Chapter 5
Overview of Problem Formulations and Optimization Algorithms in the Presence of Uncertainty

Mathieu Balesdent (✉), Loïc Brevault, Jérôme Morio, and Rudy Chocat
e-mail: mathieu.balesdent@onera.fr

Optimization under uncertainty is a key problem in order to solve complex system design problem while taking into account inherent physical stochastic phenomena, lack of knowledge, modeling simplifications, etc. Different reviews of optimization techniques in the presence of uncertainty can be found in the literature (Sahinidis 2004; Klamroth et al. 2017). The choice of the algorithm is often problem-dependent (Gabrel et al. 2014). The designer has to choose firstly the optimization problem formulation with respect to the system specifications and study but also the optimization algorithm to apply. The objective of this chapter is to present the different existing approaches to solve an optimization problem under uncertainty and to focus specifically on the uncertainty handling mechanisms. The chapter is organized as follows. Firstly, in Section 5.1, different optimization problem formulations are introduced, highlighting the importance of uncertainty measures and the distinctions between robustness-based formulation, reliability-based formulation and robustness-and-reliability-based formulation. Then, in Section 5.2, different approaches to quantify the uncertainty in optimization are discussed. Finally, in the Section 5.3, an overview of optimization algorithms is presented with a focus on stochastic gradient, population-based algorithms and surrogate-based approaches. For each type of algorithms the handling of uncertainty is analyzed and discussed.

5.1 Optimization Problem Formulations

5.1.1 Introduction

Let us begin with the following generic deterministic constrained optimization problem. One aims at minimizing an objective function (also referred to as fitness or

L. Brevault et al., *Aerospace System Analysis and Optimization in Uncertainty*,
Springer Optimization and Its Applications 156,
https://doi.org/10.1007/978-3-030-39126-3_5

criterion) $f : \mathbb{R}^n \rightarrow \mathbb{R}$ under m inequality constraints $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ $(i = 1, \ldots, m)$. $\mathbf{z} \in \mathbb{R}^n$ is the vector composed of the optimization (design) variables.

$$\min \quad f(\mathbf{z}) \tag{5.1}$$

$$\text{w.r.t.} \quad \mathbf{z} \tag{5.2}$$

$$\text{s.t.} \quad g_i(\mathbf{z}) \leq 0 \;\; \text{for } i \in \{1, \ldots, m\} \tag{5.3}$$

$$\mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max}. \tag{5.4}$$

It is assumed that equality constraints in the presence of uncertainty may be transformed into two inequality constraints with associated tolerances. In order to illustrate the differences between deterministic and uncertainty formulations of the optimization problem, let us consider the following toy problem. Let us first consider the *deterministic* optimization formulation:

$$\min \quad \left(z_1^2 + z_2 - 15\right)^2 + \left(z_1 + z_2^2 - 11\right)^2 + 10(5 - z_1) + 10(5 - z_2) \tag{5.5}$$

$$\text{w.r.t.} \quad z_1, z_2 \tag{5.6}$$

$$\text{s.t.} \quad \left((z_1 + 5)^2 + (z_2 + 6)^2 - 100\right) \leq 0 \tag{5.7}$$

$$-5 \leq z_1, z_2 \leq 5. \tag{5.8}$$

The objective function is derived from the Himmelblau's function (Himmelblau 1972) to have only one global optimum, the constraint is a parabolic function. The objective function and the constraint are depicted in Figure 5.1. This optimization problem has four local optima (one three unconstrained and one constrained) and one global constrained optimum.

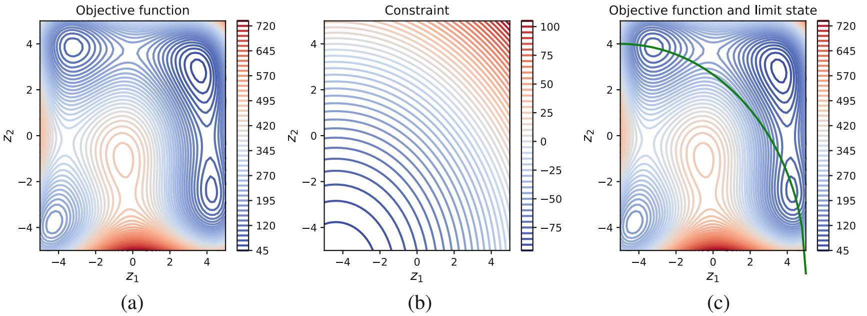


Fig. 5.1 Iso-contour of deterministic objective function (left) and constraint (center) and objective function combined with limit-state function (right). (**a**) Objective function $f(\cdot)$. (**b**) Constraint $g(\cdot)$. (**c**) Deterministic objective function $f(\cdot)$ and limit state $g(\mathbf{z}) = 0$

When uncertainties are introduced, these latter may impact both the objective function and the constraints (or only the objective function, or some constraints) and the former optimization problem becomes

$$\begin{aligned} \min \quad & f(\mathbf{z}, \mathbf{U}) && (5.9)\\ \text{w.r.t.} \quad & \mathbf{z} && (5.10)\\ \text{s.t.} \quad & g_i(\mathbf{z}, \mathbf{U}) \leq 0 \ \text{ for } i \in \{1, \ldots, m\} && (5.11)\\ & \mathbf{z}_{\text{min}} \leq \mathbf{z} \leq \mathbf{z}_{\text{max}}, && (5.12) \end{aligned}$$

where $f : \mathbb{R}^n \times \mathbb{R}^d \rightarrow \mathbb{R}$ and $g_i : \mathbb{R}^n \times \mathbb{R}^d \rightarrow \mathbb{R}$ $(i = 1, \ldots, m)$. $\mathbf{U} \in \mathbb{R}^d$ takes values in stands for the vector of uncertainties. In this section we use the probability formalism. We define a probability space $(\Omega, A, \mathbb{P})$, with Ω the sample space, A the σ-algebra, and $\mathbb{P}$ the measure of uncertainty. For more details on the probability theory framework, see Chapter 2.

In this problem, only the design variables $\mathbf{z}$ are degrees of freedom of the optimization problem. The different quantities of interest (objective function and constraints) are affected by the variability of the uncertain variables and become functions with random outputs. In practice, the problem represented by Equations (5.9–5.12) is not appropriately defined to be solved and one has to introduce measures of uncertainties on the objective function and constraints to solve the optimization problem. Let us define $\Xi[\cdot]$ and $\mathbb{K}[\cdot]$ the measures of uncertainties on, respectively, the objective function and constraints. The following optimization problem holds:

$$\begin{aligned} \min \quad & \Xi\left[f(\mathbf{z}, \mathbf{U})\right] && (5.13)\\ \text{w.r.t.} \quad & \mathbf{z} && (5.14)\\ \text{s.t.} \quad & \mathbb{K}_i\left[g_i(\mathbf{z}, \mathbf{U})\right] \leq 0 \ \text{ for } i \in \{1, \ldots, m\} && (5.15)\\ & \mathbf{z}_{\text{min}} \leq \mathbf{z} \leq \mathbf{z}_{\text{max}}. && (5.16) \end{aligned}$$

The main measures of uncertainty that can be used are recalled here:

- the **expectation** (or mean or expected value). With considering $\phi_{\mathbf{U}}(\cdot)$ the probability density function (PDF) of the uncertain variables $\mathbf{U}$, the expression of the expectation (here applied on the objective function) is $\mathbb{E}[f(\mathbf{z}, \mathbf{U})]$:

$$\mathbb{E}[f(\mathbf{z}, \mathbf{U})] = \int\limits_{\mathbb{R}^d} f(\mathbf{z}, \mathbf{u}) \phi_{\mathbf{U}}(\mathbf{u}) \mathrm{d}\mathbf{u} \qquad (5.17)$$

- the **variance**. We recall here the expression of the variance (applied to the objective function)$\mathbb{V}[f(\mathbf{z}, \mathbf{U})]$ is

$$\mathbb{V}[f(\mathbf{z},\mathbf{U})] = \mathbb{E}\left[(f(\mathbf{z},\mathbf{U}) - \mathbb{E}[f(\mathbf{z},\mathbf{U})])^2\right] \tag{5.18}$$

- the **failure probability** defined here for a constraint and a failure threshold T by

$$P_f = \mathbb{P}\left[g(\mathbf{z},\mathbf{U}) \geq T)\right]. \tag{5.19}$$

  In Chapter 4, several techniques to estimate of probability of failure have been introduced to efficiently compute such a measure of uncertainty for a constraint $g(\cdot)$.
- a **quantile**. For example, for a constraint $g(\cdot)$ and a given quantile level k, the quantile is expressed by

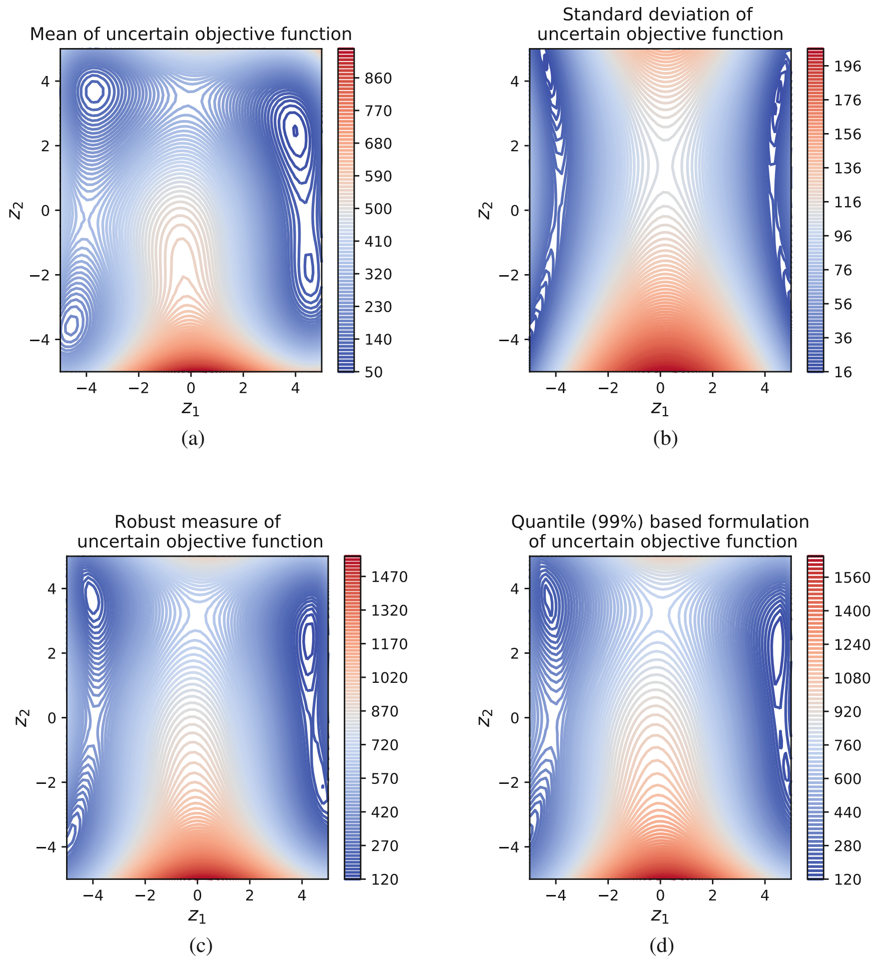
$$\mathbb{Q}_k(g(\mathbf{z},\mathbf{U})) = \inf\{\mathrm{q} \in \mathbb{R} : \mathbb{P}\left(g(\mathbf{z},\mathbf{U}) \leq \mathrm{q}\right) \geq k\}. \tag{5.20}$$

It is possible to combine several of these uncertainty measures for the objective function or the constraints as presented in the following sections. Following the previously introduced deterministic toy problem, let us now consider its "uncertainty-based version":

$$\begin{aligned}
\text{min} \quad & \left(z_1^2 + z_2 - 15 - 0.5U^2\right)^2 + \left(z_1 + z_2^2 - 11 + U\right)^2 && (5.21)\\
& +10(5 - z_1) + 10(5 - z_2) && (5.22)\\
\text{w.r.t.} \quad & z_1, z_2 && (5.23)\\
\text{s.t.} \quad & \left(\left(z_1 + 5 - 0.1(U - 2.5)^2\right)^2 + (z_2 + 6 - 2U)^2 - 100\right) \leq 0 && (5.24)\\
& -5 \leq z_1, z_2 \leq 5 && (5.25)
\end{aligned}$$

with U a scalar random variable following a Gaussian distribution $\mathcal{N}(2., 1.5)$. In Figure 5.2 are represented the expected value of the objective function (a), its standard deviation (b), a linear combination of these two latters (c) (also called robust formulation as described hereinafter) and quantile of level 0.99 (d). As it can be seen in this figure, the choice of the uncertainty measure has an impact on the result of the optimization, in terms of value of objective function and location of the optimum in the design space (for z_1, z_2). Figure 5.3 sums up different uncertainty measures (and corresponding limit-state) on the constraint. As it can be seen, the choice of a given reliability measure on the constraint has a great impact on the conservativeness on the found solution (Figure 5.4).

Depending on the different measures of uncertainty the designer has to choose the most representative uncertainty measure for his design problem. The choice of a given uncertainty measure for the objective function and constraints leads to a particular formulation of the design problem. These latter are described in the following sections. For more details relative to the different design problem



(a)

(b)

(c)

(d)

Fig. 5.2 Different measures of uncertainty applied to the objective function. (**a**) Expectation of objective function: $\mathbb{E}(f(\mathbf{z}, \mathbf{U}))$. (**b**) Standard deviation of objective function: $\mathbb{V}(f(\mathbf{z}, \mathbf{U}))^{\frac{1}{2}}$. (**c**) Linear combination of expected value and standard deviation of objective function: $\mathbb{E}(f(\mathbf{z}, \mathbf{U})) + 3\mathbb{V}(f(\mathbf{z}, \mathbf{U}))^{\frac{1}{2}}$. (**d**) Quantile (99%) of objective function: $\mathbb{Q}_{0.99}(f(\mathbf{z}, \mathbf{U}))$

formulations, the interested reader can consult (Yao et al. 2011; Lelièvre et al. 2016). The choice of the formulation is driven by the goal of the design study. Several papers (Beck et al. 2015; Lelièvre et al. 2016) discuss the differences and impact of the formulation to the obtained design. These formulations are compared in Lelièvre et al. (2016) on different numerical examples.

