Model-assisted robust optimization for continuous black-box problems

Ullah, S.

CHAPTER 2

Background

This chapter provides the necessary background and context for robust optimization. Starting with an overview of black-box optimization and related material in Section 2.1, we delineate some of the most important concepts related to robust optimization in Section 2.2. Section 2.3 provides an overview on surrogate modeling, followed by two of the widely adopted modeling techniques, namely the *response surface models* and *Kriging*. Lastly, we provide a short summary of the chapter in Section 2.4.

2.1 Black-Box Optimization

We start with an abstract system[1], which takes some input $\mathbf{x}$, and produces some output $\mathbf{y}$. The goal of optimization is to find such setting(s) of the input $\mathbf{x}$ accepted by the system, which produce the best possible output $\mathbf{y}$. We refer to such a system as a black-box, since no further information about the system is assumed (Alarie et al., 2021; Conn et al., 2009). This refers to the fact that the internal dynamics and mechanism of the system are unknown to the designer. Such a system can represent a wide range of optimization problems in practice, such as finding the optimal control parameters of an industrial production line.

In the following, we define some of the most important concepts related to the optimization of such a system.

Domain It may also be referred to as the *search space*, and contains the set of all inputs accepted by the system. We denote it with symbol $\mathcal{S}$ throughout

[1]The notion of "abstract" refers to the fact that no particularities are assumed on the input, output, and the internal functionality of the system.

this thesis. Examples of some important domains include Discrete spaces (Korte et al., 2011), Hilbert spaces of functions (Balakrishnan, 2012), and Mixed-integer spaces (Belotti et al., 2013). In this thesis, the discussion is always restricted to the search space $\mathcal{S} \subseteq \mathbb{R}^D$ with Euclidean metric, where D denotes the dimensionality. The resulting optimization problems are known as *real parameter* optimization problems. The practice of optimization that deals with real parameter problems is referred to as *continuous optimization* (Wright et al., 1999).

Objective Function In optimization, the purpose of the objective function is to assign score to each input based on the quality of the output. In this thesis, we deal with real-valued black-box functions, which means that the objective function represents a black-box system. Furthermore, no additional analytical properties, e.g., continuity, differentiability, and smoothness, are assumed on the objective function, and the only available information about the objective function is taken to be the evaluation of points in its domain (Audet and Hare, 2017).

$$f \; : \; \mathcal{S} \subseteq \mathbb{R}^D \rightarrow \mathbb{R}^M, \tag{2.1}$$

where the domain $\mathcal{S}$ is assumed to be a subset of the D-dimensional Euclidean space, and its image is $\mathbb{R}^M$.

Single-Objective Optimization Problem A *real-valued single-objective optimization problem* is a special problem in continuous optimization which has exactly one objective, i.e., $M = 1$ in Eq. (2.1). Without loss of generality, the optimization of such a problem can be defined as the problem of determining a *global minimum* $\mathbf{x}^*$ as:

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathcal{S}} \; f(\mathbf{x}), \tag{2.2}$$

where the definition of global minimizer is provided later in this section.

Multi-Objective Optimization Problem A *real-valued multi-objective optimization problem* is also a special case of an optimization problem with at least two objectives, i.e., $M > 1$ in Eq. (2.1). For this class of optimization problems, the definition of optimality is often based on the notion of Pareto dominance. Note that dominance can be defined by introducing a partial order on the space of objective function values, and can result in weak dominance, strict dominance, or in-comparability.

The majority of the current work in this context emphasizes on obtaining a representative subset of Pareto optimal solutions, which are based on the notion of non-dominance introduced by Edgeworth, and later independently by Vlifredo Pareto (Pareto et al., 1971). Within the scope of Pareto methods, nature inspired heuristics have been successfully integrated, resulting in famous algorithms, such as Non-dominated Sorting Genetic Algorithm-II (NSGA-II) (Deb et al., 2002), Strength Pareto Evolutionary Algorithm2 (SPEA2) (Kim et al., 2004), and Multi-objective particle swarm optimization (MOPSO) (Coello and Lechuga, 2002), among others. For a technical overview on multi-objective optimization, please refer to the work of Emmerich and Deutz (Emmerich and Deutz, 2018).

Definition 2.1 (Black-Box Optimization). An objective function f as defined in Eq. (2.1) is called a black-box if no prior knowledge about f is available, and the only accessible information about f is the objective value: $f(\mathbf{x}),\ \forall\ \mathbf{x} \in \mathcal{S}$. The optimization of such a function is referred to as *black-box optimization* in this thesis.

Definition 2.2 (Global Minimum). For a given single-objective optimization problem, a candidate solution: $\mathbf{x}^* \in \mathcal{S}$, is said to be a global minimum of f if: $\forall\ \mathbf{x} \in \mathcal{S},\ f(\mathbf{x}^*) \leq f(\mathbf{x})$.

Finding a global minimum is generally a difficult task due to multi-modality, discontinuity, and ill-conditioning. In practice, it is only possible to guarantee the convergence to the *local minimizer* (Wright et al., 1999).

Definition 2.3 (Local Minimum). For a given single-objective optimization problem, a candidate solution: $\mathbf{x} \in \mathcal{S}$, is said to be a local minimum of f if there is a neighborhood $N_{\mathbf{x}}$ of $\mathbf{x}$, such that: $\forall\ \mathbf{x}^{'} \in N_{\mathbf{x}},\ f(\mathbf{x}) \leq f(\mathbf{x}^{'})$.

