence speed of $(\mu/\mu_1, \lambda) - ES$ drastically although it increases the convergence accuracy. For the case of CMA$-ES$, fixing ($\mu \approx \lambda$) will decrease the noise level with no drastic effect on convergence speed. Another investigation in [1] (Section 5.3.3) reports that Resampling i.e. explicit averaging is more effective in noise reduction for $(\mu/\mu_1, \lambda) - ES$ while Implicit averaging performs better in the case of CMA$-ES$. Furthermore, CMA$-ES$ performs better than $(\mu/\mu_1, \lambda) - ES$ for controlled noise experiments conducted in [1] (Section 5.3.3). It is important to note that this can very well be the bases of further empirical investigation for ESR 3 with a special focus on the most complex noise systems generally found in industrial applications.

Another alternate technique using rescaled mutations [29] and [37-38] which uses large mutations in the selection process and smaller after that intuitively seems nice considering that it does not deteriorate convergence speed. However, practically it does not reduce noise significantly [1] (Section 5.3.4) and hence has not been widely employed in applications. Another theoretically sound technique called Thresholding [39] also fails to practically work in most situations.

2.3 Advanced Noise handling techniques

Schemes discussed in the previous section try to minimize the effect of noise. These techniques try to improve the convergence accuracy of Evolutionary Algorithms at the cost of convergence speed. Furthermore, they don't completely negate the effect of noise. As such, advanced noise handling techniques are presented in this section that theoretically seem superior. Such techniques can be divided in two parts namely the Adaptive Averaging techniques and Meta-Modelling techniques.

Adaptive Averaging techniques try to adapt the level of noise in the optimization problem in a way that there is no waste of computational resources. Two of such techniques are called Duration Scheduling and Sample Allocation [40-41] and [1] (Section 5.4.1). Both of these techniques have limitation in that they assume the noise is Stationary (Section 1.3) and Gaussian. Another technique relying on Statistical hypothesis testing is discussed in [42-43] and 1 (Section 5.4.2). The practical results in [1] , [42-43] suggest that this technique is also not suited for complex noise situations. Yet another alternate technique called Partial order based adaptive averaging is also discussed in [1] (Section 5.4.3) and [44]. It is further reported in [1] that this scheme shows promising results on a variety of noise handling situations but needs to be further investigated in details. As such, it will be an interesting opportunity for ESR to investigate it further. Other notable work in this area includes Selection Through Racing [45],[1] (Section 5.4.4) , Rank-change based uncertainty measures [27] , [1] (Section 5.4.5) and Rank-Inversions based uncertainty measures [42] ,[1] (Section 5.4.6).

Meta-Modelling [46-48] makes another class of techniques to handle uncertainty and noise in the literature. The aim is to use the previous information available on the fitness measurements of the noisy objective functions and design an optimization model using this information. They can be further divided in different classes including Memory-based Fitness estimation [46-47] ,[1] (Section 5.5.1) and Local Regression based Fitness Estimation [48] , [1] (Section 5.5.2). Meta-modelling techniques have not been studied extensively in the literature for noise handling and provide a great opportunity for the concerned ESR to further investigate the existing methodologies and propose new ones. In the next section, we discuss Evolution Strategies to find Robust Optima.

2.4 Evolution Strategies for Finding Robust Optima

Robust optimization refers to many scenarios. One of those scenarios namely the optimization of noisy objective functions has been discussed in this chapter. Yet another similar scenario namely finding the robust optima has not been discussed. Thus, this section will focus on the case of finding robust optima. The difference between the optimization of the noisy objective functions as discussed previously in this chapter and finding robust optima is indeed very minimal. In fact, authors in [7] do not differentiate clearly between the two cases as they give it a general name of Robust Optimization. In [1] however, it has been classified and studied differently. One of the possible distinction between these two scenarios is that the former namely the optimization of noisy objective functions assumes the presence of noise within the system itself while evaluating the objective functions. The latter case namely finding the robust optima specifically deals with noise in the input. Having said this, we do not think that formally there is any clear distinction between these two cases and both can be generalized as Robust Optimization as in [7]. In the remainder of this chapter, we will however try to summarize the techniques in [1] (Section 7) and [7] as we feel both are overlapping to a great extent and almost always refer to the same concept.