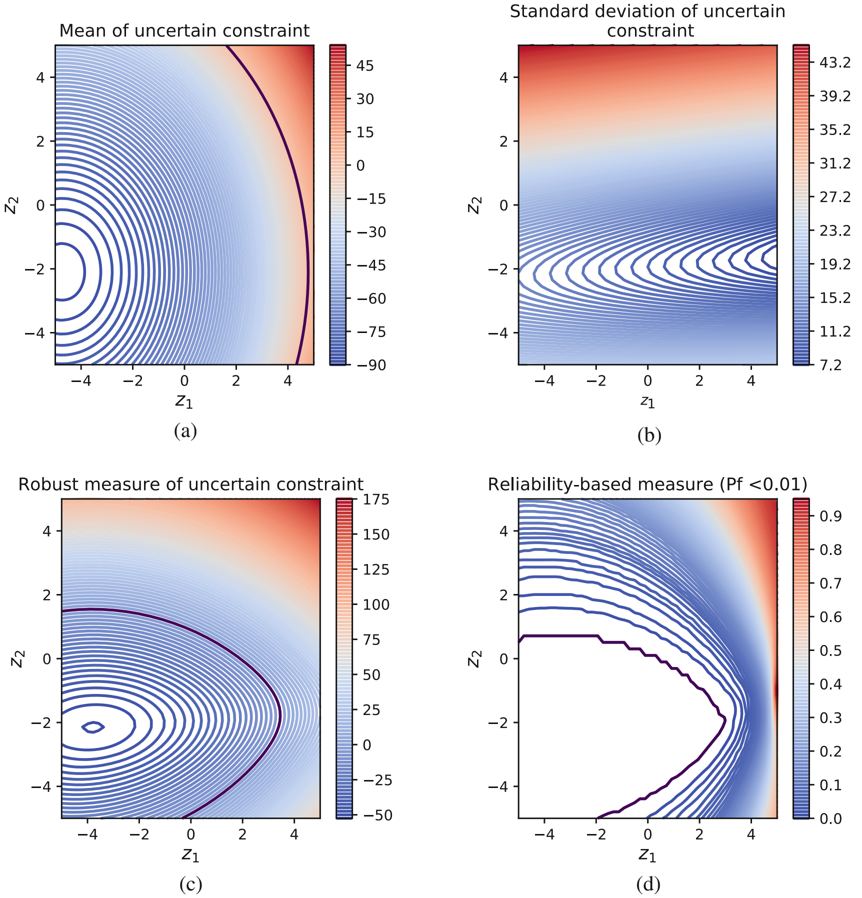


Fig. 5.3 Different measures of uncertainty (and corresponding limit state in purple) applied to the constraint. (**a**) Expectation of constraint: $\mathbb{E}(g(\mathbf{z}, \mathbf{U}))$. (**b**) Standard deviation of constraint: $\mathbb{V}(g(\mathbf{z}, \mathbf{U}))^{\frac{1}{2}}$. (**c**) Linear combination of expected value and standard deviation of constraint: $\mathbb{E}(g(\mathbf{z}, \mathbf{U})) + 3\mathbb{V}(g(\mathbf{z}, \mathbf{U}))^{\frac{1}{2}}$. (**d**) Reliability-based measure of constraint: $\mathbb{P}(g(\mathbf{z}, \mathbf{U}) \leq 0) \leq 0.01$

5.1.2 Robust Formulation

The robustness-based formulation (often referred to as robust formulation) consists in ensuring that the design is insensitive to uncertainty. Different formulations of robust design can be found in the literature (Yao et al. 2011; Lelièvre et al. 2016). This can be achieved by considering both the minimization of the objective function and the minimization of its variations, typically by solving the following bi-objective optimization problem:

$$\min \quad \{\mathbb{E}\left[f(\mathbf{z}, \mathbf{U})\right], \mathbb{V}\left[f(\mathbf{z}, \mathbf{U})\right]\} \tag{5.26}$$

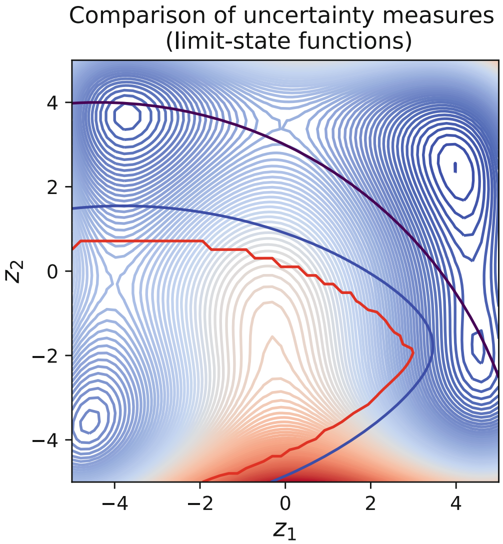


Fig. 5.4 Comparison of limit-state function for the mean (purple), robust (blue) and reliability-based (red) measures of the constraint (objective function is taken as the mean of $f(\mathbf{z}, \mathbf{U})$)

$$\text{w.r.t.} \quad \mathbf{z} \tag{5.27}$$

$$\text{s.t.} \quad \mathbb{K}_i\left[g_i(\mathbf{z}, \mathbf{U})\right] \le 0 \quad \text{for } i \in \{1, \ldots, m\} \tag{5.28}$$

$$\mathbf{z}_{\min} \le \mathbf{z} \le \mathbf{z}_{\max}. \tag{5.29}$$

Most often, this multi-objective optimization problem is replaced by a single-objective one by considering a weighted sum of the expectation and the standard deviation of the objective function (Papadrakakis et al. 2005):

$$\min \ \Xi\left[f(\mathbf{z}, \mathbf{U})\right] = \mathbb{E}\left[f(\mathbf{z}, \mathbf{U})\right] + k_s \mathbb{V}\left[f(\mathbf{z}, \mathbf{U})\right]^{\frac{1}{2}} \tag{5.30}$$

with $k_s > 0$. In this formulation, the term k_s ensures that the results will be insensitive to k_s time the standard deviation of the objective function: the larger k_s, the more conservative the design.

The same measure can also be applied on the constraint $g_i(\cdot)$ by taking into account the following relationship:

$$\mathbb{K}_i\left[g_i(\mathbf{z}, \mathbf{U})\right] = \mathbb{E}\left[g_i(\mathbf{z}, \mathbf{U})\right] + k_{s_i} \mathbb{V}\left[g_i(\mathbf{z}, \mathbf{U})\right]^{\frac{1}{2}}. \tag{5.31}$$

When applied to the constraint, this formulation is based on the statistical moments of the inequality function vector to ensure that despite the uncertainty, the system will stay feasible. k_{s_i} indicates the restriction of the feasible region to k_{s_i} standard deviations away from the mean value of the constraint function vector.

5.1.3 Reliability-Based Formulation

The reliability-based formulation considers a reliability measure and aims at ensuring that the solution at the convergence will respect a given tolerance on a probability of failure:

$$\mathbb{K}_i\left[g_i(\mathbf{z},\mathbf{U})\right] = \mathbb{P}\left[g_i(\mathbf{z},\mathbf{U}) > 0\right] - P_{t_i} \tag{5.32}$$

with P_{t_i} a given tolerance (for example, 0.01) for the constraint $g_i(\cdot)$. This formulation is based on the estimation of the failure probability of the constraint, which can be very cumbersome to evaluate (see Chapter 4 for more details on probability of failure estimations).

5.1.4 Robust and Reliability-Based Formulation

The robust and reliability-based formulation consists in combining a robust formulation for the objective function with a reliability-based formulation on the constraint vector (Yao et al. 2011). This leads to the following optimization problem:

$$\begin{aligned}
\text{min}\quad & \Xi\left[f(\mathbf{z},\mathbf{U})\right] = \mathbb{E}\left[f(\mathbf{z},\mathbf{U})\right] + k_s \mathbb{V}\left[f(\mathbf{z},\mathbf{U})\right]^{\frac{1}{2}} && (5.33)\\
\text{w.r.t.}\quad & \mathbf{z} && (5.34)\\
\text{s.t.}\quad & \mathbb{K}_i\left[g_i(\mathbf{z},\mathbf{U})\right] = \mathbb{P}\left[g_i(\mathbf{z},\mathbf{U}) > 0\right] - P_{t_i} \leq 0 \ \text{ for } i \in \{1,\ldots,m\} && (5.35)\\
& \mathbf{z}_{\text{min}} \leq \mathbf{z} \leq \mathbf{z}_{\text{max}}. && (5.36)
\end{aligned}$$

5.2 Quantification of Uncertainty in Optimization

With considering generic measures of uncertainty (expectation, variance, probability of failure, quantile, etc.), the following optimization problem holds (Figure 5.5):

$$\begin{aligned}
\text{min}\quad & \Xi\left[f(\mathbf{z},\mathbf{U})\right] && (5.37)\\
\text{w.r.t.}\quad & \mathbf{z} && (5.38)\\
\text{s.t.}\quad & \mathbb{K}_i\left[g_i(\mathbf{z},\mathbf{U})\right] \leq 0 \ \text{ for } i \in \{1,\ldots,m\} && (5.39)\\
& \mathbf{z}_{\text{min}} \leq \mathbf{z} \leq \mathbf{z}_{\text{max}} && (5.40)
\end{aligned}$$

In condition to have exact estimators of uncertainty measures $\Xi[\cdot]$ and $\mathbb{K}_i[\cdot]$, the previous optimization problem becomes deterministic and can be solved using classical deterministic optimization algorithms. However, the main difficulty comes

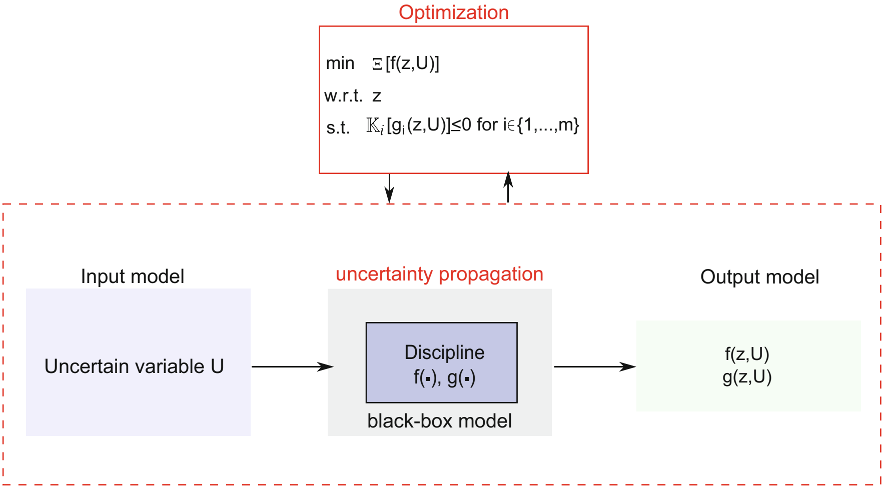
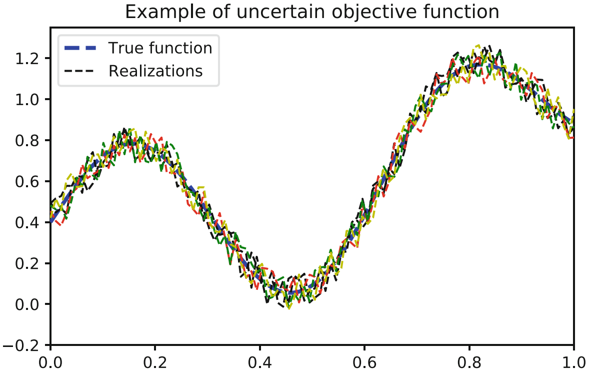


Fig. 5.5 Optimization under uncertainty for single discipline problem

Fig. 5.6 Impact of "noise" on objective function



from the estimation of uncertainty measures by numerical methods. Most often, the uncertainty measures require multivariate integral calculations which have to be estimated using numerical sampling schemes (Monte Carlo estimation, quadrature rules, etc.) due to the presence of black-box functions, see Chapter 3 for more details.

Taking into account directly the uncertainty in order to solve the optimization is crucial. The uncertainties make the objective function and constraints "noisy," i.e., these functions can take different values from a same value of deterministic optimization variables as depicted in Figure 5.6.

The global objective here is to find the values of design variables $\mathbf{z}$ that minimize the objective function $\Xi[f(\cdot)]$ and satisfying the uncertainty measures on constraints despite the impact of noise. It is thus necessary to carefully handle the noise to optimize the objective function with respect to deterministic design variables. For that purpose, different approaches can be derived (Hansen et al. 2009a).

The most common method consists in resampling the noisy function a given number of times, the uncertainty measure $\Xi\left[f(\mathbf{z},\mathbf{U})\right]$ for a same value of the design vector $\mathbf{z}$ (Aizawa and Wah 1994; Branke and Schmidt 2003; Cantú-Paz 2004; Stagge 1998) and estimating in a statistics (generally the expected value) from the samples, this statistics $\hat{\Xi}\left[f(\mathbf{z},\mathbf{U})\right]$ is then used for the optimization purpose. For instance, if $\Xi\left[f(\mathbf{z},\mathbf{U})\right] = \mathbb{E}\left[f(\mathbf{z},\mathbf{U})\right]$ and Crude Monte Carlo (CMC) is used to estimate this measure, in order to account for CMC numerical uncertainty, k repetitions of the estimation of $\mathbb{E}^{\mathrm{CMC}}\left[f(\mathbf{z},\mathbf{U})\right]$ are carried out leading to $\hat{\Xi}\left[f(\mathbf{z},\mathbf{U})\right] = \frac{1}{k}\sum_{j=1}^{k}\mathbb{E}_j^{\mathrm{CMC}}\left[f(\mathbf{z},\mathbf{U})\right]$ for the same value of $\mathbf{z}$. Even if this method is easy to understand and implement, the main drawback is the numerical cost that can be very large if the noisy function is numerically expensive. In order to reduce this cost, specific resampling methods have been proposed as described in Cantú-Paz (2004) and Stagge (1998). One of the main ideas of such methods is to quantify the number of required resampling using statistical tests such as t-test (Stagge 1998). The numerical cost can also be reduced by limiting the design points that will be re-evaluated according to their rank of the variance of the function values on these points.