As the search space is a subset of the metric space $\mathbb{R}^D$, in principle, any metric on $\mathbb{R}^D$ can be used to define the neighborhood. For instance, in the case of Euclidean metric, the neighborhood can be defined as a subset of the search space $\mathcal{S}$, which contains an open Euclidean ball around $\mathbf{x}$ as: $B_\epsilon(\mathbf{x}) = \{\mathbf{x}^{'} \in \mathcal{S} : ||\mathbf{x} - \mathbf{x}^{'}|| < \epsilon\}$, for any $\epsilon > 0$.

Definition 2.4 (Constrained Optimization Problems). When solving a black-box optimization problem in practice, we might encounter a set of *constraint functions*: $\mathcal{G} = \{g_1, \ldots, g_p\},\ p\ \in \mathbb{N}_0$. A real-valued black-box optimization problem with a set of constraint functions $\mathcal{G}$ is referred to as a *constrained optimization problem.*

Note that the set of constraint functions can include two types of constraints, namely inequality and equality constraints. In this thesis, we only present the inequality constraints for convenience since equality constraints can be easily transformed into inequality ones. In the presence of inequality constraints, the global minimizer $\mathbf{x}^*$ must belong to the set of feasible solutions $\mathcal{A}$, which can be defined as:

$$\mathcal{A} = \{\forall\ \mathbf{x} \in \mathcal{S} \mid g_i(\mathbf{x}) \geq 0\ ,\ i = 1, \ldots, p\}. \tag{2.3}$$

Definition 2.5 (Practical Goal of Optimization). Given a black-box optimization problem with an optimization goal and a finite amount of computational resources, the practical goal of optimization is to use these resources in an optimal way to find as good a solution as possible.

In an alternative sense, one can also aim for finding solution(s) that are an improvement with respect to the previously known best solution(s). Based on these reasons, one can also define a *global optimization algorithm* as the algorithm, that, given an infinite amount of computational resources, would get arbitrarily close to the global optimum.

2.2 Robust Optimization

The traditional view on black-box optimization as presented in the previous section does not account for the unexpected drifts and changes in the optimization setup. However, this is unrealistic for many real-world optimization scenarios. For instance, often in engineering applications, a non-deterministic simulator replaces the actual (physical) system (Bhosekar and Ierapetritou, 2018). The goal of optimization in these scenarios is to find solutions, such that the real-world realizations of these solutions are also of a good quality, even if they are subject to perturbations (Kruisselbrink, 2012).

It is intuitive to believe that we can face several types of uncertainties and noise in real-world applications (Beyer and Sendhoff, 2007). For instance, we can think of the uncertainty because an approximate model substitutes the actual (physical) system. Furthermore, the model itself may be non-deterministic/stochastic in nature. We can also think of the uncertainty in this situation because the real-world manufacturing of different parts of the system is precise only to a certain degree. Therefore, when these parts are assembled together, the system may not perform as expected. Taking these observations into consideration, we are faced with three issues as:

- How can uncertainties and noise arise in black-box optimization?
- In what ways can we classify such uncertainties based on their common characteristics?
- How can we mitigate the effects of such uncertainties in practical scenarios?

In the following, we summarize the state-of-the-art to answer these questions.

2.2.1 Uncertainties and Noise in Black-Box Optimization

Uncertainties and noise comprise one of the most challenging areas in black-box optimization. They are encountered frequently in real-world optimization problems (Jurecka, 2007). Below, we provide a few reasons why they can appear in practical scenarios.

- The search variables, also referred to as the decision variables, can not be controlled with unlimited precision in reality, e.g., manufacturing tolerances.
- The operational or environmental conditions for an industrial product or process can only be known to a certain extent.
- The output of the (physical) system, or a model of the system, is intrinsically stochastic.
- An approximate model may replace the real-world system within the optimization loop.
- The objective and the constraint functions can be fuzzy in nature, e.g., a degree of vagueness on the objective and constraint functions might exist.

Because of these reasons, we can establish that uncertainties and noise surround the black-box system in practical scenarios, since they can emerge in the input and output of the system, in addition to the modeling and evaluation of the system (Beyer and Sendhoff, 2007). Therefore, it is intuitive to believe that the common assumptions for solving real-world black-box problems can be significantly compromised in the face of uncertainty and noise. But in order to effectively account for these uncertainties and noise, a nomenclature is needed. To this end, we follow the categorization of Beyer and Sendhoff to a large degree (Beyer and Sendhoff, 2007), in combination with the work of Kruisselbrink (Kruisselbrink, 2012).

2.2.2 Sources of Uncertainty and Noise

Here, we first categorize uncertainty and noise by looking at their origin within the general loop of black-box optimization. For this purpose, an elaborated version of the black-box optimization loop is provided in Fig. 2.1, which highlights the potential sources of uncertainty and noise. In this figure, an optimizer[1] is coupled to the system, or a model of the system, for which optima are sought. The optimizer generates some candidate solution(s), which is/are fed to the system. The system evaluates this/these solution(s), and provides a quality score of this/these candidate solution(s). Based on this feedback, the optimizer generates a new set of candidate solution(s), which is/are fed to the system again for evaluation (Pošík et al., 2012). This loop is repeated until either a satisfactory solution is found, or a predefined computational budget, or other termination criterion is reached (Audet and Hare, 2017).

In Fig. 2.1, we can identify five regions of interest where uncertainty or noise can arise and affect the black-box optimization loop.

(I) Uncertainties and/or noise in the search/decision variables, denoted as $\mathbf{x}$.