In the case of a single-objective black-box real parameter optimization, finding robust optima refers to the optimization of objective function which is represented in (2.7) while validating the constraints in (2.8). The relationship in (2.7-2.8) are inspired from (1.3-1.6) i.e. changes in the input and design variables are both merged together in (2.7-2.8).

$$opt \to f_i(x + \delta_x) \qquad i = 1, ..., d \tag{2.7}$$

$$g_z(x + \delta_x) \geq 0 \qquad z = 1, ..., k \tag{2.8}$$

It has been mentioned in [1] and [7] that a neighbourhood $\eta_{\mathbf{x}}$ of the possible set of disturbances in the input **x** can be defined. This neighbourhood will help us examine the behaviour of the changes in the input. Once we are able to see the implications of the corresponding neighbourhood, we can define a measure of Robustness. Indeed it has been argued in the literature that there are many such measures. We will however concern ourselves to **expected solution quality** [49-54] , **worst solution quality** [55-56] , **Threshold acceptance probability** [57] , **Performance variance** [58-59] and **Sensitivity region** [60] which are summarized and compared in [1] (Section 7.1.4). Although we will not be discussing the robustness measures for constraint functions explicitly in this report , the distinction between soft and hard constraints can be made [61]. Similarly, measures based on **Probability Theory**, **Virtual Bounds** [62] and **Transformation of objective function** are also a mainstay in literature.

Now that we understand the essence of Robust Optima and methodologies to represent robustness measures, the next logical question is how to actually use these robustness measures to find the robust optima. This refers to the algorithmic schemes to find the robust optima assuming there's a fixed robustness measure e.g. **expected solution quality** [49-54] agreed upon. In [1] (Section 7.2), such techniques have been classified with much granularity as opposed to [7] (Section 4) which focuses with less detailed classification mechanism. The work in [1] focuses on Evolutionary Algorithms with two specific implementations $(\mu/\mu_1, \lambda) - ES$ and CMA$-ES$. As such, the techniques to find Robust Optima in [1l labelled as **Myopic approaches**, **Single and Multi-evaluation approaches**, **Adaptive Averaging approaches**, **Archive based approaches** and **Meta-modeling approaches** are discussed in the presence of $(\mu/\mu_1, \lambda) - ES$ and CMA$-ES$. A description of these techniques is provided below.

- **Myopic Approach** refers to the intuition that Evolutionary Algorithms are in general fairly robust against uncertainty. This point has previously been discussed in the case of optimization for noisy objective functions (Section 2.2) and also in [1] and [29-33]. In [1] (Section

7.2.1), the experimental results conclude that in some trivial cases, Evolutionary Algorithms do not need any extra computation to find Robust Optima.

- **Single and Multi-Evaluation** methods refer to the Monte Carlo techniques to approximate the **expected solution quality** with sample mean. This is in accordance with the discussion in section 2.2 for the optimization of noisy objective functions [63]. These method take the name of **Single Evaluation Mode** and **Multi Evaluation Mode** corresponding to the number of samples for evaluation of **expected solution quality**.

  $$\hat{f_R}(\mathbf{x}) = \frac{1}{L}\sum_{j=1}^{L} f(\mathbf{x}+h_i) \quad , \quad h_i \sim \delta \tag{2.9}$$

  (2.9) approximates the expected solution quality for a solution $\mathbf{x}$. The experimental results in [1] (Section 7.2.2-7.2.4) report that these methods produce decent results but in general are not very promising to find Robust Optima using Evolutionary Algorithms.