Another classical method consists in building a surrogate model of the noisy functions with a Design of Experiment (Branke 1998; Branke et al. 2001). The noisy function is replaced by the surrogate model within the optimization problem. The impact of noise is reduced as the surrogate model is used to smooth the optimization functions. This kind of methods may present difficulties in case of high-dimensional problem (Hansen et al. 2009a).

Once the optimization formulation is set, the problem solving requires an appropriate optimization algorithm. In the following, three different classes of optimizers under uncertainty are discussed: the gradient-based, the population-based, and the surrogate-based techniques. Concerning the population-based algorithms such as the evolutionary algorithms, as described hereafter, the population evolution mechanisms can be adapted specifically in order to handle the noise impact (Branke and Schmidt 2003; Beyer 2000).

5.3 Brief Overview of Optimization Algorithms

5.3.1 Stochastic Gradient

The most common method based on gradient information and dedicated to uncertain optimization is the stochastic gradient (Gardner 1984). This method uses the classical gradient descent approach with a Lagrangian method to minimize a given objective function as a linear combination of the objective criterion and the constraints. The gradient methods are very efficient methods to solve deterministic differentiable optimization problems and are based on the estimation of the gradient of the objective and the constraint functions. In the case of noisy functions, the main difficulty consists in estimating the gradient of the objective function and the constraints as their estimators are not directly analytically available. An adaptation of the Arrow–Hurwicz method (Andrieu et al. 2007) has been proposed to deal with noisy optimization problem. This algorithm is based on the information on $f(\mathbf{z}, \mathbf{U})$ and not on $\Xi\left[f(\mathbf{z}, \mathbf{U})\right]$ to determine the gradient of the noisy estimator of $f(\mathbf{z}, \mathbf{U})$. This algorithm is adapted for objective function $\Xi[\cdot]$ that can be expressed as a sum of realizations (typically the empirical expectation). This descent method is an iterative method and is based on the gradient of the objective and constraint functions and on particular realization of the uncertainty $\mathbf{U}$. This formulation is inspired from classic deterministic gradient-based approaches. The optimization under constraints can be handled as unconstrained optimization by using Lagrangian method. The new design point at the iteration $k+1$ is derived as a linear combination of the point at the iteration k and a function of the stochastic gradient of the objective function and constraints.

Let consider for simplification sake the following notations: $J(\mathbf{z}) = \Xi\left[f(\mathbf{z}, \mathbf{U})\right]$ and $\Gamma_i(\mathbf{z}) = \mathbb{K}_i\left[g_i(\mathbf{z}, \mathbf{U})\right] \leq 0$. Most often, the objective function considered is the expectation (Andrieu 2004; Arnaud 2014; Mercier 2018). The constraint is often the expectation of $g(\cdot)$ or a probability of failure (Andrieu 2004) (reliability-based formulation). Thus, the following optimization problem holds:

$$\text{min} \quad J(\mathbf{z}) \tag{5.41}$$

$$\text{w.r.t.} \quad \mathbf{z} \tag{5.42}$$

$$\text{s.t.} \quad \Gamma_i(\mathbf{z}) \leq 0 \ \text{ for } i \in \{1, \ldots, m\} \tag{5.43}$$

$$\mathbf{z}_{\text{min}} \leq \mathbf{z} \leq \mathbf{z}_{\text{max}} \tag{5.44}$$

with considering there exists an optimum in the admissible subspace Z^{ad} of design variable search space $\mathbf{z}$, the Lagrangian $L(\mathbf{z}, \boldsymbol{\kappa})$ can be written as follows:

$$L(\mathbf{z}, \boldsymbol{\kappa}) = J(\mathbf{z}) + \boldsymbol{\kappa}\Gamma(\mathbf{z}) \tag{5.45}$$

with $\boldsymbol{\kappa}$ the vector of Lagrange multipliers. If $\forall \mathbf{z} \in [\mathbf{z}_{\text{min}}, \mathbf{z}_{\text{max}}]$ the quantities $J(\mathbf{z})$ and $\Gamma(\mathbf{z})$ and their respective gradients are available, then the iterative equation of the Arrow–Hurwicz method without taking into account uncertainties can be applied with respect to the design variables and the Lagrange multipliers:

$$\mathbf{z}^{k+1} = \Pi_{Z^{ad}}\left(\mathbf{z}^k - \epsilon^k\left(\nabla_{\mathbf{z}} J\left(\mathbf{z}^k\right) + \boldsymbol{\kappa}^k \nabla_{\mathbf{z}}\Gamma\left(\mathbf{z}^k\right)\right)\right) \tag{5.46}$$

$$\boldsymbol{\kappa}^{k+1} = \Pi_{+}\left(\boldsymbol{\kappa}^k + \rho^k \Gamma\left(\mathbf{z}^k\right)\right) \tag{5.47}$$

with $\mathbf{z}^k$ the design point vector at the iteration k, $\Pi_{Z^{ad}}$ is the projection over the admissible subspace of $\mathbf{z}$, ϵ^k is the optimization step at the iteration k. $\nabla_{\mathbf{z}}$ is the gradient with respect to the design variables $\mathbf{z}$, Π_+ is the projection over the positive space of the Lagrange multipliers $\boldsymbol{\kappa}$.

This iterative sequence converges towards an optimum if the optimization step $(\epsilon^k)_{k\in\mathbb{N}}$ satisfies the Robbins Monroe conditions (Cohen and Culioli 1994; Arrow et al. 1958):

$$\sum_{k\in\mathbb{N}} \epsilon^k = +\infty, \sum_{k\in\mathbb{N}} \left(\epsilon^k\right)^2 \leq +\infty. \tag{5.48}$$

The step concerning the Lagrange multipliers $(\rho^k)_{k\in\mathbb{N}}$ follows the same condition.

In the context of estimation by sampling methods (CMC, etc.), the functions $J(\cdot)$ and $\Gamma(\cdot)$ become uncertain if the sampling budget is limited, that is often the case if the computational codes are numerically expensive. In Andrieu et al. (2007), is proposed an approach consisting in considering the information given by the objective and constraint $f(\cdot)$ and $\mathbf{g}(\cdot)$ as non-biased estimation of the estimator $J(\cdot)$ and $\Gamma(\cdot)$ and their gradients. In Andrieu (2004) and Andrieu et al. (2007), the constraints are handled as reliability constraints.

$$\Gamma(\mathbf{z}) = P_t - \mathbb{P}\left(g_i(\mathbf{z}, \mathbf{U}) > 0\right) \tag{5.49}$$

$\Gamma(\cdot)$ is defined by a limit-state function: $\mathbb{1}_{g(\mathbf{z},\mathbf{u})>0}$ which is not continuous:

$$\Gamma(\mathbf{z}) = \int \mathbb{1}_{g(\mathbf{z},\mathbf{u})>0}\phi(\mathbf{u})\mathrm{d}\mathbf{u} \ \leq P_t. \tag{5.50}$$

In Andrieu et al. (2011), the authors propose to replace the gradient of $J(\cdot)$ and $\Gamma(\cdot)$ by the gradient of objective function and constraints evaluated on a realization of uncertainty $\mathbf{u}^{k+1}$:

$$\mathbf{z}^{k+1} = \Pi_{Z^{ad}}\left(\mathbf{z}^k - \epsilon^k\left(\nabla_{\mathbf{z}} f\left(\mathbf{z}^k, \mathbf{u}^{k+1}\right) + \boldsymbol{\kappa}^k \nabla_{\mathbf{z}} \mathbb{1}_{g(\mathbf{z}^k,\mathbf{u}^{k+1})>0}\right)\right) \tag{5.51}$$

$$\boldsymbol{\kappa}^{k+1} = \Pi_+\left(\boldsymbol{\kappa}^k + \rho^k \mathbb{1}_{g(\mathbf{z}^k,\mathbf{u}^{k+1})>0}\right) \tag{5.52}$$

with $\mathbb{1}_{g(\mathbf{z}^k,\mathbf{u}^{k+1})>0}$ the limit-state function, $\Pi_{Z^{ad}}$ the projection of $\mathbf{z}$ in the feasible space, Π_+ the projection of Lagrange multipliers $\boldsymbol{\kappa}$ in the positive space. This algorithm is known as the stochastic Arrow–Hurwicz gradient. The Robbins Monroe conditions on the optimization step allow to converge through an optimum the presence of uncertainty (see Andrieu et al. 2011 for details). This method has been applied to the uncertainty-based design of a wing taking into account aeroelastic phenomena in Arnaud and Poirion (2014). Extension of this method to solve multi-objective optimization problems have been proposed in Mercier et al. (2018). These approach consists in coupling the Multiple Gradient Descent Algorithm (Désidéri 2014) with the stochastic gradient algorithm.

5.3.2 *Population-Based Algorithms*

General Principles

Population-based algorithms are very popular and huge amount of algorithms have been proposed since the last decades (Nabil 2016; Mirjalili and Lewis 2016) (Figure 5.7). These algorithms often try to mimic natural process or animal behaviors. Population-based algorithms are generally based on the scheme described in Figure 5.8.

These algorithms are based on the two fundamental principles:

- Exploitation of the results obtained for the current iteration in order to find promising areas of the search space (local search),
- (Stochastic) exploration of new areas in order to avoid local minima (global search).

The compromise between exploitation and exploration is crucial for the performance of such algorithms. For the sake of conciseness, only the most known

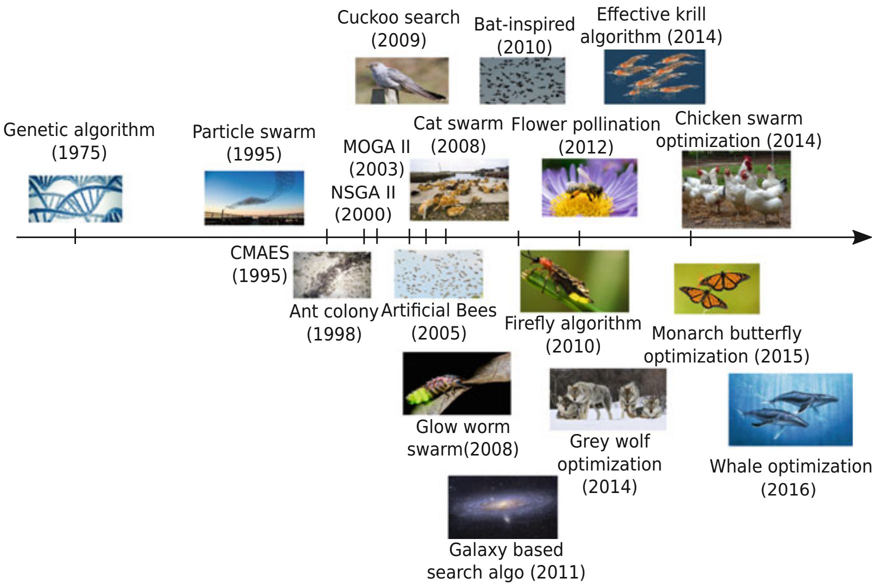


Fig. 5.7 Examples of population-based algorithms

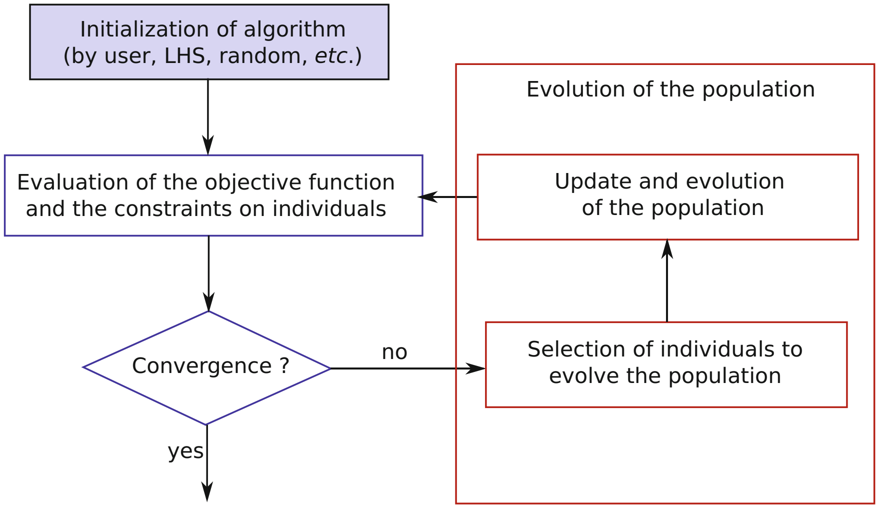

Initialization of algorithm
(by user, LHS, random, *etc.*)
Evaluation of the objective function
and the constraints on individuals
Convergence ?
no
yes
Evolution of the population
Update and evolution
of the population
Selection of individuals to
evolve the population

Fig. 5.8 General scheme of population-based algorithms

population-based algorithms will be described in the following of this section. For a complete survey of such algorithms, the interested readers may consult (Talbi 2009; Boussaïd et al. 2013).

Evolutionary Algorithms

Principles

Evolutionary algorithms are based on a population of N individuals. At each iteration of the algorithms several individuals of this population are selected to evolve the whole population. This can be achieved by involving recombination or mutation processes (e.g., genetic algorithm (Davis 1991; Holland 1975; Goldberg 1989)), or by defining sampling law to generate new individuals in interesting zones (e.g., Cross-Entropy (Rubinstein 1999), Evolution Strategies (Hansen et al. 2003)).