(II) Uncertainties and/or noise in the environmental or operating conditions (generally referred to as the environmental variables, and denoted as α).

(III) Uncertainties and/or noise in the evaluation(s) of the candidate solution(s), denoted as $\mathbf{y}$.

(IV) Vagueness when modeling the constraints.

[1] The notion of "optimizer" is used to refer to a particular solution, as well as an optimization algorithm, in this thesis.

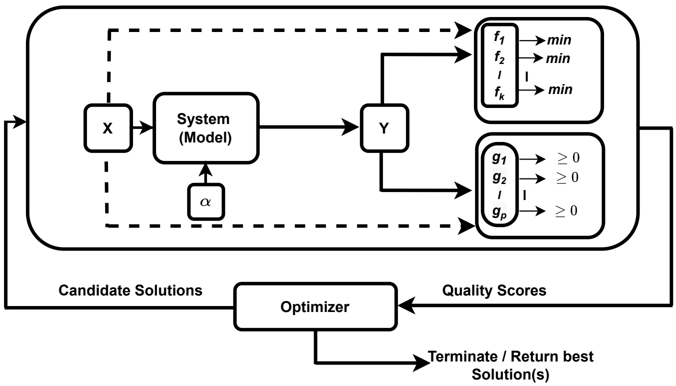

Figure 2.1: The general black-box optimization loop with five different sources of uncertainty. These sources include the decision variables $\mathbf{x}$, the environmental variables α), the evaluation of the system $\mathbf{y}$, the objectives f_i, and the constraint functions g_i.

(V) Preference uncertainty in the objectives, if the optimization problem has more than one objective.

The effect of these sources of uncertainty is presented in several different ways in black-box optimization.

(I) **Uncertainties and/or noise in the decision variables**

This type of uncertainty arises in practical scenarios because the real-world realizations of the candidate solutions differ arbitrarily much from their nominal values, which are used to find the optimal solutions. For instance, in the area of product engineering, we might encounter this uncertainty due to manufacturing tolerances, i.e., realizing a candidate solution to its nominal value might be too costly, and may not make an economic sense (Beyer and Sendhoff, 2007). With this type of uncertainty, we can usually face either of the following two scenarios.

Scenario 1:

An approximate model replaces the actual (physical) system within the black-box optimization loop (presented in Fig. 2.1). Thus, although the model might accept inputs with unlimited precision, the real-world (physical) system can only realize these inputs to a certain degree, e.g., in automobile design optimization, physical parts of the vehicle can only be manufactured with a limited precision (Chowdhury and Taguchi, 2016).

Scenario 2:

In case the real-world (physical) system is enclosed within the black-box optimization loop, the uncertainty in the inputs can propagate through the output, to the set of objective and constraint functions. Generally, the motif of the uncertainty in the inputs is unknown in advance. Hence, one can only make very general assumptions on the structure of the uncertainty (Rehman, 2016), e.g., additive vs multiplicative, deterministic vs stochastic, symmetric vs non-symmetric (Averbakh and Zhao, 2008).

The effect of the additive uncertainty $\Delta_{\mathbf{x}}$, in the search variables, can be represented by reformulating the objective function as:

$$\tilde{f}(\mathbf{x}) = f(\mathbf{x}, \Delta_{\mathbf{x}}) = f(\mathbf{x} + \Delta_{\mathbf{x}}), \tag{2.4}$$

For constraint functions, similar formulation can be adopted:

$$\tilde{g}_j(\mathbf{x}) = g_j(\mathbf{x}, \Delta_{\mathbf{x}}) = g_j(\mathbf{x} + \Delta_{\mathbf{x}}),\ j = \{1, \ldots, p\}. \tag{2.5}$$

In Eqs. (2.4)-(2.5), we are not making any assumption on the way in which the uncertainty $\Delta_{\mathbf{x}}$ is mathematically modeled, which would be the topic of interest later in this chapter.

(II) **Uncertainties and/or noise in the environmental variables**

In design optimization, the uncontrollable environmental variables are generally assumed to be system constants. In practical scenarios, however, it is found that they fluctuate, and can affect the performance of anotherwise stable system (Beyer and Sendhoff, 2007). As such, it is useful to think that these fluctuations can also affect the objective and constraint functions, similar to the uncertainty in the search variables (Jin and Branke, 2005).

(III) **Uncertainties and/or noise in the output**

This class of uncertainty is formed in the evaluation of the candidate solutions. Here, we can distinguish between two different scenarios.

Scenario 1:

The system is inherently non-deterministic in nature. Therefore, the precise evaluation of the candidate solutions is impossible, and the resulting output is noisy and stochastic (Nissen and Propach, 1998).

Scenario 2:

The system produces a deterministic output. However, this output can not be realized in practice, e.g., due to the manufacturing imprecisions and tolerances, or similar issues.

(IV) **Uncertainty in the constraints**

Another class of uncertainty is the ambiguity and the vagueness when mathematically formulating the set of constraint functions. This is due to the fact that there are several different types of constraints, e.g., soft vs hard constraints, and probabilistic vs deterministic. The requirements for the satisfaction of these constraints can therefore be represented in various different ways in black-box optimization (Shahraki and Noorossana, 2014).

(V) **Preference uncertainty in the objectives**

Intrinsically, when dealing with a black-box optimization problem with multiple conflicting objectives, a source of uncertainty lies in the trade-off of the objective functions. This is due to the fact that the quality of the candidate solutions can only be known a posteriori. Hence, regarding the importance and trade-off of different objective functions, highly subjective decisions have to be made. This type of uncertainty deals with multi-objective optimization, and can be compensated for by introducing a partial order on the space of objective function values, such as Pareto dominance (Kruisselbrink, 2012).