- **Adaptive Averaging Techniques** taken from Section 2.3 in the context of noisy objective functions can also be adapted to find Robust Optima. Adaptive Averaging seems intuitively more powerful than explicit averaging since it adapts the computational resources to the need of noise level [64-65]. In [65], adaptation of Evolution Strategies has been classified as Adaptive Averaging Technique which shows promising results [65] and [1] (Section 7.2.6).

- **Archiving** [52] is another methodology to approximate the **expected solution quality** based on the weighted mean of the quality of the solution at a point with the measures of solution qualities of its neighbours. It has been concluded in [1] (Section 7.2.7) with empirical investigations that Archive based approach is impressive.

- **Meta-Modeling Techniques** [25] are very similar to Archive based approaches. The aim is to use a rough local model as a tool to approximate the behaviour of the objective function and thus fulfill the goal of Robust Optimization defined by a measure. A thorough empirical investigation in [1] (Section 7.2.8) reports that Meta-Modeling is one of the very promising areas within Evolution Strategies to find Robust Optima.

2.5 Robust Optima beyond Evolution Strategies

In [7], several techniques to find Robust Optima are reviewed. Some of these techniques include **Robust Optimization with Mathematical Programming** [66-68] , **Simulation Optimization** [6], **Reliability based Optimization** [10-12] , **Sensitivity Robustness approach** [69] , **Monte-Carlo Approaches** [10] , **Meta-Model Approaches** [23], [25] and **Evolutionary Algorithms** [1] (Section 4). It has been concluded in [7] that direct search methods such as **Evolution Strategies** , **Monte-Carlo Approaches** and **Meta-Modeling techniques** are intuitively a good start to solve the problem of Robust Optimization. In the next chapter, we will discuss the compatibility of learning modules namely AM3, AM4, AM5 and AM6 with ESR's project with focus on WP 2.3.

Chapter 3

Research and Future

In this chapter, We will present brief taxonomy on the ESR's project with focus on WP 2.3. This is done in Table 3.1 which summarizes Chapter 1 and 2. Next, we will discuss the compatibility of ESR's project (WP 2.3) with AM3, AM4, AM5 and AM6. In the final section of this chapter, we will propose some open questions and future research line.

3.1 Uncertainty and Robust Optimization

Table 3.1 reports major uncertainty types (Section 1.2) alongside their behaviour (Section 1.3) , their mathematical representation (Section 1.4) and algorithmic methodologies employed for Robust Optimization. The first column refers to the type of Uncertainty i.e. A-E. The second column describes the behaviour of such uncertainties and classifies them into stationary or non-stationary. The third column also describes the behaviour of uncertainties on much more philosophical terms i.e. Aleatory and Epistemic as denoted by A and E respectively. The fourth column reads the mathematical representation used to describe such uncertainties i.e. Possibilistically, Deterministically and Probabilistically as represented by PS, DE and PR correspondingly. The second last column explains the major algorithmic methodologies employed for Robust Optimization in the presence of uncertainty i.e. Mathematical Programming and Direct Search Methods as represented by MP and DSM respectively. Finally, the last column reports some famous work to deal with each uncertainty type. In the next section, we will evaluate the compatibility of AM3, AM4, AM5 and AM6.

Type	S/NS	A/E	Representation	Major Techniques	References
A	Both	**A**,E	PS,DE, **PR**	MP,**DSM**	[1][7][58][9][23][65]
B	Both	A,E	**PS**,**DE**, **PR**	MP, **DSM**	[1][7][23][65][29-32][17]
C	Both	A,E	**PS**,**DE**, **PR**	DSM	[9][23][1][7][17]
D	S	E	PS,**DE**	**MP** , DSM	[24][1][7][16][13-17]
E	S	E	PS,**DE**	**MP** , DSM	[1][7][17][21][23]

Table 3.1: A brief taxonomy of Major Uncertainty types along side their mathematical representations and algorithmic schemes to minimize their effects.