Evolutionary algorithms are often based on the same principles: *parents* are selected, then recombined, mutated, and evaluated to form *children* (Figure 5.9). In genetic algorithms (GA), information about the individuals is encoded into a *chromosome* (for instance, a binary string). The recombination and mutation are performed by exchanging a chain of genes between two individuals (crossover) or randomly inverting genes (in the binary case, swapping 0–1 and reciprocally) for an individual (mutation). The probabilities of performing these operations are hyperparameters of the algorithm and depend on the problem type and designer choice. Different kinds of selection processes of individuals to perform recombination and mutation may be used (roulette, tournament, ranking, etc.).

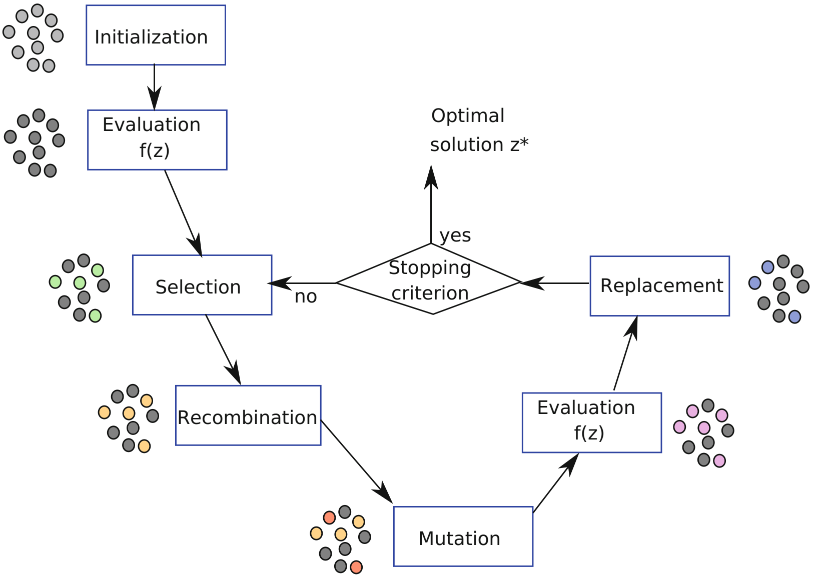


Fig. 5.9 Principle of genetic algorithm

In Evolution Strategies, the mutation is performed by sampling a normal centered multivariate law using a standard deviation σ which quantifies the mutation step. Different mechanisms have been proposed to adapt this mutation rate (Schwefel and Rudolph 1995; Bäck et al. 2013). The Covariance Matrix Adaptation-Evolution Strategy (CMA-ES) (Hansen et al. 1995) follows the paradigm of Evolution Strategy and appears to be very competitive to solve local and global optimization problems (Auger et al. 2009). In this algorithm (see Equation 5.53), dedicated mechanisms are used to update the normal law used to generate new individuals, based on an iterative refinement on the covariance matrix C and step size σ.

$$\mathbf{z}_i^{(k+1)} \sim \mathbf{m}^{(k)} + \sigma^{(k)} \mathscr{N}\left(0, \mathbf{C}^{(k)}\right) \qquad \text{for } i = 1, \ldots, \lambda \tag{5.53}$$

with $\mathscr{N}(0, \mathbf{C}^{(k)})$ the multivariate normal distribution, λ the number of individuals, $\mathbf{C}^{(k)} \in \mathbb{R}^{n \times n}$ the parameterized covariance matrix, $\mathbf{z}_i^{(k)} \in \mathbb{R}^n$ the ith individual of the population involved at the (k)th generation, $\mathbf{m}^{(k)} \in \mathbb{R}^n$ the mean of the *parents* individuals, and $\sigma^{(k)} \in \mathbb{R}$ the stepsize parameter. This allows to generate new individuals directly in promising areas of the search space. A large number of variants of CMA-ES exists ($1 + 1$, bi-pop, $\lambda - \mu$, etc.) but are not described here for the sake of conciseness. All the mechanisms describing the adaptation of covariance matrix and stepsize parameter can be found in Auger and Hansen (2012). Basically, these mecanisms allow *rotations of sampling* in the search space in order to converge more quickly to interesting zones (Figure 5.10).

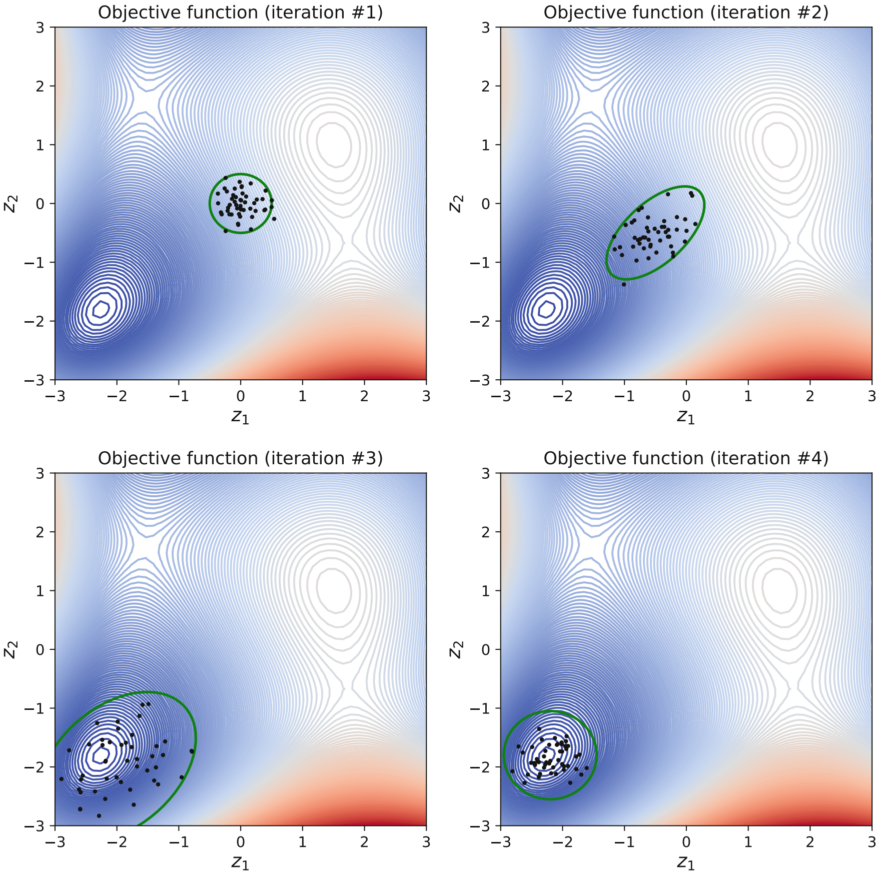


Fig. 5.10 Illustration of CMA-ES evolution

Handling of Uncertain Functions

Performance of evolutionary algorithms in an uncertain environment has been analyzed in Arnold (2001), Arnold and Beyer (2002), Branke et al. (2005), Vallejo and Corne (2016), Beyer (2000), and Rakshit et al. (2017). As explained in the comprehensive survey (Jin and Branke 2005), several mechanisms can be used for handling uncertainty in objective and constraint functions for evolutionary algorithms:

- Averaging the objective and constraint functions in order to reduce the uncertainty. This generally consists in evaluating several times the objective function and constraints in order to have mean estimates of these quantities of interest (Aizawa and Wah 1994; Branke and Schmidt 2003; Cantú-Paz 2004; Stagge 1998), or sampling in the vicinity of the individual locations as proposed in

Branke et al. (2001). This strategy is often very costly. This is even more true for large population and computationally costly simulation codes. Another idea consists in determining the number of required resampling using statistical tests such as t-test as proposed in Stagge (1998). The sampling size may be adapted depending on the variance of the estimations for the individuals and the magnitude of uncertainty over the search space (Aizawa and Wah 1993; Di Pietro et al. 2004; Cantú-Paz 2004; Stagge 1998). The mechanisms of evolutionary algorithms, such as the selection process can also be adapted to the handling on uncertainty by sampling as mentioned in Branke and Schmidt (2003).

- Using large population size. Because in the evolutionary algorithms, many individuals are generated in the interesting areas of the search space, increasing the population size may attenuate the effect of uncertainty. In Fitzpatrick and Grefenstette (1988), this strategy is compared to resampling technique and results show better results. Finally, several papers proposed to adaptively tune population size (Goldberg et al. 1991; Miller and Goldberg 1996; Miller et al. 1995) depending on the uncertainty.
- Adapting selection mechanisms. A threshold can be used to select children only if their fitness are greater than the parents by the threshold (Markon et al. 2001; Rudolph 1998; Branke and Schmidt 2003).

Moreover, these mechanisms can be adapted for each algorithm in order to enhance the performance of noise handling. For example, for CMA-ES, a two-step approach is proposed in Hansen et al. (2009b) for handling the uncertainty:

- firstly, the uncertainty is quantified by resampling the functions in order to analyze if the ranking of the individuals is impacted by the uncertainty.
- secondly, if in the first step the uncertainty is revealed to be too impactful, the stepsize parameter σ in Equation (5.53) is increased in order to enhance the diversity of the population, to decrease the impact of local uncertainty to the current mean point, to increase the move of the individuals, and to limit the premature convergence.

Swarm Algorithms

Principles

Swarm algorithms, of which the Particle Swarm Optimization (PSO) (Kennedy 2011) is the most known, are based on a set of individuals, named particles, which are randomly placed in the search space and evolve all along the optimization process in order to find local optima and global optimum. In PSO, each particle has in memory its best position and can dialog with the other particles within its neighborhood. The particles evolve taking into account their own past positions, and the best positions found by the other particles. From this information, the particle will follow a track which depends on different factors: its current velocity, its best position over the last iterations and the best position taken by the particles in its neighborhood (Figure 5.11).

PSO is based on two basic principles: the exploitation and the exploration. The exploration consists in searching global information on the whole search space, whereas exploitation consists in refining the search for information only in a restricted space around the particle current positions. At the iteration k, the particle i is defined by its position $\mathbf{z}_i^k$ and velocity $\mathbf{v}_i^k$ in the search space. In the most common PSO algorithm, the displacement in the search space at the iteration $k+1$ is defined by:

- the velocity:

$$\mathbf{v}_i^{k+1} = \psi \left(\mathbf{v}_i^k + c_1 r_1 \left(\mathbf{p}_i^k - \mathbf{z}_i^k \right) + c_2 r_2 \left(\mathbf{p}_g^k - \mathbf{z}_i^k \right) \right) \tag{5.54}$$

  with ψ the inertial factor, c_1 and c_2 two factors defining the exploration and exploitation, $r_1, r_2 \sim \mathscr{U}[0, 1]$ correspond to two random variables drawn from the uniform law [0, 1], $\mathbf{p}_g$ is the position of the best particle in the neighborhood, and $\mathbf{p}_i$ is the best past position of the current particle.
- the position:

$$\mathbf{z}_i^{k+1} = \mathbf{z}_i^k + \mathbf{v}_i^{k+1}. \tag{5.55}$$

The different hyperparameters of the algorithm have to be tuned depending on the optimization problem specifications (Bartz-Beielstein et al. 2007; Pal et al. 2012; Rada-Vilela et al. 2014).

Uncertainty Handling

PSO has been used in the case of uncertain functions and Parsopoulos and Vrahatis 2001 showed interesting convergence properties. The combination of the different particle positions allows to have an averaging phenomena and makes PSO less influenced by uncertain functions (Parsopoulos and Vrahatis 2001).

Specific mechanisms can be proposed in the framework of PSO to handle the noisy environment. In Pan et al. (2006) the authors used statistical tests on uncertain objective function in order to quantify the uncertainty, maintain the diversity of the swarm, and enhance the convergence of the algorithm.

In Rada-Vilela et al. (2015) the authors identify three statistics to represent the manners the uncertainty can perturb the displacement of the particles: the deception (when the best solution in the neighborhood of a particle is not the true best due to noise), the blindness (when the current particle cannot identify a better solution to improve its position), and the disorientation (when the particle replaces its position by a worse one). Several algorithms have been proposed to cope with these phenomena by using dedicated mechanisms in PSO. For example, in Cui et al. (2005), PSO with evaporation has been proposed to reduce the blindness. It

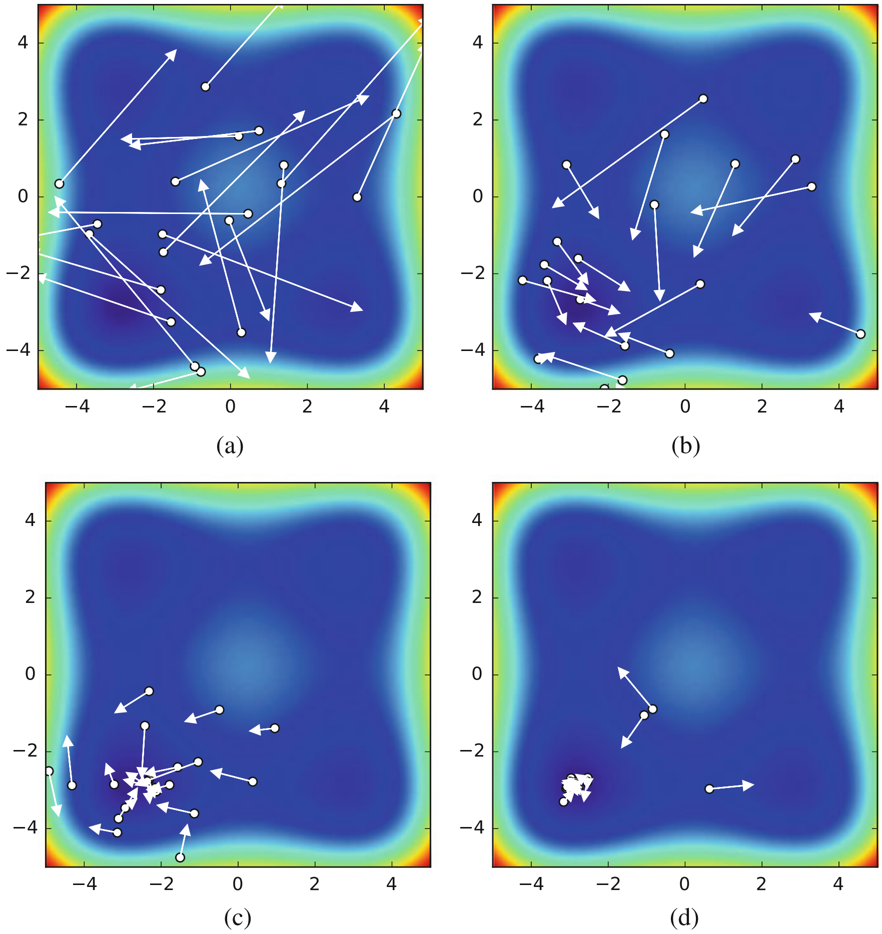


Fig. 5.11 Illustration of PSO evolution (Schwefel function (Laguna and Martí 2005)). (**a**) PSO at initialization. (**b**) PSO after 3 iterations. (**c**) PSO after 5 iterations. (**d**) PSO at convergence

consists in adding an evaporation factor for the particles that did not achieve to improve their position. In Rada-Vilela et al. (2014), several population statistics are proposed and combined with resampling techniques (named PSO with Equal Resampling (PSO-ER), PSO with Optimal Computing Budget Allocation (PSO-OCBA) or PSO with Equal Resampling and Allocations to the top-N solutions) to improve the convergence properties of classical PSO. These resampling methods aim at balancing between the effort made for improving the accuracy of the objective function and the effort made for exploring new search areas. In Rada-Vilela et al. (2015), PSO with average neighborhood (PSO-AN) is described as a single-evaluation method which does not involve resampling of the different particles at each iteration. This method uses the averaging on the neighborhood in order to improve the accuracy of the objective function calculated by each particle. Finally, in Kang et al. (2018), a method called Opposition-Based Hybrid PSO uses Opposition-Based Learning to improve the diversity of the different particles with a given probability. This mechanism has been coupled with PSO-AN, PSO-OCBA described previously.