2.2.3 Modeling Uncertainty and Noise

To properly account for the uncertainties and noise in black-box optimization, we have to describe the mathematical ways in which they can be modeled. However, for that, we first have to make an important, albeit an informal distinction between uncertainty and noise.

Uncertainty vs Noise In the existing literature on robust optimization, there does not appear to be a clear distinction between uncertainty and noise[1]. Here, we provide an informal distinction between both concepts, which are also related to *aleatory* and *epistemic* uncertainties respectively. The most basic understanding of aleatory uncertainty is that it is fundamentally irreducible, completely random, and almost certainly unavoidable (Der Kiureghian and Ditlevsen, 2009). This type of uncertainty can informally be thought of as the *additive noise* in the context of black-box design optimization (Beyer and Sendhoff, 2007).

Epistemic uncertainty on the other hand, is, in principle, due to the lack of understanding, knowledge, or information on the optimization problem (Der Kiureghian and Ditlevsen, 2009). The effect of this kind of uncertainty, can, therefore, be minimized by representing the optimization problem in another way, and/or with more data. Epistemic uncertainty can also be referred to as just the uncertainty, albeit in an informal setting. Similar view is also adopted by Cornell (Paté-Cornell, 1996). For a thorough discussion on the differences between uncertainty and noise, please refer to the work of Kruisselbrink (Kruisselbrink, 2012).

In the following, we review different ways of mathematically representing the uncertainty.

1. **Deterministic**

   One of the most important ways to mathematically describe the uncertainty is with the help of deterministic *crisp sets*, which describe the crisp possibility of the states of the uncertain variables (Ionescu-Bujor and Cacuci, 2004). Here, an uncertain variable is usually modeled as a pair: (A, m_A), where A serves as the crisp set, and m_A describes the membership function. Note that the membership function is usually of the form: $m_A : A \to \{0, 1\}$. A particular design $\mathbf{x} \in A$ can then take one of the two forms:

   - $\mathbf{x}$ is a member of the set A, if $m_A(\mathbf{x}) = 1$.
   - $\mathbf{x}$ is not a member of the set A, if $m_A(\mathbf{x}) = 0$.

   Note that crisp sets may also be referred to as the *classical set* or *full membership sets* in the literature (Ben-Tal et al., 2009; Beyer and Sendhoff, 2007).

[1] In this thesis, we use both terms interchangeably, which refer to the unexpected drifts and changes in the optimization setup.

2. **Probabilistic**

   In a probabilistic setting, an uncertain variable is assumed to be of a stochastic nature. A probabilistic measure can be established by measuring the probabilistic frequency of the events that may occur. Uncertainties of this type can be represented by the *probability (density) functions*. This refers to the fact that a function: $p : A \to \mathbb{R}_0$ maps every event $\mathbf{x} \in A$, to a probability value, which quantifies the likeliness of that event (Kruisselbrink, 2012).

3. **Possibilistic**

   In this setting, the uncertainty is formulated with fuzzy statements, which describe the possibility (or degree of membership) about the states of the uncertain variables of interest. As opposed to crisp sets in the deterministic setting, here we make use of the *fuzzy sets*. An uncertain variable is modeled as a pair: (A, m_A), where A serves as the fuzzy set, and m_A describes the membership function (Kruisselbrink, 2012). Note that the membership function in this setting is usually of the form: $m_A : A \to [0, 1]$. Thus, the degree of membership is a real value between 0 and 1. The degree of membership increase as we get close to 1 (Bagheri et al., 2016).

2.2.4 Cases of Uncertainty and Noise

So far, we have seen five major classes of uncertainty in black-box optimization (based on their origins), along side three common ways to mathematically represent them. This gives rise to a total of 15 scenarios in which we can encounter uncertainty in practical situations. It is intuitive that not all of these scenarios are equally important. Therefore, a question arises as to which of these scenarios should be given more consideration over the others for achieving robustness? Answering this question will limit the scope of this thesis to a well-defined class of uncertainty, which will then be the focus for the rest of the thesis. In Table 2.1, we provide a summary of different classes of uncertainty based on their conceptual distinction, mathematical representation, and common characteristics.

It is pertinent to note that the first two classes of uncertainty discussed in this section, namely the Class (I) and (II), are related to the so-called "sensitivity

Table 2.1: A summary of different categorizations of uncertainty in black-box optimization as described by Kruisselbrink (Kruisselbrink, 2012).

Conceptual Classification	Modeling	Characteristics
Epistemic (uncertainty)	Possibilistic	Domain unknown Probabilities unknown
	Deterministic	Domain known Probabilities unknown
Aleatory (noise)	Probabilistic	Domain known Probabilities known

robustness[1]" (Beyer and Sendhoff, 2007). Furthermore, Class (IV) and (V) are related to each other in that they do not directly affect the output of the black-box system. Instead, they influence the search space $\mathcal{S}$, and the set of feasible solutions $\mathcal{A}$. Uncertainty of Class (IV) is also related to another important concept – the so-called "reliability-based robustness" (Shahraki and Noorossana, 2014).