3.2 Learning Material for ESR's Project

In this section, we evaluate the relevance of ESR's project (WP 2.3 only) with the learning material provided. The learning material discussed in this section comprises of AM3 **Machine Learning**, AM4 **Multiple-Criteria Optimization and Decision Analysis** , AM5 **Advances in Data Mining** and AM6 the **Evolutionary Algorithms**. As it can been seen from Chapter 1 and 2, AM5 is not directly linked to the ESR's project (WP 2.3) however it strongly relates to WP 3.2. Since concerned report focuses on WP 2.3, such relation is not investigated. However, it is clear that WP 2.3 is strongly connected to AM6 **Evolutionary Algorithms** [1-5][7][18][20][25-31][34-44][48-57]. This is also verified by Table 3.1 which includes DSM i.e. Direct Search Methods to handle almost all major types of uncertainty. The empirical investigations in [1] and [7] also conclude that Evolutionary Algorithms are indeed a reliable source to handle uncertainty and to find robust optima. This provides ESR with an interesting opportunity to extend the literature of **Evolutionary Algorithms for Robust Optimization** from trivial cases i.e. unconstrained single-objective real parameter optimization to more complex cases i.e. Multi-objective optimization. This will involve AM4 **Multiple-Criteria Optimization and Decision Analysis** [2-5] [19-20]. As such, it can be seen that AM4 and AM6 are strongly connected to ESR's future goals. Furthermore, investigations in [1],[6-7],[12],[17],[23-25],[33] and [46-47] report Meta-Modeling techniques as a decent algorithmic scheme to find Robust Optima. From that, it can be safely deduced that supervised Machine Learning (strongly related to AM3 **Machine Learning**) e.g. Polynomial Regression, Kriging, Neural Networks and Support Vector Machines might play an important role in ESR's investigations on WP 2.3. From the discussion in Chapter 1,2 and Table 3.1, it is thus safe to conclude that AM3, AM4 and AM6 are strongly related to the goals of the project and ESR must enhance the skill set in those areas. In the next section, we propose some interesting research questions.

3.3 Research Questions

Based on the observations in [1] and [7], we propose some interesting research questions that can be coincided with the ESR's project goals. It is logical to assume that these questions are merely the tip of the iceberg and a series of more advanced questions and research line can be formulated. Some of these open questions are stated below.

- Can the problem of finding optimal robust design be defined as multiple criteria decision problem? if so, in which situations? under what assumptions? Are the assumptions flexible? Can we model desired robustness measure as an additional constraint or objective? [1-5] [19-20].

- Can the type A and type B uncertainties be considered identical (e.g. Sensitivity Robustness) [7-8]? Discuss empirically the behavior of type A and type B uncertainties on a wide range of industrial problems to find the answer.

- Can the type D and type E uncertainties be considered identical? Review literature to find the answer [9-12] [15][21-23].

- How to process the Type C uncertainties? The Operations-research and Engineering optimization do not discuss it in details. Combine literature from Statistics, Signal processing and Machine Learning to discuss model uncertainty in Engineering? Is the idea of ensembling applicable to Engineering problems, what about model selection in Engineering optimization? Can we go beyond ensembling to process model uncertainty? [23].

- Is the idea of modelling uncertainty through Fuzzy-logic applicable anymore in Engineering problems? [13-15].

- Discuss direct-search methods for robust design optimization and compare them (already-existing) to find out their applicability in Engineering. What about their complexity and generalization? Which class(es) of uncertainty are these methods suited for? Discuss empirically and provide intuition for their working [1],[6-7],[12],[17],[23-25],[33] and [46-47].

- Are meta-modelling techniques a good solution to the problem of finding robust optimum? Their empirical investigation is missing from the literature. See [7, Section 4.2.2]

- The approach of Pattern-search methods [70] to find robust design optimum is missing from the literature. Discuss them, provide their working principle, intuition and investigate empirically if they can be a reliable approach to find robust design optimum. See [7, Section 4.2.2]

- Properties of Genetic Algorithms for Robust solution searching scheme [1][7][29][34-35] are not discussed theoretically and empirically in detail, discuss them in details. Is increasing the population size a step towards robustness? If so, verify that empirically and (if possible) theoretically.