Constraint Handling

In the vast majority of the cases, the constraints in the population-based algorithms are handled by a penalization factor on the objective function:

$$\tilde{f}(\mathbf{z}) = f(\mathbf{z}) + p(\mathbf{z}). \tag{5.56}$$

Different penalization functions exist in the literature. The most used are recalled in Table 5.1. These penalty functions depend on residual functions $res(\mathbf{z})$ which quantify the nonsatisfaction of the constraint. These residuals are equal to zero if all the constraints are satisfied. The main residual functions are described in Table 5.2.

The death penalty consists in rejecting all the individuals for which the constraints are not satisfied. This kind of strategies is very simple to implement but may make the algorithm convergence difficult to obtain if the feasible space is very restricted. Static penalty consists in considering a penalization which only depends on the value of residuals, with a fixed weight. In order to enhance the convergence of static strategy, evolutive strategy has been introduced. This strategy involves an adaptive evolution of the penalty weight in order to enforce the satisfaction of the constraints all along the optimization convergence depending on the constraint satisfaction observed for the population.

In most of the cases, the equality constraints are replaced by inequality ones by introducing a tolerance:

Table 5.1 Penalization functions

Strategy	Penalization function	Parameters
Death penalty	$p(\mathbf{z}) = \begin{cases} 0 & \text{if } res(\mathbf{z}) = 0 \\ \infty & \text{if } res(\mathbf{z}) > 0 \end{cases}$	None
Static	$p(\mathbf{z}) = K.res(\mathbf{z})$	K
Evolutive	$p(\mathbf{z}) = \lambda(k).res(\mathbf{z})$ with k the current iteration $\lambda(k+1) = \begin{cases} 1/\beta_1.\lambda(k) & \text{if case } \sharp 1 \\ \beta_2.\lambda(k) & \text{if case } \sharp 2 \\ \lambda(k) & \text{else} \end{cases}$ Case $\sharp$1: the best individuals over the last m iterations are feasible Case $\sharp$2: the best individuals over the last m generations have never been feasible	$\lambda(1), \beta_1, \beta_2, m$

Table 5.2 Examples of residual functions

Norm	Residual
l^1	$res(\mathbf{z}) = \sum_{i=1}^{n_{eq}} \lvert h_i(\mathbf{z})\rvert - \sum_{i=1}^{n_{in}} \min(0, g_i(\mathbf{z}))$
l^2	$res(\mathbf{z}) = \sqrt{\sum_{i=1}^{n_{eq}} \lvert h_i(\mathbf{z})\rvert^2 + \sum_{i=1}^{n_{in}} \min(0, g_i(\mathbf{z}))^2}$
l^∞	$res(\mathbf{z}) = \max(\lvert h_i(\mathbf{z})\rvert_{i=1,\ldots,n_{eq}}, \lvert \min(0, g_i(\mathbf{z}))\rvert_{i=1,\ldots,n_{in}})$

$$h(\mathbf{z}) = 0 \rightarrow |h(\mathbf{z})| - \epsilon \leq 0. \tag{5.57}$$

Other specific penalization-based methods, such as the Oracle penalization (Schlüter and Gerdts 2010), have been proposed in order to improve the convergence of the population-based algorithms but are not described here for the sake of conciseness.

Moreover, other approaches can be used instead of the penalization to handle the constraints, for example:

- the resolution of a multi-objective optimization problem (objective function and sum of constraint violations) (Mezura-Montes and Flores-Mendoza 2008),
- stochastic ranking (Runarsson and Yao 2000),
- constraint dominance (Coello and Montes 2002), etc.

The population-based algorithms can also have dedicated methods to handle the constraints:

- Reparation methods (evolutionary algorithms),
- Modification of the covariance matrix of CMA-ES in order to avoid the generation of individuals in the nonfeasible zones (Chocat et al. 2015)
- Creation of "leader" particles in PSO to guide the swarn in the feasible zone (Pulido and Coello 2004)
- etc.

5.3.3 Surrogate-Based Approaches

Principles

In the context of aerospace system design, the objective function and constraints often result from computationally expensive functions. These functions can be high-fidelity models and may take several hours to several days to produce one value for the objective and/or constraint functions. In this framework, the use of classical optimization algorithms (gradient-based, metaheuristics, etc.) cannot be affordable. This is strengthened by the handling of uncertainty which often involves resampling methods. Using surrogate models is an interesting way to have fast approximation of the objective function and constraints and to use them in the optimization context. These methods (Polynomial Regression, Neural-Network (Cochocki and

Unbehauen 1993), Support Vector Machines (Forrester et al. 2008), Radial-Basis Function (Gutmann 2001), Gaussian Processes (Jones et al. 1998), Moving Least-Squares (Toropov et al. 2005), etc.) consist in first building approximate models from a Design of Experiment and using them for predicting optimal values of objective and constraints (see Chapter 3 for more details on surrogate models). Two kinds of approaches can be found: either the surrogate model is built before the optimization process and is not updated during the optimization, either the surrogate model is refined at each iteration of the optimization process, in order to improve its accuracy either in the zone of large uncertainty (exploration) either in the vicinity of the current optimum (exploitation). A very large number of surrogate models has been proposed in the literature. The interested reader can consult (Forrester et al. 2008; Forrester and Keane 2009) for complete surveys. In this chapter, we only focus on a specific one which is adapted to the handling of uncertainty and will be used in the following chapters.

Deterministic Global Optimization Algorithm Based on Gaussian Process (Efficient Global Optimization)

Efficient Global Optimization in a Nutshell

All the concepts of this section lie on the Gaussian process (or Kriging) theory (see Chapter 3 for more details on Gaussian process). First, let consider an unconstrained deterministic global optimization problem:

$$\min \quad f(\mathbf{z}) \tag{5.58}$$

$$\text{w.r.t.} \quad \mathbf{z} \tag{5.59}$$

$$\mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max}. \tag{5.60}$$

The black-box function $f(\cdot)$ is modeled as a path of a Gaussian process (GP) $F(\cdot)$ to be characterized for any $\mathbf{z} \in \mathbb{R}^n$. Let $\mathscr{X}_q = \{\mathbf{z}_{(1)}, \dots, \mathbf{z}_{(q)}\}$ the current DoE and $\mathscr{Y}_q = \{y_{(1)} = f(\mathbf{z}_{(1)}), y_{(2)} = f(\mathbf{z}_{(2)}), \dots, y_{(q)} = f(\mathbf{z}_{(q)})\}$ the corresponding (scalar) responses. Several strategies have been studied to use the Kriging model for estimating the minimum of the function on a limited budget of evaluations. One trivial way to use Kriging to optimize a computationally expensive black-box function could be to minimize the Kriging mean prediction. However, this strategy does not take into account the variance information, and is often not efficient (Jones 2001) because the zones with large uncertainty are not considered. Two main infill criteria have been proposed in the literature: the *Probability of Improvement* and the *Expected Improvement*. The main idea here is to process the Kriging surrogate to propose a point (determined by maximizing an *improvement* function) that is pertinent to evaluate the expensive function in order to improve the estimation of the global minimum.

As described in Chapter 3, within the GP framework, the exact function $f(\cdot)$ to minimize is considered as the realization of a Gaussian variable with mean $\widehat{f}(\mathbf{z})$, and standard deviation $\widehat{s}(\mathbf{z})$. The improvement from the current samples $y_{\text{min}} = \min\{y_{(1)}, \ldots, y_{(q)}\}$ is defined as follows:

$$I(\mathbf{z}) = \max(y_{min} - F(\mathbf{z}), 0) \tag{5.61}$$

The Probability of Improvement (PI) is defined as the probability that $F(\mathbf{z}) \leq y_{\text{min}}$,

$$\begin{aligned} \text{PI}(\mathbf{z}) &= \mathbb{P}[F(\mathbf{z}) \leq y_{\text{min}}] = \frac{1}{\widehat{s}(\mathbf{z})\sqrt{2\pi}} \int_{-\infty}^{y_{min}} \exp\left(\frac{-\left(t - \widehat{f}(\mathbf{z})\right)^2}{2\widehat{s}(\mathbf{z})^2} \right) \mathrm{d}t \\ \text{PI}(\mathbf{z}) &= \Phi\left(\frac{y_{\text{min}} - \widehat{f}(\mathbf{z})}{\widehat{s}(\mathbf{z})} \right), \end{aligned} \tag{5.62}$$

with $\Phi(\cdot)$ the Cumulative Density Function of the normal distribution (Kushner 1964). The points to refine the model are determined by maximizing this criterion. However, these points are often located near the current best point, where the probability is high. Consequently, this criterion is not relevant for global exploration and has to be used for local refinement near the estimated minimum of the function.

Expected Improvement (EI) (Jones et al. 1998) has been defined in order to improve the balance between exploration and exploitation. This criterion takes into account the amount of improvement expected instead of only finding zones in which there is a large probability of improving the function.

Then, the mathematical expectation of the improvement is defined in Equation (5.61) with $\phi(\cdot)$ the normal PDF.

$$\begin{aligned} \text{EI}(\mathbf{z}) &= \mathbb{E}[\max(y_{\text{min}} - F(\mathbf{z}), 0.))] \\ \text{EI}(\mathbf{z}) &= \left(y_{\text{min}} - \widehat{f}(\mathbf{z})\right) \Phi\left(\tfrac{y_{\text{min}} - \widehat{f}(\mathbf{z})}{\widehat{s}(\mathbf{z})}\right) + \widehat{s}(\mathbf{z})\phi\left(\tfrac{y_{\text{min}} - \widehat{f}(\mathbf{z})}{\widehat{s}(\mathbf{z})}\right) \end{aligned} \tag{5.63}$$

This equation consists of two parts: the first one is relative to the determination of the zone in the search space in which there is potential improvement of the function (scaled by discrepancy between the prediction and the current minimum). The second part allows at exploring the zones for which the uncertainty of the model is high. The expected improvement allows to balance the part relative to exploration and the part dedicated to exploitation in the refinement criterion.

The *Efficient Global Optimization* (EGO) is an iterative process which proceeds (Jones et al. 1998) as follows (see Figure 5.12):

1. Start by defining a DoE $\mathscr{Z}_q = \left\{\mathbf{z}_{(1)}, \mathbf{z}_{(2)}, \ldots, \mathbf{z}_{(q)}\right\}$ and evaluate $\mathscr{Y}_q = \left\{y_{(1)} = f(\mathbf{z}_{(1)}),\quad y_{(2)} = f(\mathbf{z}_{(2)}), \ldots, y_{(q)} = f(\mathbf{z}_{(q)})\right\}$. Learn the GP hyperparameters (training of the GP model by maximization of the log likelihood, see Chapter 3).
2. Find the point to add to the DoE $\mathbf{z}_{\text{new}}$ by maximizing the infill criterion, either PI with Equation (5.62) or EI with Equation (5.63), and compute the expensive black-box function value y_{new}.
3. Append $\mathbf{z}_{\text{new}}$ to $\mathscr{Z}_q$ and y_{new} to $\mathscr{Y}_q$ giving the new training sets: $\mathscr{Z}_{q+1}$ and $\mathscr{Y}_{q+1}$. Learn the GP hyperparameters.
4. Repeat the process from Step 2 until a stopping criterion is met. Classically the maximal number of function evaluations or a minimal threshold on infill criterion can be used as stopping criterion.

The EI criterion is particularly well suited for global optimization with EGO as explained in Schonlau et al. (1996), Forrester et al. (2008), and Sasena (2002). Figure 5.13 shows the two described infill criteria on a one-dimensional toy case. These infill criteria may present a large number of local maxima. Therefore, the infill optimization step of EGO has to be performed by an appropriate global search algorithm (e.g., CMA-ES, genetic algorithm, multi-start gradient-based algorithm).

