Evolution strategies for robust optimization

Kruisselbrink, J.W.

Chapter 7

Finding Robust Optima

This chapter focuses on the scenario depicted in Figure 7.1. In this scenario we have to deal with the fact that in the real-world system, the design variables cannot be set arbitrarily precise, yet there is a (simulation) model in which the solutions can be evaluated arbitrarily precise and that replaces the real-world system. Hence, the aim is to find optima that are actually useful in practice even when the real-world realizations of these solutions differ from their theoretically optimal or desired settings due to uncertainties and noise.

The intent of this chapter is to answer the following questions: 1) How can the aim of finding robust optima be modeled in the optimization problem statement? 2) How does the aim to find robust optima affect the difficulty of an optimization problem? 3) How should Evolution Strategies be adapted in order to find robust optima?

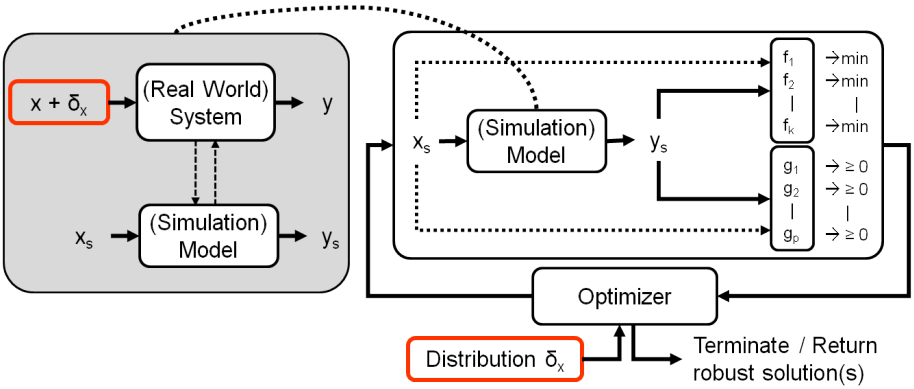

Figure 7.1: A typical robust optimization scenario: there is a real-world system in which the solutions cannot be realized arbitrarily precise. For the optimization, a (simulation) model (for which the solutions can be evaluated arbitrarily precise) replaces the real-world system; the aim is to find optima that are robust with respect to the anticipated input disturbances.

This chapter starts in Section 7.1 with providing an overview of the main concepts of the problem of finding robust optima for general real-parameter optimization problems. Section 7.2 focuses on the application of Evolution Strategies to a more specific class of optimization problems, namely on unconstrained single objective optimization problems. Here, we will restrict ourselves to a particular type of noise and a particular type of robustness measure. For this, we summarize and compare several techniques that have been proposed in the literature. Section 7.3 closes with a summary and discussion.

7.1 Problem Definitions for Finding Robust Optima

For the scenario depicted in Figure 7.1, the objective and constraint functions are of the same form as a normal real-parameter optimization problem, however, instead of aiming to find optima for the normal objective and constraint functions, the aim is to find solutions that are optimal for the problem formulation

$$f_i(\mathbf{x} + \boldsymbol{\delta}_\mathbf{x}) \rightarrow \min, i = 1, \ldots, k, \tag{7.1}$$

$$g_j(\mathbf{x} + \boldsymbol{\delta}_\mathbf{x}) \geq 0, j = 1, \ldots, p. \tag{7.2}$$

Here, $\boldsymbol{\delta}_\mathbf{x}$ denotes the uncertainty or possible variability of the input variables, of which the distribution $\mathrm{pdf}_{\boldsymbol{\delta}_\mathbf{x}}$ is assumed to be known. Hence, for each candidate solution, the objective and constraint functions become uncertain variables of which the possible variation depends on the quality of neighboring solutions that can be possible perturbations of that candidate solution. Or, in other words, the quality of a candidate solutions $\mathbf{x}$ is based on the η-*neighborhood*, defined as:

Definition 7.1.1 (η-neighborhood)**:** For a real-parameter optimization problem $(\mathcal{X}, \mathcal{F}, \mathcal{G})$ for which disturbances of the design variables and environmental parameters are anticipated, the neighborhood $\eta_\mathbf{x}$ is the set of possible disturbances of $\mathbf{x}$, i.e.,

$$\eta_\mathbf{x} = \{\mathbf{x}' \mid \mathbf{z} = \mathbf{x} - \mathbf{x}' \text{ and } \mathrm{pdf}_{\boldsymbol{\delta}_\mathbf{x}}(\mathbf{z}) > 0\}. \tag{7.3}$$

Similar to when having noisy objective functions, also for uncertainty and/or noise in the design variables we can distinguish between stationary and non-stationary noise/uncertainty. That is, in a stationary noise case, $\boldsymbol{\delta}_\mathbf{x}$ is independent of $\mathbf{x}$ and hence can simply be written as $\boldsymbol{\delta}$. Otherwise, $\boldsymbol{\delta}_\mathbf{x}$ is said to be non-stationary. In this work we will restrict ourselves to stationary uncertainty/noise and henceforth we will use the notation $\boldsymbol{\delta}$ for denoting the input perturbations.