- Discuss Evolution strategies for robust optimization for industrial problems. Provide a review of [1,7] for a range of industrial problems with special focus on $(\mu/\mu_1, \lambda)$-ES and CMA-ES. Compare ES with other direct search methods (overlap with previous Questions).

- There is a need for a set of scalable test functions and corresponding robust optimization problems that cold serves as a library for testing different optimization algorithms. See [7, Section 5.1].

Bibliography

[1] **J.Kruisselbrink**. *Evolution Strategies for Robust Optimization*. PhD Dissertation, Leiden University, The Netherlands, 2012.

[2] **K.Deb, A.Pratap, S.Agarwal and T.Meyarivan**. *A fast and elitist multiobjective genetic algorithm: NSGA-II*. IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182-197, 2002.

[3] **E.Zitzler,M.Laumanns and L.Thiele**. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm, 2001.*

[4] **N.Beume,B.Naujoks and M.Emmerich**. *SMS-EMOA: Multiobjective selection based on dominated hypervolume*. European Journal of Operational Research Volume 181, Issue 3, 16 September 2007, Pages 1653-1669.

[5] **M.Emmerich and A.H.Deutz**. *A tutorial on multiobjective optimization: fundamentals and evolutionary methods*. Natural Computing, Volume 17, Issue 3, pp 585–609 https://doi.org/10.1007/s11047-018-9685-y, 2018.

[6] **S.Andradottir**. *A review of simulation optimization techniques*. Proceedings of the 1998 Winter Simulation Conference, IEEE, Piscataway, NJ, 1998, pp. 151–158.

[7] **H.G.Beyer and B.Sendhoff**. *Robust optimization – A comprehensive survey*. Computer Methods in Applied Mechanics and Engineering Volume 196, Issues 33–34, 1 July 2007, Pages 3190-3218 2007.

[8] **D.Jung and B.Lee**. *Development of a simple and efficient method for robust optimization*,2002.

[9] **B.Sudret and A.D.Kiureghian**. *Stochastic Finite Element Methods and Reliability A State-of-the-Art Report*,2000.

[10] **H.Agarwal**. *Reliability based design optimization: Formulations and methodologies*,Ph.D. Thesis, University of Notre Dame, South Bend, IN, 2004.

[11] **R.Rackwitz**. *Reliability analysis – a review and some perspectives*, Structural Safety 23 365–395,2001.

[12] **J.Martin and T.Simpson**. *A Monte Carlo method for reliability-based design optimization*,Tech. Rep. 2006-2146, American Institute of Aeronautics and Astronautics AIAA, 2006.

[13] **F.Campos,J.Villar and M.Jimenez**. *Robust solutions using fuzzy chance constraints*, Engrg.Optim. 38 (6) (2006) 627–645, 2006.

[14] **S.Chen**. *Comparing probabilistic and fuzzy set approaches for design in the presence of uncertainty*, Ph.D. thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2000.

[15] **M.Arakawa, H.Yamakawa and H.Ishikawa**. *Robust design using fuzzy numbers with intermediate variables*, 3rd World Congress of Structural and Multidisciplinary Optimization, 1999.

[16] **M.E.Pate-Cornell**. *Uncertainties in Risk Analysis: Six Levels of Treatment.* , Reliability Engineering and System Safety, 54(2-3):95–111, November 1996.

[17] **A.Der Kiureghian and O.Ditlevsen**. *Aleatory or Epistemic? Does it matter ?* ,Structural Safety, 31(2):105–112, March 2009.

[18] **A.Eiben, T.Bäck, M.Schoenauer and H.P.Schwefel**. *Parallel Problem Solving from Nature*, vol. 5, SpringerVerlag, Heidelberg, 1998, pp. 119–128.

[19] **W.Chen, M.Wiecek and J.Zhang**. *Quality utility – a compromise programming approach to robust design*, ASME J. Mech. Des. 121 (2) (1999) 179–187.