2.2.5 Scope of Robust Optimization

Given five different classes of uncertainty in black-box optimization, alongside three mathematical approaches to model them, one can identify several different scenarios of robust optimization as presented in Table 2.2. For the scope of robust optimization in this thesis, however, we limit ourselves to a few of these scenarios. This is due to the fact that not all of these cases are considered to belong to robust optimization in the literature. For instance, Bertsimas (Bertsimas et al., 2010, 2011) only considers the uncertainty in the decision variables to define robust optimization.

In this work, we only consider the first two types of uncertainties – Class (I) and (II) – to represent robust optimization. This refers to the fact that we only deal with sensitivity robustness – robustness which is associated with the sensitivity of the objective function with respect to the specific changes in the decision and environmental variables. The most important reasons for limiting the scope of this work to only the first two types of uncertainties.

[1]"Sensitivity robustness" refers to the sensitivity of the objective function with respect to the specific changes in the optimization setup.

Table 2.2: A summary of different cases of uncertainty and noise in black-box optimization as described by Kruisselbrink (Kruisselbrink, 2012). Bold types of modeling and algorithmic approaches are more common in the literature.

Class	Modeling	Major Approaches
Class (I)	(1) **Deterministic**	(1) **Evolutionary Algorithms**
	(2) **Probabilistic**	(2) Surrogate Modeling
	(3) Possibilistic	(3) Quasi-Newton Methods
Class (II)	(1) **Deterministic**	(1) Evolutionary Algorithms
	(2) **Probabilistic**	(2) Surrogate Modeling
	(3) Possibilistic	(3) Mathematical Programming
Class (III)	(1) Deterministic	(1) **Evolutionary Algorithms**
	(2) **Probabilistic**	(2) Surrogate Modeling
	(3) **Possibilistic**	(3) **Mathematical Programming**
Class (IV)	(1) Possibilistic	(1) **Fuzzy Logic**
	(2) Probabilistic	(2) **Monte-Carlo Methods**
Class (V)	(1) Deterministic	(1) **Evolutionary Algorithms**
	(2) Possibilistic	(2) Surrogate Modeling

- As indicated earlier, uncertainties of Class (IV) and (V) do not directly affect the candidate solutions. Instead, they affect the search space $\mathcal{S}$ and its image $\mathbb{R}^M$ when formulating the optimization problem. For this reason, they have been considered separately from robust optimization in the literature (Jurecka, 2007).

- Uncertainties of type (I) and (II) are most frequent in design optimization, and can also determine the practical applicability of the optimal solutions to a large degree (Rehman, 2016). Accounting for these types of uncertainties is therefore critical (Jurecka, 2007; Kruisselbrink, 2012).

- Accounting for the uncertainty of type (III) refers to optimizing a noisy objective function, instead of finding robust optima. This formulates another scenario of optimization under uncertainty. Although important, this scenario is often considered separately from robust optimization, where the focus is to find optimal solutions, which are practically applicable despite varying conditions. In his work, Kruisselbrink (Kruisselbrink, 2012) also considers this type of uncertainty into robust optimization, but treats it differently from the first two types.

Based on the information provided so far, we can now extend the practical goal of optimization in Definition 2.5 to formulate the general goal of robust black-box optimization.

Definition 2.6 (Practical Goal of Robust Black-Box Optimization). Given a black-box optimization problem with uncertainty and/or noise in the decision and environmental variables, alongside an optimization goal, and a limited amount of computational resources. the *practical goal of robust black-Box optimization* is to use these resources to find as good as possible solutions despite uncertainty and/or noise, which are also optimal and useful in the face of uncertainties/noise.

In the remainder of this thesis, we will only deal with single-objective numerical optimization problems, which are subject to uncertainty and noise in the decision/search variables. We will further assume that the uncertainty/noise is additive and structurally symmetric in nature, and can only be represented in a deterministic, or a probabilistic fashion.

2.3 Surrogate Modeling

Continuous optimization problems in real-world application domains, e.g., mechanics, engineering, economics and finance, can encompass some of the most complicated optimization setups. Principal obstacles in solving the optimization tasks in these areas involve multi-modality (Beasley et al., 1993), high dimensionality (Shan and Wang, 2010), and unexpected drifts and changes in the optimization setup (Kruisselbrink, 2012; Beyer and Sendhoff, 2007). Due to these obstacles and the black-box assumption on the optimization setup, traditional numerical optimization schemes, e.g., gradient descent and Newton methods, are rendered inapplicable. The majority of the optimization schemes applied in these areas now focus on utilizing direct-search methods (Lewis et al., 2000; Beyer and Sendhoff, 2007), in particular EAs (Bäck et al., 2018), and SAO (Keane et al., 2008).

The use of direct-search methods, also referred to as derivative-free methods (Audet and Hare, 2017), in numerical black-box optimization, can be attributed to the following reasons.

- Direct-search methods perform well in practice, since many of them are based on sound heuristics. Recent analysis demonstrates the global convergence behavior for some of these methods, similar to the results known for the globalized quasi-Newton methods (Lewis et al., 2000).

- Some of the most important characteristics of direct-search methods, e.g., no evaluation of the derivative, dictate the practical applicability of these methods, where more sophisticated techniques fail to perform (Audet and Kokkolaras, 2016).

Within the scope of direct search methods, EAs and SAO formulate two of the most important classes of techniques to solve non-linear black-box problems. This thesis deals with SAO since we further assume that the black-box problem is expensive to evaluate.

In the following, we provide a brief introduction to SAO.