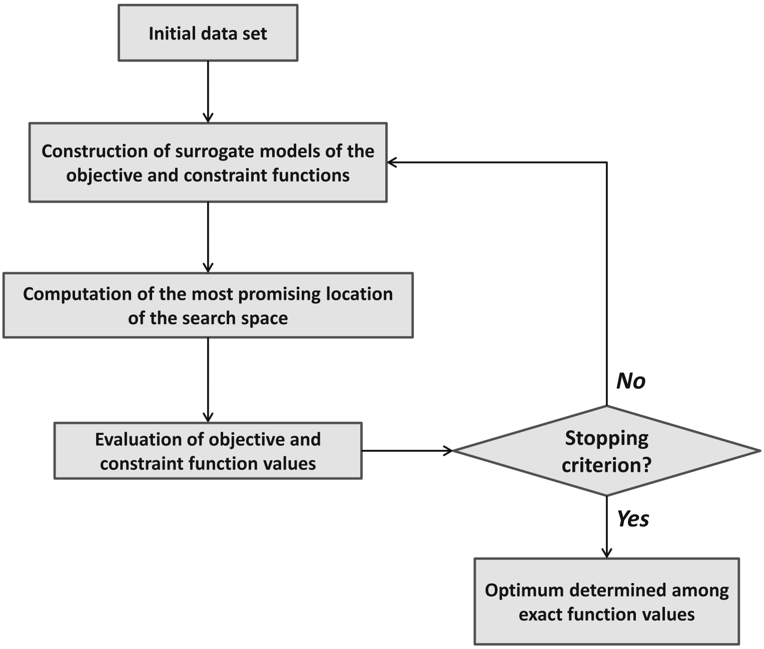

Initial data set
Construction of surrogate models of the objective and constraint functions
Computation of the most promising location of the search space
Evaluation of objective and constraint function values
Stopping criterion?
No
Yes
Optimum determined among exact function values

Fig. 5.12 Gaussian process based optimization algorithm

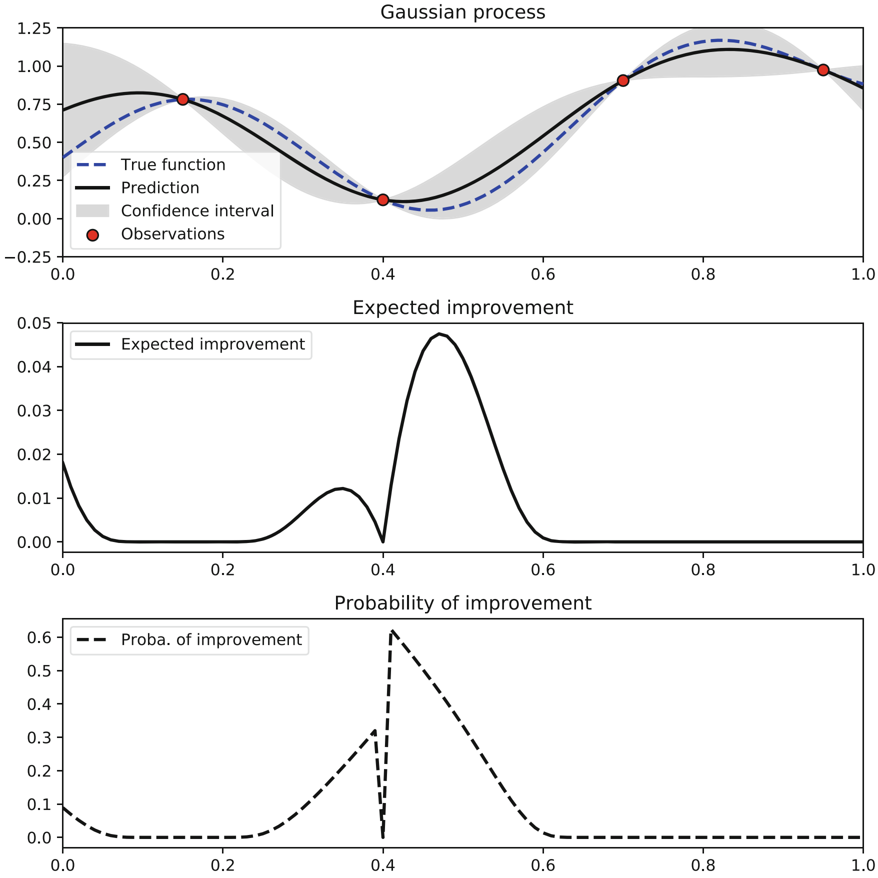


Fig. 5.13 Examples of PI and EI infill criteria

Constraint Handling

Let us add constraints to the previous optimization problem.

$$\min \quad f(\mathbf{z}), \tag{5.64}$$

$$\text{w.r.t.} \quad \mathbf{z} \tag{5.65}$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{z}) \leq \mathbf{0} \tag{5.66}$$

$$\mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max}. \tag{5.67}$$

Here, the vector of constraint functions $\mathbf{g}(\cdot)$ is also assumed to be computationally costly. In this context, each constraint may be approximated by a Gaussian process model. The main difference with respect to the unconstrained case consists of the refinement phase. In the constrained optimization problem, the infill criteria (Step 2 of EGO) has to take into account the approximated constraints.

The first strategy consists in considering the so called *Probability of Feasibility* (Parr et al. 2012), defined for the constraint i as follows:

$$\mathrm{PF}_i(\mathbf{z}) = \Phi\left(-\frac{\widehat{g}_i(\mathbf{z})}{\widehat{s}_{g_i}(\mathbf{z})}\right). \tag{5.68}$$

The constrained optimization problem of EI at Step 2 of EGO is transformed into an unconstrained one by multiplying the value of EI by the probability that the point is feasible (Schonlau et al. 1998). The infill criterion becomes

$$\mathrm{EI}(\mathbf{z}) \times \mathrm{PF}(\mathbf{z}) = \mathrm{EI}(\mathbf{z}) \prod_{i=1}^{m} \mathrm{PF}_i(\mathbf{z}), \tag{5.69}$$

with m the number of constraints. The infill criterion will be close to zero where there is a very low probability of feasibility for any of the constraints.

The second strategy, called *constrained EI* (Sasena et al. 2001), is very simple and consists in solving the EI optimization as a constrained optimization problem, using the prediction (Kriging mean) of the constraints. This strategy is highly dependent on the accuracy of the constraint surrogate models. If the constraints are active, this strategy will add points close to the value $\widehat{\mathbf{g}}(\mathbf{z}) = \mathbf{0}$:

$$\max_{\mathbf{z}} \quad \mathrm{EI}(\mathbf{z}), \tag{5.70}$$

$$\text{w.r.t.} \quad \mathbf{z} \tag{5.71}$$

$$\text{s.t.} \quad \widehat{\mathbf{g}}(\mathbf{z}) \leq 0 \tag{5.72}$$

$$\mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max}. \tag{5.73}$$

To solve this problem, an auxiliary global optimization algorithm allowing to handle constraints is required.

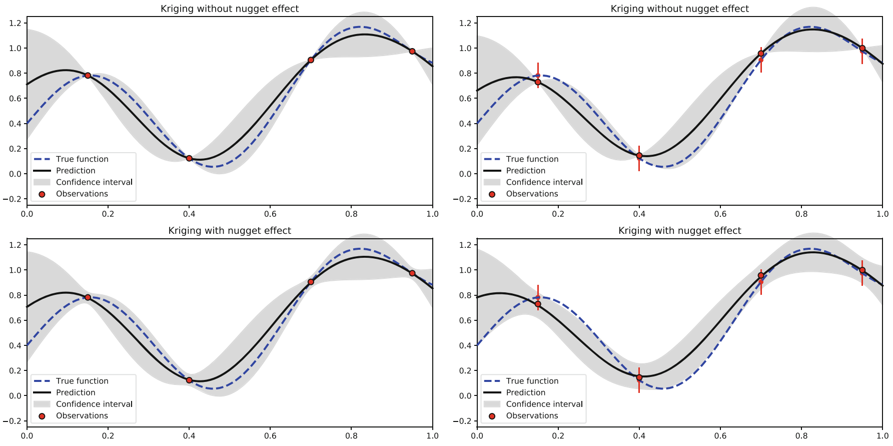
The last strategy presented in this chapter is called *Expected Violation* (EV) and formulates the constraints as follows (Audet et al. 2000):

$$\mathrm{EV}_i(\mathbf{z}) = (\widehat{g}_i(\mathbf{z}) - 0)\varPhi\left(\frac{\widehat{g}_i(\mathbf{z}) - 0}{\widehat{s}_{g_i}(\mathbf{z})}\right) + \widehat{s}_{g_i}(\mathbf{z})\phi\left(\frac{\widehat{g}_i(\mathbf{z}) - 0}{\widehat{s}_{g_i}(\mathbf{z})}\right) \tag{5.74}$$

EV is quite similar to the EI criterion. It is high in regions where the constraints have a large probability to be saturated or where there is a high uncertainty on the constraint models. The auxiliary optimization problem to solved reads

$$\begin{aligned} \max_{\mathbf{z}} \quad & \mathrm{EI}(\mathbf{z}), \\ \text{s.t.} \quad & \mathrm{EV}_i(\mathbf{z}) \leq t_{\mathrm{EV}} \text{ for } i = 1, \ldots, m \\ & \mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max} \end{aligned} \tag{5.75}$$

with t_{EV} a threshold defined by the user.



(a) (b)

Fig. 5.14 Regression using Gaussian processes with and without noise (top: models without nugget, bottom: models including nugget). (**a**) Deterministic function. (**b**) Handling of noise

Uncertainty Handling

Gaussian processes define a valuable framework to handle uncertainty directly in the optimization process. This is mainly possible using the *nugget* effect of Gaussian processes (see Chapter 3). It corresponds to the addition of a term representing the observation variance in the diagonal of the covariance matrix used in Kriging. Considering this effect, the Kriging model does not strictly interpolate the data anymore and can take into account some *noise* in the collected data due to the presence of uncertainty (in the prediction of uncertainty measures, for example, see Figure 5.14). Example of practical case of using nugget can be found in Le Riche et al. (2009). Nugget can be fit as other Kriging hyperparameters as explained in Chapter 3.

The nugget effect can be used in different ways to handle uncertainty (Picheny and Ginsbourger 2014). The first one is to replace in the EI the $y_{\min}$ term by the best collected mean. The second is to add nugget and use alternative infill criteria such as the Augmented Expected Improvement (AEI) (Huang et al. 2006) (see Equation 5.76) or the Expected Quantile Improvement (EQI) (Picheny et al. 2013a) (see Equation 5.77).

$$\begin{aligned}\mathrm{AEI}(\mathbf{z}) = \left(y_{\min} - \hat{f}(\mathbf{z})\right) \Phi\left(\frac{y_{\min} - \hat{f}(\mathbf{z})}{\hat{s}(\mathbf{z})}\right) \\ + \hat{s}(\mathbf{z})\phi\left(\frac{y_{\min} - \hat{f}(\mathbf{z})}{\hat{s}(\mathbf{z})}\right) \cdot \left(1 - \frac{\sigma}{\sqrt{\sigma^2 + \hat{s}^2(\mathbf{z})}}\right)\end{aligned} \tag{5.76}$$

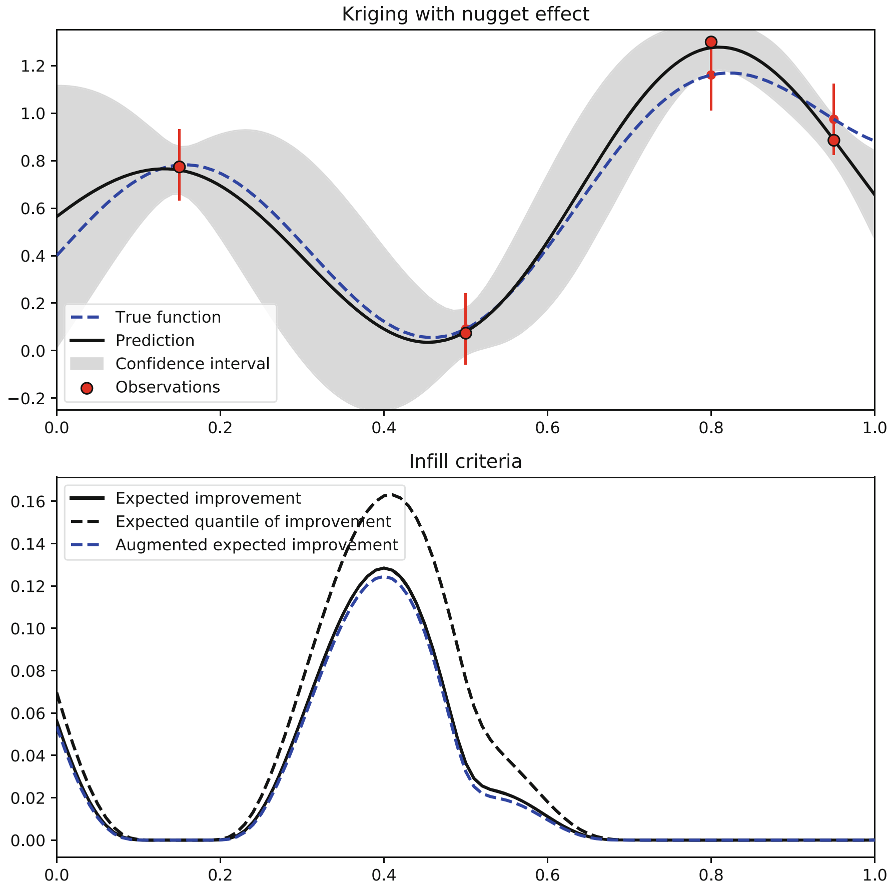


Fig. 5.15 Expected Improvement, Augmented Expected Improvement, and Expected Quantile Improvement)

$$\mathrm{EQI}(\mathbf{z}) = \left(q_{\min} - \hat{q}(\mathbf{z})\right) \Phi\left(\frac{q_{\min} - \hat{q}(\mathbf{z})}{\hat{s}_{q(\mathbf{z})}}\right) + \hat{s}_{q(\mathbf{z})} \phi\left(\frac{q_{\min} - \hat{q}(\mathbf{z})}{\hat{s}_{q(\mathbf{z})}}\right) \tag{5.77}$$

with σ the nugget, $q_{\min}$ the minimum quantile calculated on the training points, $\hat{q}(\mathbf{z})$ $\hat{s}_q(\mathbf{z})$ the estimations of the prediction and standard deviation of the quantile. The quantities involved in these equations can be computed analytically. The interested reader can consult (Picheny et al. 2013a) for the formulations. For more details about the different infill criteria in noisy environment, the interested reader can consult (Picheny et al. 2013b; Huang et al. 2006). The difference between the Expected Improvement and the different infill criteria is represented on a toy case function in Figure 5.15.