[20] **N.Lagaros, V.Plevris and M.Papadrakakis**. *Multi-objective design optimization using cascade evolutionary computation*, Comput. Methods Appl. Mech. Engrg. 194 (2005) 3496–3515.

[21] **X.Du and W.Chen**. *Towards a better understanding of modeling feasibility robustness in engineering design*,ASME J. Mech. Des. 122 (2) (2000) 385–394.

[22] **M.Papadrakakis, N.Lagaros and V.Plevris**. *Design optimization of steel structures considering uncertainties*,Engrg. Struct. 27 (9) (2005) 1408–1418.

[23] **M.Fu**. *Optimization for simulation: theory vs. practice*, INFORMS J. Comput. 14 (2002) 192–215.

[24] **K.H.Lee, G.J.Park and W.S.Joo**. *A general robust optimization using the kriging-based approximation method*, Proceedings of the 6th World Congress on Structural and Multidisciplinary Optimization (WCSMO6), ISSMO,Rio de Janeiro, 2005.

[25] **Y.Jin**. *A comprehensive survey of fitness approximation in evolutionary computation*, Soft Comput. J. 9 (1) (2005) 3–12.

[26] **U.Hammel and T.Bäck**. *Evolution Strategies on Noisy Functions, How to Improve Convergence Properties*, In Parallel Problem Solving from Nature (PPSN III), volume 866 of LNCS, pages 159–168. Springer, 1994.

[27] **N.Hansen, A.S.P.Niederberger, L.Guzzella and P. Koumoutsakos**. *A Method for Handling Uncertainty in Evolutionary Optimization with an Application to Feedback Control of Combustion*, IEEE Transactions on Evolutionary Computation, 13(1):180–197, February 2009.

[28] **Y.Jin and B.Sendhof**. *Trade-off between Performance and Robustness: An Evolutionary Multiobjective Approach*, In: Fonseca C.M., Fleming P.J., Zitzler E., Thiele L., Deb K. (eds) Evolutionary Multi-Criterion Optimization. EMO 2003. Lecture Notes in Computer Science, vol 2632. Springer, Berlin, Heidelberg.

[29] **H.G.Beyer**. *Evolutionary Algorithms in Noisy Environments: Theoretical Issues and Guidelines for Practice*, Computer Methods in Applied Mechanics and Engineering, 186(2-4):239–267, June 2000.

[30] **V.Heidrich Meisner**. *Evolutionary Direct Policy Search in Noisy Environments*, PhD thesis, University of Bochum, 2011.

[31] **U.Hammel and T.Bäck**. *Evolution Strategies Applied to Perturbed Objective Functions*,In IEEE Congress on Evolutionary Computation (CEC 1994), pages 40–45. IEEE Press, 1994.

[32] **S.Meyer-Nieberg and H.G.Beyer**. *Why Noise May Be Good: Additive Noise on the Sharp Ridge.*,In Genetic and Evolutionary Computation Conference (GECCO 2008), pages 511–518, 2008.

[33] **S.Wager, S.Wang and P.Liang**. *Dropout Training as Adaptive Regularization*,arXiv:1307.1493v2 , 2013.

[34] **J.M.Fitzpatrick and J.J.Grefenstette**. *Genetic Algorithms in Noisy Environments*, Machine Learning, 3:101–120, October 1988.

[35] **B.L.Miller and D.E.Goldberg**. *Genetic Algorithms, Selection Schemes and the Varying Effects of Noise*, Evolutionary Computation, 4:113–131, June 1996.

[36] **N.Hansen and A.Ostermeier**. *Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation*, IEEE Congress on Evolutionary Computation (CEC 1996), pages 312–317. IEEE Press, 1996.

[37] **I.Rechenberg**. *Evolutionstrategie '94. Frommann-Holzboog, 1994.*

[38] **H.G.Beyer**. *Mutate Large, but Inherit Small! On the Analysis of Rescaled Mutations in (1,λ)-ES with Noisy Fitness Data* In Parallel Problem Solving from Nature (PPSN V), volume 1498 of LNCS, pages 109–118. Springer-Verlag, 1998.