2.3.1 Introduction

Surrogate-Assisted Optimisation (SAO) refers to solving the optimisation problem with the help of a surrogate model, also referred to as the meta-model, which replaces the actual function evaluations by the model prediction (Keane et al., 2008). The surrogate model estimates the true values of the objective function under consideration. This is desirable if the objective function is too costly to evaluate. The abstraction provided by the surrogate model is useful in a variety of situations. For instance, it simplifies the task to a great extent in simulation-based modeling and optimisation, by providing the opportunity to evaluate the objective function indirectly if the exact computation is intractable. Surrogate models can also provide practically useful insights, e.g., space visualization and comprehension, about the search space (Forrester et al., 2008).

The idea of SAO was proposed as early as 1974 (Schmit Jr and Farshi, 1974). This line of research was particularly useful in structural optimization with the name of *response surface approximation*. Some of the most important contributions, concerning the applicability of SAO for structural optimization, include the initial work of Svanberg (Svanberg, 1987), Toropov (Toropov et al., 1993), Roux (Roux et al., 1998), Box (Box and Draper, 1987), and Myers (Myers et al., 2016). The first attempt to classify such methodologies, based on their accuracy, in the context of structural engineering, was made in 1993 (Barthelemy and Haftka, 1993).

In the context of optimization under uncertainty, surrogate models were investigated by Rehman (Rehman, 2016), Jurecka (Jurecka, 2007), Jin (Jin et al., 2003), and Persson (Persson and Ölvander, 2013). For a detailed review of surrogate modeling and its applications in structural engineering, please refer to the work of Jurecka (Jurecka, 2007).

In the following, we provide a brief overview for *response surface models* and *Kriging* models, two of the widely utilized modeling techniques.

2.3.2 Response Surface Models

The term response surface models (RSM) can be somewhat misleading, since all types of surrogate models construe a "surface", which enables the designer to estimate the function response at untried locations. For this reason, the term RSM has also been used as a synonym for surrogate models in the literature. A different understanding of the term, however, points to the "polynomial regression models" (Bishop, 2007), which were initially utilized for the analysis of physical experiments (Santner et al., 2003; Jurecka, 2007).

The basic idea behind RSM is to establish an explicit functional relationship (response surface) between the input and the output variables (Hastie et al., 2009). We start with the design data of N vectors of co-variate values $\mathrm{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}^\top$, where each one of these vectors denotes a sample point in the search space $\mathcal{S} \subseteq \mathbb{R}^D$. The corresponding response values of the function f are denoted as $\mathbf{y} = [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_N)]^\top$. Then, a polynomial approximation of the function f, of degree M, at an untried location $\mathbf{x}$, can be written as:

$$\hat{f}(\mathbf{x}, M, \boldsymbol{\gamma}) = \gamma_0 + \gamma_1 \mathbf{x} + \gamma_2 \mathbf{x}^2 + \cdots + \gamma_M \mathbf{x}^M = \sum_{j=0}^{M} \gamma_j \mathbf{x}^j, \tag{2.6}$$

where the free parameters: $\boldsymbol{\gamma} = \{\gamma_0, \gamma_1, \ldots, \gamma_M\}^\top$ can be estimated through the maximum likelihood principle as: $\boldsymbol{\Phi}\boldsymbol{\gamma} = \mathbf{y}$, and $\boldsymbol{\Phi}$ is the Vandermonde matrix (Kalman, 1984) defined as:

$$\boldsymbol{\Phi} = \begin{pmatrix} 1 & \mathbf{x}_1 & \mathbf{x}_1^2 & \cdots & \mathbf{x}_1^M \\ 1 & \mathbf{x}_2 & \mathbf{x}_2^2 & \cdots & \mathbf{x}_2^M \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \mathbf{x}_N & \mathbf{x}_N^2 & \cdots & \mathbf{x}_N^M \end{pmatrix}. \tag{2.7}$$

The maximum likelihood estimate of $\boldsymbol{\gamma}$ can be proven to be:

$$\boldsymbol{\gamma} = \boldsymbol{\Phi}^+ \mathbf{y}, \tag{2.8}$$

where $\mathbf{\Phi}^{+} = (\mathbf{\Phi}^{\top}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\top}$, is the Moore-Penrose pseudo-inverse (Golub and Van Loan, 2013) of $\mathbf{\Phi}$. Using Eq. (2.8), we can estimate the value of the free parameters.

Note that the polynomial approximation $\hat{f}$ of the objective function f, based on M degrees is essentially, a Taylor series expansion of f truncated after $M+1$ terms. From this observation, it follows that a greater value of M will yield a better approximation. However, with greater number of terms, the approximation also becomes too flexible, and there might be over-fitting and poor generalization capability. We can prevent this phenomenon by restricting the value of M to be small (Forrester et al., 2008; Bishop, 2007; Hastie et al., 2009).

One of the ways to do it is through cross-validation, which can determine the optimal setting of M for a given problem. We can also prevent over-fitting with the help of regularization, such as Lasso and Ridge regularization (Hastie et al., 2009; Bishop, 2007). Throughout this thesis, we utilize a RSM with degree $M = 2$, combined with the so-called *elastic-net penalty*, which linearly combines the Lasso and Ridge regularization terms. It is also pertinent to mention that a higher value of M will result in a more (computationally) expensive approximation of f, since computing the inverse of $\mathbf{\Phi}$ in Eq. (2.7) will become much costlier.