In Janusevskis and Le Riche (2013), in the case the uncertainty can be controlled, the authors proposed an alternative approach consisting in building a surrogate model in the joint space $(\mathbf{z}, \mathbf{U})$. The statistical measures can be computed on the surrogate model to build another surrogate of the uncertainty measures, depending only on $\mathbf{z}$. Then the first method (EI noisy) is applied to find infill candidate in both $\mathbf{z}$ and $\mathbf{U}$ space ($\mathbf{U}$ is sampled from its PDF). This algorithm has been compared to Monte Carlo based algorithms on analytic functions in Janusevskis and Le Riche (2013).

5.4 Brief Review on Optimization Methods for Uncertainty Described with Non-Probabilistic Framework

As mentioned in Chapter 2, the uncertainties can be described with other formalisms than the probability theory. In this case, the uncertainty propagation technique, optimization problem formulation, and the optimization algorithms have to be adapted on the representation of uncertainty. In this section, different problem formulations and optimization algorithms are briefly presented for uncertainty described with interval, Dempster–Shafer theory or fuzzy logic.

5.4.1 Optimization Under Uncertainty Described with Interval Formalism

When uncertainty is described through interval, the optimization can often be resumed to a worst-case optimization, when the designer wants to optimize the objective function subject to the "worst effect" that can be induced by uncertainty. This is often called a minimax problem in literature (Rustem and Howe 2009; Hansen and Walster 2003; Marzat et al. 2013; Shimizu and Aiyoshi 1980), or bi-level programming (Tsoukalas et al. 2009). Review of formulations and algorithms can be found in Du and Pardalos (1995).

$$\min_{\mathbf{z}} \max_{\mathbf{U}} \quad f(\mathbf{z}, \mathbf{U}), \tag{5.78}$$

$$\text{w.r.t.} \quad \mathbf{z} \tag{5.79}$$

$$\text{s.t.} \quad \mathbf{z}_{\text{min}} \leq \mathbf{z} \leq \mathbf{z}_{\text{max}} \tag{5.80}$$

$$\mathbf{U}_{\text{min}} \leq \mathbf{U} \leq \mathbf{U}_{\text{max}} \tag{5.81}$$

with $\mathbf{U} = \left\{U^{(1)}, \ldots, U^{(n)}\right\}, \forall i, U_{i_{\text{min}}} \leq U^{(i)} \leq U_{i_{\text{max}}}$ the vector of uncertain variables. Numerous algorithms have been proposed in the literature to solve such problems. Several of them need continuous or differentiable properties to use interval algebra or suppose the existence of subgradient. Evolutionary algorithms have been extensively used for such problems (Cramer et al. 2009; Lung and Dumitrescu 2011) or surrogate-based model (Marzat et al. 2013; Zhou and Zhang 2010; Ur Rehman and Langelaar 2015) when the objective function is numerically costly. Distinction can be made between the methods solving directly the previous problem and methods that split the optimization problem into two subproblems depending either on the **z**-space or on the **u**-space. These later are called *relaxation methods* (Shimizu and Aiyoshi 1980).

5.4.2 Optimization with Uncertain Variables Described Using Evidence Theory

Like the uncertainties described using the probability formalism, the uncertainty measures introduced in the Dempster–Shafer theory (belief, plausibility) can be used directly as uncertain measures for the constraint in order to formulate the uncertain optimization problem (Agarwal et al. 2003; Mourelatos and Zhou 2006). In Huang et al. (2013), the authors performed a review of application of Dempster–Shafer theory in design optimization. The Evidence-Based Design Optimization problem (EBDO) can be formulated as follows:

$$\min_{\mathbf{z}} \quad f(\mathbf{z}, \bar{\mathbf{U}}), \tag{5.82}$$

$$\text{w.r.t.} \quad \mathbf{z} \tag{5.83}$$

$$\text{s.t.} \quad \mathrm{Pl}(g_i(\mathbf{z}, \mathbf{U})) \leq T, \ \ i = 1, \ldots, m \tag{5.84}$$

$$\mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max} \tag{5.85}$$

written here using the use of plausibility measure for the constraints with T a given threshold and $\bar{\mathbf{U}}$ the nominal values of the uncertain variables **U** (Huang et al. 2017). To solve this kind of problems, gradient-based algorithms (Alyanak et al. 2008; Fan et al. 2016), evolutionary algorithms (Salehghaffari et al. 2013; Srivastava and Deb 2011; Su et al. 2016) can be employed. In order to reduce the computational burden induced by both the propagation of uncertainty coupled with optimization, response surfaces as proposed in Salehghaffari et al. (2013) and Cao et al. (2018) or multi-level approaches (Huang et al. 2017; Zhang et al. 2018) can be used.

5.4.3 Optimization Under Uncertainty Using Possibility Theory

When the uncertain variables **U** are described using possibilistic framework (or fuzzy-set), the Possibility Optimization Design Problem (PBDO) can be formulated using a fuzzy measure on the constraint, for example, an a-cut (see Chapter 2) (Mourelatos and Zhou 2005; Choi et al. 2005).

$$\min_{\mathbf{z}} \quad f(\mathbf{z}), \tag{5.86}$$

$$\text{w.r.t.} \quad \mathbf{z} \tag{5.87}$$

$$\text{s.t.} \quad \mathrm{Pos}(g_i(\mathbf{z}, \mathbf{U})) \leq a, \ \ i = 1, \ldots, m \tag{5.88}$$

$$\mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max} \tag{5.89}$$

with Pos$(\cdot)$ the possibility measure and a the level of the a-cut (see Chapter 2 for the description of the possibility measure and a-cut). This optimization problem often consists of a double loop: the calculation of the a-cut in an inner loop and the optimization purpose in an outer loop. Sequential algorithm has been proposed in Zhou and Mourelatos (2006) to solve this problem with reduced computational cost. Bayesian optimization (use of Gaussian process) can also be used in order to handle the computational cost as described in Du et al. (2006a,b). Possibilistic design optimization has been extensively studied in literature (see, for example, (Choi et al. 2004, 2005; Yin et al. 2018)). The interested reader can consult (He et al. 2007) for a review of the use of possibility theory and its application to design optimization.

5.5 Summary

In this chapter the optimization under uncertainty, with a focus on uncertainty described using probability theory, has been briefly described. First, the different problem formulations (robust, reliability-based, and robust and reliability-based) have been presented and their conservativeness have been illustrated on a toy case. Then, an overview of the different optimization algorithms (gradient, population-based or surrogate model-based) has been proposed, focusing on the handling of uncertainty and constraints. Finally, extensions to uncertainties described with other formalisms than probability theory have been briefly described. The different choices the designer has to make when dealing with optimization under uncertainty are illustrated in Figure 5.16.

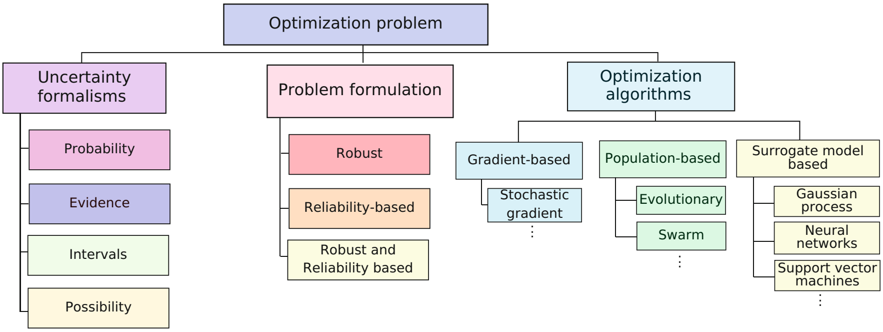


Fig. 5.16 Summary of different key steps when dealing with optimization under uncertainty: selection of the uncertainty formalism, formulation of the optimization problem and choice of the appropriate optimization algorithm

References

Agarwal, H., Renaud, J., and Preston, E. (2003). Trust region managed reliability based design optimization using evidence theory. In *44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Northfolk, VA, USA*.

Aizawa, A. and Wah, B. (1994). Scheduling of genetic algorithms in a noisy environment. *Evolutionary Computation*, 2(2):97–122.

Aizawa, A. N. and Wah, B. W. (1993). Dynamic control of genetic algorithms in a noisy environment. In *Fifth International Conference on Genetic Algorithms, Urbana-Champaign, IL, USA*.

Alyanak, E., Grandhi, R., and Bae, H.-R. (2008). Gradient projection for reliability-based design optimization using evidence theory. *Engineering Optimization*, 40(10):923–935.

Andrieu, L. (2004). *Optimisation sous contrainte en probabilité (in French)*. PhD thesis, Ecole des Ponts ParisTech.

Andrieu, L., Cohen, G., and Vázquez-Abad, F. J. (2007). Stochastic programming with probability constraints. *arXiv preprint arXiv:0708.0281*.

Andrieu, L., Cohen, G., and Vázquez-Abad, F. J. (2011). Gradient-based simulation optimization under probability constraints. *European Journal of Operational Research*, 212(2):345–351.

Arnaud, R. (2014). *Etude d'un problème d'optimisation en aéroélasticité avec incertitudes (in French)*. PhD thesis, Ecole Centrale Paris.

Arnaud, R. and Poirion, F. (2014). Optimization of an uncertain aeroelastic system using stochastic gradient approaches. *Journal of Aircraft*, 51(3):1061–1066.

Arnold, D. V. (2001). Evolution strategies in noisy environments—a survey of existing work. In *Theoretical aspects of evolutionary computing*, pages 239–249. Springer.

Arnold, D. V. and Beyer, H.-G. (2002). *Noisy optimization with evolution strategies*, volume 8. Springer Science & Business Media.

Arrow, K. J., Hurwicz, L., and Uzawa, H. (1958). *Studies in linear and non-linear programming*. Cambridge Univ. Press.

Audet, C., Booker, A. J., Dennis Jr, J. E., Frank, P. D., and Moore, D. W. (2000). A surrogate-model-based method for constrained optimization. In *8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, USA*.

Auger, A. and Hansen, N. (2012). Tutorial CMA-ES: evolution strategies and covariance matrix adaptation. In *14th International Conference on Genetic and Evolutionary Computation (GECCO' 12), Philadelphia, PA, USA*.

Auger, A., Hansen, N., Zerpa, J. P., Ros, R., and Schoenauer, M. (2009). Experimental comparisons of derivative free optimization algorithms. In *International Symposium on Experimental Algorithms, Dortmund, Germany*.

Bäck, T., Foussette, C., and Krause, P. (2013). *Contemporary evolution strategies*. Springer.

Bartz-Beielstein, T., Blum, D., and Branke, J. (2007). Particle swarm optimization and sequential sampling in noisy environments. In *Metaheuristics*, pages 261–273. Springer.

Beck, A. T., Gomes, W. J., Lopez, R. H., and Miguel, L. F. (2015). A comparison between robust and risk-based optimization under uncertainty. *Structural and Multidisciplinary Optimization*, 52(3):479–492.

Beyer, H.-G. (2000). Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice. *Computer methods in applied mechanics and engineering*, 186(2):239–267.

Boussaïd, I., Lepagnot, J., and Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, 237:82–117.

Branke, J. (1998). Creating robust solutions by means of evolutionary algorithms. In Eiben, A., Back, T., Schoenauer, M., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature, PPSN V*, volume 1498 of *Lecture Notes in Computer Science*, pages 119–128. Springer Berlin Heidelberg.

Branke, J., Chick, S. E., and Schmidt, C. (2005). New developments in ranking and selection: an empirical comparison of the three main approaches. In *37th conference on Winter simulation, Orlando, FL, USA*.

Branke, J. and Schmidt, C. (2003). Selection in the presence of noise. In *Genetic and Evolutionary Computation Conference (GECCO' 2003), Chicago, IL, USA*, pages 766–777. Springer.

Branke, J., Schmidt, C., and Schmeck, H. (2001). Efficient fitness estimation in noisy environments. In *3rd Annual Conference on Genetic and Evolutionary Computation, San Francisco, CA, USA*.

Cantú-Paz, E. (2004). Adaptive sampling for noisy problems. In Deb, K., editor, *Genetic and Evolutionary Computation, GECCO 2004*, volume 3102 of *Lecture Notes in Computer Science*, pages 947–958. Springer Berlin Heidelberg.

Cao, L., Liu, J., Han, X., Jiang, C., and Liu, Q. (2018). An efficient evidence-based reliability analysis method via piecewise hyperplane approximation of limit state function. *Structural and Multidisciplinary Optimization*, pages 1–13.

Chocat, R., Brevault, L., Balesdent, M., and Defoort, S. (2015). Modified Covariance Matrix Adaptation–Evolution Strategy algorithm for constrained optimization under uncertainty, application to rocket design. *International Journal for Simulation and Multidisciplinary Design Optimization*, 6:A1.

Choi, K., Du, L., and Youn, B. (2004). A new fuzzy analysis method for possibility-based design optimization. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, USA*.

Choi, K. K., Du, L., and Youn, B. D. (2005). Integration of reliability-and possibility-based design optimizations using performance measure approach. Technical report, SAE Technical Paper.

Cochocki, A. and Unbehauen, R. (1993). *Neural networks for optimization and signal processing*. John Wiley & Sons, Inc.

Coello, C. A. C. and Montes, E. M. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, 16(3):193–203.

Cohen, G. and Culioli, J.-C. (1994). Optimisation stochastique sous contraintes en espérance (in French). *Rapport interne Centre Automatique et Systèmes, Ecole des Mines de Paris*.

Cramer, A. M., Sudhoff, S. D., and Zivi, E. L. (2009). Evolutionary algorithms for minimax problems in robust design. *IEEE Transactions on Evolutionary Computation*, 13(2):444–453.