[39] **S.Markon, O.Markon, D.V.Arnold, T.Bäck, T. Beielstein and H.G.Beyer**. *Thresholding — A Selection Operator for Noisy ES* In Congress on Evolutionary Computation (CEC'01), pages 465–472. IEEE Press, 2001.

[40] **A.N.Aizawa and B.W.Wah**. *Dynamic Control of Genetic Algorithms in a Noisy Environment* , In Proceedings of the Fifth International Conference on Genetic Algorithms, pages 48–55. Morgan Kaufman, 1993.

[41] **A.N.Aizawa and B.W.Wah**. *Scheduling of Genetic Algorithms in a Noisy Environment* , Evolutionary Computation, 2(2):97–122, June 1994.

[42] **J.Kruisselbrink, M.Emmerich, and T.Bäck**. *On the Limitations of Adaptive Resampling Using the Student's t-Test in Evolution Strategies* , In Genetic and Evolutionary Computation Conference (GECCO 2009), pages 2649–2656. ACM, 2009.

[43] **E.Cantu'Paz**. *Adaptive Sampling for Noisy Problems* , In Genetic and Evolutionary Computation Conference (GECCO 2004), pages 947–958. ACM, 2004.

[44] **G.Rudolph**. *A Partial Order Approach to Noisy Fitness Functions* , In IEEE Congress on Evolutionary Computation (CEC 2001), pages 318–325. IEEE Press, 2001.

[45] **V.Heidrich-Meisner and C.Igel**. *Hoeffding and Bernstein Races for Selecting Policies in Evolutionary Direct Policy Search* , In Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, pages 401–408. ACM, 2009.

[46] **Y.Sano and H.Kita**. *Optimization of Noisy Fitness Functions by Means of Genetic Algorithms Using History of Search* , In Parallel Problem Solving from Nature (PPSN VI), volume 1917 of LNCS, pages 571–580. Springer, 2000.

[47] **Y.Sano and H.Kita**. *Optimization of Noisy Fitness Functions by Means of Genetic Algorithms Using History of Search with Test of Estimation.* , In IEEE Congress on Evolutionary Computation (CEC 2002), pages 360–365. IEEE Press, 2002.

[48] **J.Branke, C.Schmidt, and H.Schmeck**. *Efficient Fitness Estimation in Noisy Environments.* , In Genetic and Evolutionary Computation Conference (GECCO 2001), pages 243–250, 2001.

[49] **S.Tsutsui and A.Ghosh**. *Genetic Algorithms with a Robust Solution Searching Scheme.* , IEEE Transactions on Evolutionary Computation, 1(3):201–208, September 1997.

[50] **D.Wiesmann, U.Hammel, and T.Bäck**. *Robust Design of Multilayer Optical Coatings by means of Evolutionary Algorithms* , IEEE Transactions on Evolutionary Computation, 2(4):162–167, November 1998.

[51] **S.Tsutsui**. *A Comparative Study on the Effects of Adding Perturbations to Phenotypic Parameters in Genetic Algorithms with a Robust Solution Searching Scheme* , In IEEE Systems, Man, and Cybernetics Conference (SMC 1999), pages 585–591, 1999.

[52] **J.Branke**. *Creating Robust Solutions by Means of Evolutionary Algorithms* , In Parallel Problem Solving from Nature (PPSN V), volume 1498 of LNCS, pages 119–128. Springer-Verlag, 1998.

[53] **J.Branke**. *Reducing the Sampling Variance when Searching for Robust Solutions* , In Genetic and Evolutionary Computation Conference (GECCO 2001), pages 235–242. Morgan Kaufmann, 2001.

[54] **S.Tsutsui and A.Ghosh**. *Effects of Adding Perturbations to Phenotypic Parameters in Genetic Algorithms for Searching Robust Solutions* , In Advances in Evolutionary Computing: Theory and Applications, pages 351–365. SpringerVerlag, 2003.