2.3.3 Kriging

Kriging is an interpolation technique based on geostatistics (Woodard, 2000; Rasmussen and Williams, 2006), and has been widely utilized as a surrogate modeling tool in Design and Analysis of Computer Experiments (Sacks et al., 1989; Santner et al., 2003), Surrogate-Assisted Evolutionary Algorithms (Emmerich, 2005), Global Optimization (Jones et al., 1998), and Algorithm Configuration (Hutter et al., 2011). Similar to RSM, we start with the data of N vectors of co-variate values as: $\mathrm{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}^{\top}$, and the corresponding functions responses as: $\mathbf{y} = [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_N)]^{\top}$. Kriging formulates that the function response at any untried search point $\mathbf{x}$ can be described as a normally distributed random variable $Y(\mathbf{x})$ with mean μ and variance σ^2. Furthermore, for any pair $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^D$, the correlation between $f(\mathbf{x})$ and $f(\mathbf{x}')$ is modeled by a kernel function (Rasmussen and Williams, 2006). Here, we describe the popular Matérn 3/2 kernel:

$$k(\mathbf{x}, \mathbf{x}') = \left(1 + \sqrt{3}l\right) e^{-\sqrt{3}l},\ l = \sqrt{\sum_{i=1}^{D} \theta_j (\mathbf{x}_j - \mathbf{x}'_j)^2}, \tag{2.9}$$

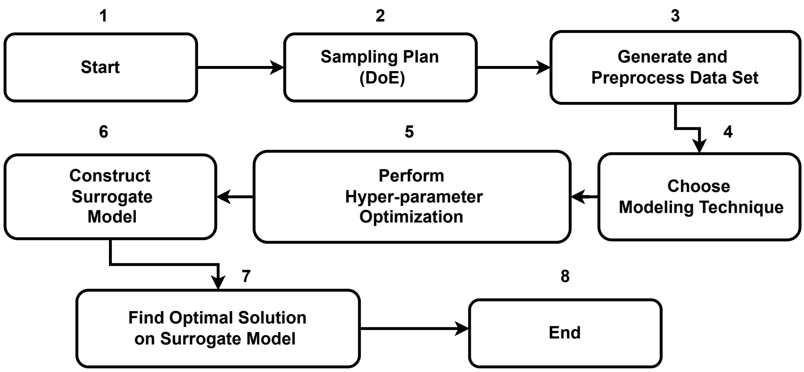

Figure 2.2: Flowchart showing the implementation of a "one-shot optimization" strategy in this thesis. The "one-shot optimization" strategy is based on surrogate modeling, to find the optimal solutions in an efficient manner.

where D represents the dimensionality of the problem, and θ_j measures the influence of the j-th dimension with respect to the search domain. Then, the Kriging prediction of the function response at any untried point $\mathbf{x}$ can be shown to be:

$$\hat{f}(\mathbf{x}) = \hat{\mu} + \mathbf{c}^\top \mathbf{\Sigma}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}), \tag{2.10}$$

where $\mathbf{c}$ is the vector of correlations between $\mathbf{x}$ and each of the N sample points, $\hat{\mu}$ is the generalized least square estimator of μ, $\mathbf{\Sigma}$ is the $N \times N$ correlation matrix between N sample points with elements defined by Eq. (2.9), and $\mathbf{1}$ is a vector of 1's.

An estimated mean squared error (MSE) of $\hat{f}$ arises naturally from Kriging's theoretical setup:

$$s^2(\mathbf{x}) \coloneqq \mathrm{E}\{Y(\mathbf{x}) - \hat{f}(\mathbf{x})\}^2 = \sigma^2 \left[1 - \mathbf{c}^\top \mathbf{\Sigma}^{-1}\mathbf{c} + \frac{1 - \mathbf{1}^\top \mathbf{\Sigma}^{-1}\mathbf{c}}{\mathbf{1}^\top \mathbf{\Sigma}^{-1}\mathbf{1}}\right]. \tag{2.11}$$

The MSE is zero at the sample points since the true response of the function is known at these locations.

2.3.4 Surrogate Modeling in Practice

In this thesis, we implement SAO in two different ways: in the framework of a "one-shot optimization" (OSO) strategy (Ta'asan et al., 1992), and with the help of a "sequential model-based optimization" (SMBO) framework (Jones et al., 1998). As we shall see, the former is more desirable with regards to practicality (Chapter 3), due to the potential difficulties and pitfalls of extending the latter to the robust scenario (ur Rehman et al., 2014). However, we note that the latter is more stringent in nature, due to the fact that it updates the surrogate model in an iterative manner, according to a sampling infill criterion (Jurecka, 2007). The sampling infill criterion encodes the search behavior, i.e., balances the trade-off of exploration and exploitation, and can be utilized to find a globally optimal solution on the model surface (Jones et al., 1998).

The working mechanism of OSO strategy in this thesis is described as follows. We start by generating an initial design data set $\mathcal{D} = (\mathrm{X}, \mathbf{y})$, on the objective function f. The locations $\mathrm{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ can be determined by the DoE methodologies, such as the Latin Hyper-cube Sampling (LHS) scheme (Montgomery, 2017). After this, objective function values $\mathbf{y} = [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_N)]^\top$ are computed on these locations. The next step involves constructing the surrogate model based on the available data set $\mathcal{D}$. Note that before constructing the surrogate model, we perform Hyper-parameter Optimization (HPO) to estimate the best configuration of the corresponding hyper-parameters, in order to achieve the best quality surrogate model based on the available function evaluations (Hutter et al., 2009). Once the surrogate model is constructed, we utilize a benchmark numerical optimization algorithm (Wright et al., 1999) to find the optimal solution on the model surface, and the process comes to a halt (Ullah et al., 2019). Since we do not perform an adaptive sampling in this case, the sampling infill criterion does not need to be extended to care for robustness. This allows us to be much more thorough and comprehensive in our approach, as we can take into account the variability in external factors without involving the prohibitively high computational costs (Bossek et al., 2019).