Cui, X., Hardin, C. T., Ragade, R. K., Potok, T. E., and Elmaghraby, A. S. (2005). Tracking non-stationary optimal solution by particle swarm optimizer. In *6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks, Towson, MD, USA*. IEEE.

Davis, L. (1991). *Handbook of genetic algorithms*. Van Nostrand Reinhold Co., New York, USA.

Désidéri, J.-A. (2014). Multiple-gradient descent algorithm for pareto-front identification. In *Modeling, Simulation and Optimization for Science and Technology*, pages 41–58. Springer.

Di Pietro, A., While, R. L., and Barone, L. (2004). Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions. In *IEEE Congress on Evolutionary Computation, Portland, RO, USA*.

Du, D.-Z. and Pardalos, P. M. (1995). *Minimax and applications*. Springer.

Du, L., Choi, K. K., and Youn, B. D. (2006a). Inverse possibility analysis method for possibility-based design optimization. *AIAA Journal*, 44(11):2682–2690.

Du, L., Choi, K. K., Youn, B. D., and Gorsich, D. (2006b). Possibility-based design optimization method for design problems with both statistical and fuzzy input data. *Journal of Mechanical Design*, 128(4):928–935.

Fan, S., Jiang, C., Zhang, Z., and Liu, J. (2016). A reliability-based design optimization method for structures using evidence theory. *Scientia Sinica Technologica*, 46(7):706–716.

Fitzpatrick, J. M. and Grefenstette, J. J. (1988). Genetic algorithms in noisy environments. *Machine learning*, 3(2–3):101–120.

Forrester, A., Sobester, A., and Keane, A. (2008). *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons.

Forrester, A. I. and Keane, A. J. (2009). Recent advances in surrogate-based optimization. *Progress in aerospace sciences*, 45(1–3):50–79.

Gabrel, V., Murat, C., and Thiele, A. (2014). Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483.

Gardner, W. (1984). Learning characteristics of stochastic-gradient-descent algorithms: A general study, analysis, and critique. *Signal Processing*, 6(2):113–133.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition.

Goldberg, D. E., Deb, K., and Clark, J. H. (1991). Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6:333–362.

Gutmann, H.-M. (2001). A radial basis function method for global optimization. *Journal of Global Optimization*, 19(3):201–227.

Hansen, E. and Walster, G. W. (2003). *Global optimization using interval analysis: revised and expanded*. CRC Press.

Hansen, N., Müller, S. D., and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation*, 11(1):1–18.

Hansen, N., Niederberger, A., Guzzella, L., and Koumoutsakos, P. (2009a). 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.

Hansen, N., Niederberger, A. S., Guzzella, L., and Koumoutsakos, P. (2009b). 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.

Hansen, N., Ostermeier, A., and Gawelczyk, A. (1995). On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In *6th International Conference on Genetic Algorithms, Pittsburgh, PA, USA*.

He, L.-P., Huang, H.-Z., Du, L., Zhang, X.-D., and Miao, Q. (2007). A review of possibilistic approaches to reliability analysis and optimization in engineering design. In *International Conference on Human-Computer Interaction*, pages 1075–1084. Springer.

Himmelblau, D. M. (1972). *Applied nonlinear programming*. McGraw-Hill Companies.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* University of Michigan Press.

Huang, D., Allen, T. T., Notz, W., and Zheng, N. (2006). Global optimization of stochastic black-box systems via sequential Kriging meta-models. *Journal of Global Optimization*, 34(3):441–466.

Huang, H.-Z., He, L., Liu, Y., Xiao, N.-C., Li, Y.-F., and Wang, Z. (2013). Possibility and evidence-based reliability analysis and design optimization. *American Journal of Engineering and Applied Sciences*, 6:95–136.

Huang, Z., Jiang, C., Zhang, Z., Fang, T., and Han, X. (2017). A decoupling approach for evidence-theory-based reliability design optimization. *Structural and Multidisciplinary Optimization*, 56(3):647–661.

Janusevskis, J. and Le Riche, R. (2013). Simultaneous kriging-based estimation and optimization of mean response. *Journal of Global Optimization*, 55(2):313–336.

Jin, Y. and Branke, J. (2005). Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on evolutionary computation*, 9(3):303–317.

Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492.

Kang, Q., Xiong, C., Zhou, M., and Meng, L. (2018). Opposition-based hybrid strategy for particle swarm optimization in noisy environments. *IEEE Access*, 6:21888–21900.

Kennedy, J. (2011). Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer.

Klamroth, K., Köbis, E., Schöbel, A., and Tammer, C. (2017). A unified approach to uncertain optimization. *European Journal of Operational Research*, 260(2):403–420.

Kushner, H. J. (1964). A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Fluids Engineering*, 86(1):97–106.

Laguna, M. and Martí, R. (2005). Experimental testing of advanced scatter search designs for global optimization of multimodal functions. *Journal of Global Optimization*, 33(2):235–255.

Le Riche, R., Picheny, V., Meyer, A., Kim, N.-H., and Ginsbourger, D. (2009). Gears design with shape uncertainties using controlled Monte Carlo simulations and kriging. In *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Palm Springs, CA, USA*.

Lelièvre, N., Beaurepaire, P., Mattrand, C., Gayton, N., and Otsmane, A. (2016). On the consideration of uncertainty in design: optimization-reliability-robustness. *Structural and Multidisciplinary Optimization*, 54(6):1423–1437.

Lung, R. I. and Dumitrescu, D. (2011). A new evolutionary approach to minimax problems. In *2011 IEEE Congress on Evolutionary Computation (CEC), New Orleans, LA, USA*.

Markon, S., Arnold, D. V., Back, T., Beielstein, T., and Beyer, H.-G. (2001). Thresholding-a selection operator for noisy ES. In *2001 IEEE Congress on Evolutionary Computation (CEC), Seoul, South Korea*.

Marzat, J., Walter, E., and Piet-Lahanier, H. (2013). Worst-case global optimization of black-box functions through kriging and relaxation. *Journal of Global Optimization*, 55(4):707–727.

Mercier, Q. (2018). *Optimisation multicritere sous incertitudes : un algorithme de descente stochastique (in French)*. PhD thesis, Université Sofia Antipolis.

Mercier, Q., Poirion, F., and Désidéri, J.-A. (2018). A stochastic multiple gradient descent algorithm. *European Journal of Operational Research*, 271(3):808–871.

Mezura-Montes, E. and Flores-Mendoza, J. (2008). Multiobjective problems solving from nature: From concepts to applications. *Nature-Inspired Algorithms for Optimization, Studies in Computational Intelligence Series, Ch. Improved Particle Swarm Optimization in Constrained Numerical Search Spaces*, pages 978–3.

Miller, B. L. and Goldberg, D. E. (1996). Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary computation*, 4(2):113–131.

Miller, B. L., Goldberg, D. E., et al. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex systems*, 9(3):193–212.

Mirjalili, S. and Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95:51–67.

Mourelatos, Z. P. and Zhou, J. (2005). Reliability estimation and design with insufficient data based on possibility theory. *AIAA Journal*, 43(8):1696–1705.

Mourelatos, Z. P. and Zhou, J. (2006). A design optimization method using evidence theory. *Journal of Mechanical Design*, 128(4):901–908.

Nabil, E. (2016). A modified flower pollination algorithm for global optimization. *Expert Systems with Applications*, 57:192–203.

Pal, S. K., Rai, C., and Singh, A. P. (2012). Comparative study of firefly algorithm and particle swarm optimization for noisy non-linear optimization problems. *International Journal of intelligent systems and applications*, 4(10):50.

Pan, H., Wang, L., and Liu, B. (2006). Particle swarm optimization for function optimization in noisy environment. *Applied Mathematics and Computation*, 181(2):908, 919.

Papadrakakis, M., Lagaros, N. D., and Plevris, V. (2005). Design optimization of steel structures considering uncertainties. *Engineering Structures*, 27(9):1408–1418.

Parr, J. M., Keane, A. J., Forrester, A. I. J., and Holden, C. M. E. (2012). Infill sampling criteria for surrogate-based optimization with constraint handling. *Engineering Optimization*, 44(10):1147–1166.

Parsopoulos, K. E. and Vrahatis, M. N. (2001). Particle swarm optimizer in noisy and continuously changing environments. In *M.H. Hamza (Ed.), Artificial Intelligence and Soft Computing*, pages 289–294.

Picheny, V. and Ginsbourger, D. (2014). Noisy kriging-based optimization methods: a unified implementation within the DiceOptim package. *Computational Statistics & Data Analysis*, 71:1035–1053.

Picheny, V., Ginsbourger, D., Richet, Y., and Caplin, G. (2013a). Quantile-based optimization of noisy computer experiments with tunable precision. *Technometrics*, 55(1):2–13.

Picheny, V., Wagner, T., and Ginsbourger, D. (2013b). A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization*, 48(3):607–626.

Pulido, G. T. and Coello, C. A. C. (2004). A constraint-handling mechanism for particle swarm optimization. In *2004 IEEE Congress on Evolutionary Computation (CEC), Portland, OR, USA*.

Rada-Vilela, J., Johnston, M., and Zhang, M. (2014). Population statistics for particle swarm optimization: Resampling methods in noisy optimization problems. *Swarm and Evolutionary Computation*, 17:37–59.

Rada-Vilela, J., Johnston, M., and Zhang, M. (2015). Population statistics for particle swarm optimization: Hybrid methods in noisy optimization problems. *Swarm and Evolutionary Computation*, 22:15–29.

Rakshit, P., Konar, A., and Das, S. (2017). Noisy evolutionary optimization algorithms–a comprehensive survey. *Swarm and Evolutionary Computation*, 33:18–45.

Rubinstein, R. (1999). The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190.

Rudolph, G. (1998). Evolutionary search for minimal elements in partially ordered finite sets. In *7th International Conference on Evolutionary Programming, San Diego, CA, USA*.

Runarsson, T. P. and Yao, X. (2000). Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on evolutionary computation*, 4(3):284–294.

Rustem, B. and Howe, M. (2009). *Algorithms for worst-case design and applications to risk management*. Princeton University Press.

Sahinidis, N. V. (2004). Optimization under uncertainty: state-of-the-art and opportunities. *Computers Chemical Engineering*, 28(6):971–983. FOCAPO 2003 Special issue.

Salehghaffari, S., Rais-Rohani, M., Marin, E., and Bammann, D. (2013). Optimization of structures under material parameter uncertainty using evidence theory. *Engineering Optimization*, 45(9):1027–1041.

Sasena, M. J., Papalambros, P. Y., and Goovaerts, P. (2001). The use of surrogate modeling algorithms to exploit disparities in function computation time within simulation-based optimization. In *Proceedings of the 4th Congress on Structural and Multidisciplinary Optimization*, Dalian, China, 2001.

Sasena, M. K. (2002). *Flexibility and Efficiency enhancements for constrained global design optimization with Kriging approximation*. PhD thesis, University of Michigan.

Schlüter, M. and Gerdts, M. (2010). The oracle penalty method. *Journal of Global Optimization*, 47(2):293–325.

Schonlau, M., Welch, W. J., and Jones, D. (1996). Global optimization with nonparametric function fitting. *Proceedings of the American Statistical Association's on physical and engineering sciences*, pages 183–186.

Schonlau, M., Welch, W. J., and Jones, D. R. (1998). Global versus local search in constrained optimization of computer models. *Lecture Notes-Monograph Series*, pages 11–25.

Schwefel, H.-P. and Rudolph, G. (1995). Contemporary evolution strategies. In *European conference on artificial life*, pages 891–907. Springer.

Shimizu, K. and Aiyoshi, E. (1980). Necessary conditions for min-max problems and algorithms by a relaxation procedure. *IEEE Transactions on Automatic Control*, 25(1):62–66.

Srivastava, R. K. and Deb, K. (2011). An EA-based approach to design optimization using evidence theory. In *13th annual conference on Genetic and evolutionary computation, Dublin, Ireland*.

Stagge, P. (1998). Averaging efficiently in the presence of noise. In Eiben, A., Back, T., Schoenauer, M., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature*, volume 1498 of *Lecture Notes in Computer Science*, pages 188–197. Springer Berlin Heidelberg.

Su, Y., Tang, H., Xue, S., and Li, D. (2016). Multi-objective differential evolution for truss design optimization with epistemic uncertainty. *Advances in Structural Engineering*, 19(9):1403–1419.

Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons.

Toropov, V. V., Schramm, U., Sahai, A., Jones, R. D., and Zeguer, T. (2005). Design optimization and stochastic analysis based on the moving least squares method. *6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil*.

Tsoukalas, A., Rustem, B., and Pistikopoulos, E. N. (2009). A global optimization algorithm for generalized semi-infinite, continuous minimax with coupled constraints and bi-level problems. *Journal of Global Optimization*, 44(2):235–250.

Ur Rehman, S. and Langelaar, M. (2015). Efficient global robust optimization of unconstrained problems affected by parametric uncertainties. *Structural and Multidisciplinary Optimization*, 52(2):319–336.

Vallejo, M. and Corne, D. W. (2016). Evolutionary algorithms under noise and uncertainty: a location-allocation case study. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece*.

Yao, W., Chen, X., Luo, W., van Tooren, M., and Guo, J. (2011). Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles. *Progress in Aerospace Sciences*, 47(6):450–479.

Yin, H., Yu, D., Yin, S., and Xia, B. (2018). Possibility-based robust design optimization for the structural-acoustic system with fuzzy parameters. *Mechanical Systems and Signal Processing*, 102:329–345.

Zhang, J., Xiao, M., Gao, L., Qiu, H., and Yang, Z. (2018). An improved two-stage framework of evidence-based design optimization. *Structural and Multidisciplinary Optimization*, pages 1–21.

Zhou, A. and Zhang, Q. (2010). A surrogate-assisted evolutionary algorithm for minimax optimization. In *2010 IEEE Congress on Evolutionary Computation (CEC), Barcelona, Spain*.

Zhou, J. and Mourelatos, Z. P. (2006). A sequential algorithm for possibility-based design optimization. In *ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Charlotte, NC, USA*.