[55] **S.Tsutsui and A.Ghosh**. *Inverse Multi-Objective Robust Evolutionary Design Optimization in the Presence of Uncertainty* , In Genetic and Evolutionary Computation Conference (GECCO 2005), pages 55–62. ACM, 2005.

[56] **Y.S.Ong,P.B.Nair and K.Y.Lum**. *Max-Min Surrogate-Assisted Evolutionary Algorithm for Robust Design* , IEEE Transactions on Evolutionary Computation, 10(4):392–404, August 2006.

[57] **H.G.Beyer and B.Sendhoff**. *Functions with Noise-induced Multi-Modality: A Test for Evolutionary Robust Optimization — Properties and Performance Analysis.* , IEEE Transactions on Evolutionary Computation, 10(5):507–526, October 2006.

[58] **D.H.Jung and B.C.Lee**. *Development of a Simple and Efficient Method for Robust Optimization.* , International Journal for Numerical Methods in Engineering, 53(9):2201–2215, March 2002.

[59] **Y.S.Ong, P.B.Nair, and K.Y.Lum**. *Max-Min Surrogate-Assisted Evolutionary Algorithm for Robust Design.* , IEEE Transactions on Evolutionary Computation, 10(4):392–404, August 2006.

[60] **C.Barrico and C.H.Antunes**. *Robustness Analysis in Multi-Objective Optimization Using a Degree of Robustness Concept.* , In IEEE Congress on Evolutionary Computation (CEC 2006). IEEE Press, July 2006.

[61] **N.J.Samsatli,L.G.Papageorgiou and N.Shah**. *Robustness Metrics for Dynamic Optimization Models Under Parameter Uncertainty.* ,AIChE Journal, 44(9):1993–2006, April 1998.

[62] **P.N.Koch, R.J.Yang, and L.Gu**. *Design for Six Sigma Through Robust Optimization.* ,Structural and Multidisciplinary Optimization, 26(3-4):235–248, January 2004. 1998.

[63] **S.Tsutsui, A.Ghosh, and Y.Fujimoto**. *A Robust Solution Searching Scheme in Genetic Search.*, In Parallel Problem Solving from Nature (PPSN IV), volume 1141 of LNCS, pages 543–552. Springer-Verlag, 1996.

[64] **J.Kruisselbrink,E.Reehuis,A.Deutz,T.Bäck and M.Emmerich**. *Using the Uncertainty Handling CMA-ES for Finding Robust Optima.*, In Genetic and Evolutionary Computation Conference (GECCO 2011). ACM, 2011.

[65] **H.G.Beyer and B.Sendhoff** *Evolution Strategies for Robust Optimization.*, In IEEE Congress on Evolutionary Computation (CEC 2006), pages 1346–1353. IEEE Press, 2006.

[66] **A.Ben-Ta and A.Nemirovski**. *Robust convex optimization.*, Math. Oper. Res. 23 (4) (1998) 769–805.

[67] **A.Ben-Ta and A.Nemirovski**. *Robust optimization – methodology and application.*, Math. Programm. 92 (3) (2002) 453–480.

[68] **A.Ben-Ta and A.Nemirovski**. *Robust solutions of uncertain quadratic and conic-quadratic problems.*, SIAM J. Optim. 13 (2) (2002) 535–560.

[69] **J.Darlington, C.C.Pantelides, B.Rustem and B.Tanyi**. *Algorithms for optimal decisions under uncertainty.*,in: 4th International Parallel Computing Workshop, PCW'95, Imperial College/Fujitsu Parallel Computing Research Centre, London, 1995, pp. 137–157

[70] **M.Wright**. *Direct search methods: once scorned, new respectable.*,in: D. Griffiths, G. Watson (Eds.), Numerical Analysis, Addison Wesley Longman Limited, Redwood, 1995, pp. 191–208.