Similar to the previous case, in the SMBO approach, we also construct the initial design data set $\mathcal{D} = (\mathrm{X}, \mathbf{y})$ (Jurecka, 2007). After this, we construct the surrogate model based on the available data set. Following this, the next query point $\mathbf{x}_{\text{new}}$ (to sample the function) is determined with the help of a sampling infill criterion, such as the "Expected Improvement" criterion (Jones et al., 1998). The function response $f(\mathbf{x}_{\text{new}})$ is computed at this location, and the data set $\mathcal{D}$ is extended by appending the pair $(\mathbf{x}_{\text{new}}, f(\mathbf{x}_{\text{new}}))$ to it. The surrogate model is then updated based on the extended data set (Močkus, 2012). This process is repeated until either a satisfactory solution is obtained, or a predetermined computational budget,

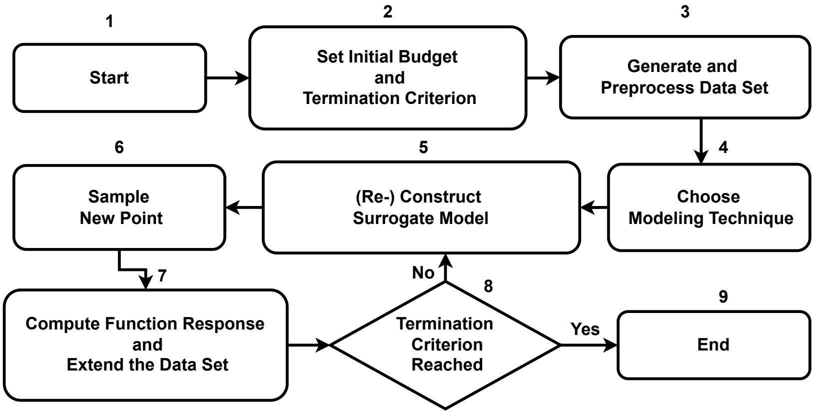

Figure 2.3: Flowchart shows the implementation of "sequential model-based optimization" framework in this thesis. The "sequential model-based optimization" framework updates the surrogate model based on a sampling infill criterion, which encodes the search behavior to find the optimal solution on the model surface.

or another termination criterion is reached. Since at each iteration, the surrogate model is updated according to an infill criterion, the optimal solution can be obtained in an efficient manner (Jones et al., 1998). While the SMBO approach is deemed a powerful heuristic to find a globally optimal solution on the model surface, extending it to the robust scenario is a much more difficult task. The potential difficulties and pitfalls of extending the SMBO approach to the robust scenario are explained in detail in Chapter 4.

In this thesis, we attempt to answer some research questions with the help of a OSO strategy (Chapter 3). These research questions target the potential of surrogate modeling to find robust solutions, and the related difficulties thereof. The potential of surrogate modeling in this context is evaluated by varying sample size, modeling technique, problem landscape, dimensionality, robustness formulation, and noise level among others. The OSO strategy is readily applicable to answer these questions in an empirical fashion (Ullah et al., 2019). However, we also assume that SMBO is a a powerful heuristic, and it is possible to answer more advanced research questions with it (Ullah et al., 2021). These research questions deal with the impact of the sampling infill criterion, computational cost of robustness, and the choice of a robustness criterion in practical scenarios (Chapters 4 and 5).

Important Remarks

In the remainder of the thesis, the terms "efficient global optimization", "sequential model-based optimization", and "Bayesian optimization" are used interchangeably to refer to the same concept – surrogate-assisted optimization, where the surrogate model is iteratively updated according to a criterion/merit to find the optimal solution. The chosen criterion/merit is referred to as the "sampling infill criterion", or the "acquisition function", which controls the search behavior of the algorithm, i.e., balances the trade-off between exploration and exploitation.

2.4 Summary and Discussion

This chapter provides a concise overview on three different but related topics, namely black-box optimization, robust optimization, and surrogate modeling, respectively. Section 2.1 provides a short description of black-box optimization, as well as definitions for some of the most important and related concepts. This section also emphasizes on the practical goal of optimization with regards to computational tractability, i.e., computational resources/budget available. Note that the practical goal of optimization proposes to use the computational resources in an optimal way to find better solutions, which are an improvement with respect to the previously known best solutions (Wright et al., 1999).

Section 2.2 introduces the notion of uncertainty and noise in black-box optimization. Based on their origins, five different classes of uncertainty and noise are identified. The sources of uncertainty include decision/search variables, environmental variables, output/evaluation of the system, constraints, and objectives, respectively. Furthermore, three mathematical ways of modeling these uncertainties: deterministic, probabilistic, and possibilistic, are presented. Based on the combinations of different classes of uncertainties alongside their mathematical representations, we limit the scope of this thesis to only deal with the uncertainties of the first two types, which can be represented in a deterministic or a probabilistic fashion (Ullah et al., 2019).

In Section 2.3, we provide a short overview of surrogate modeling, which utilizes the empirical models to substitute the expensive function evaluations. Surrogate modeling can be helpful in multiple different ways in black-box optimization (Forrester et al., 2008). We describe the working mechanism of two of the most important modeling techniques, namely the RSM and Kriging. We also describe two different manifestations of surrogate modeling in this thesis, namely the “one-shot optimization” approach, and the “sequential model-based optimization” approach respectively.