Although the idea behind the desire to find robust optima is clear, there are different (possibly conflicting) ways to define a quality measure or *robustness measure*. That is, to define a measure that incorporates both quality and robustness. Choosing a robustness measure is a design choice that should be made when modeling a given problem as an optimization problem. For every objective and every constraint, one should consciously ask in which respect

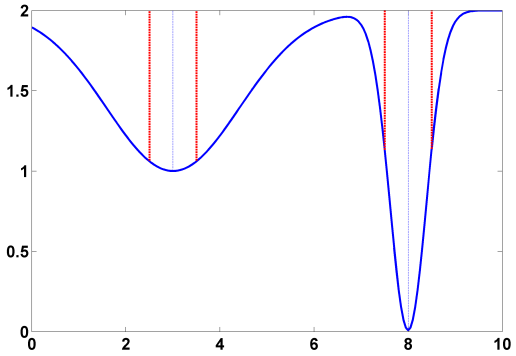

Figure 7.2: Illustration of the trade-off between quality and stability: with variation $\delta \sim \mathcal{U}(-\sigma_\epsilon, \sigma_\epsilon)$ in design variable x, optimizer $x_1 = 3$ is the more stable, but optimizer $x_2 = 8$ is still qualitatively better, given the variation bandwidth $\sigma_\epsilon = 0.5$.

solutions are desired to be robust. Moreover, the aim for robustness may be different for each objective and each constraint. Especially between objectives and constraints there may be different aims for robustness. In the context of robust design, Park [Par07] notes that there is a distinction between *reliability design* and *robust design*. In robust design the emphasis lies on low sensitivity with respect to the objective function(s), while in reliability design the emphasis lies on assuring that the constraints are not violated.

7.1.1 Robustness Measures for Objective Functions

For objective functions, the notion of robustness refers to the quality of a solution with respect to variations caused by uncertainties and noise. As noted by Jin and Sendhoff [JS03], the aim for robustness can be approached from two points of view that are often conflicting: robustness and performance (or: quality and stability):

- **Performance / quality**: Robustness of a solution is measured from the perspective of **overall performance** under the variation of the uncertain parameters of the solution.
- **Robustness / stability**: The robustness of a solution is measured from the perspective of **minimal performance variation** under the variation of the uncertain parameters of the solution.

Figure 7.2 illustrates the trade-off between quality and stability. It shows the fitness landscape of a one-dimensional, single objective, real-parameter optimization problem with design variable x that has a uniform uncertainty variation $\delta \sim \mathcal{U}(-\sigma_\epsilon, \sigma_\epsilon)$, with $\sigma_\epsilon = 0.5$. One could choose here for the most stable optimum at $x_1 = 3$ which has almost constant performance with respect to variations of x, or one could choose for the qualitatively best solution $x_2 = 8$, which has a better overall objective function value than x_1, but is less stable.

Although the choice for quality or stability generally provides an indication of what is meant by robust quality, there are still varying ways in which these aims can be translated into a robust quality measure. That is, a formulation of *effective* or *robust* objective functions is required to capture the aim for robustness in a concrete measure. Commonly used measures for robust quality are:

- **Expected solution quality**: The expected performance under variation of the uncertain design variables. This measure is considered in, amongst others, [TG97, WHB98, Tsu99, Bra98, Bra01, TG03, BS06a, ONL06, PBJ06]. For an objective function $f(\mathbf{x})$, this measure is described as an *expected objective function* as:

$$f_{\text{eff}}(\mathbf{x}) = f_{\text{exp}}(\mathbf{x}) = \mathbf{E}\left[f(\mathbf{x}+\boldsymbol{\delta}) \,|\, \mathbf{x}\right] = \int_{\mathbf{z}\in\mathbb{R}^n} f(\mathbf{x}+\mathbf{z})\ \text{pdf}_{\boldsymbol{\delta}_\mathbf{x}}(\mathbf{z})\ d\mathbf{z}. \tag{7.4}$$

  Obviously, this measure requires that the uncertainties in the design variables $\boldsymbol{\delta}$ are of stochastic nature, or at least can be modeled as such, with a probability density function $\text{pdf}_{\boldsymbol{\delta}_\mathbf{x}}(\mathbf{z})$.

- **Worst-case solution quality**: The worst-case quality with respect to the alternatives under variation of the uncertain variables (considered in, e.g., [LOL05, ONL06]). For an objective function $f(\mathbf{x})$ that is to be minimized, the robust objective function is determined as the maximum function value in the η-neighborhood of each candidate solution. That is,

$$f_{\text{eff}}(\mathbf{x}) = f_{\text{wc}}(\mathbf{x}) = \sup_{\mathbf{x}'\in\eta_\mathbf{x}} f(\mathbf{x}'). \tag{7.5}$$

  Note that for this measure, the η-neighborhood should be bounded and candidate solutions should have distinct η-neighborhoods in order to obtain distinct objective function values.

- **Threshold acceptance probability**: The maximal probability that a perturbation of a solution satisfies a certain threshold (considered in, e.g., [BS06b]). Or, in other words, maximizing the probability that a perturbation of a solution is part of the L_q level set. Given a threshold value q that indicates the acceptable solution quality, the optimization goal is to maximize the conditional probability of the objective functions satisfying this threshold:

$$f_{\text{eff}}(\mathbf{x}) = f_{\text{sat}}(\mathbf{x}) = \text{P}[f(\mathbf{x}+\boldsymbol{\delta}) \leq q \,|\, \mathbf{x}] \rightarrow \max. \tag{7.6}$$

Likewise, commonly used measures for stability are:

- **Performance variance:** Minimizing the performance variance under variation of the uncertain design variables. In this case, the original objective function $f(\mathbf{x})$ can be

remodeled into a robust objective function $f_{\text{eff}}(\mathbf{x})$ as to minimize the conditional variance:

$$f_{\text{eff}}(\mathbf{x}) = f_{\text{var}}(\mathbf{x}) = \text{Var}[f(\mathbf{x} + \boldsymbol{\delta}) \,|\, \mathbf{x}] \rightarrow \min . \tag{7.7}$$

Similar to the expected objective function Eq. 7.4, this measure requires that the uncertainties in the design variables $\boldsymbol{\delta}$ are of stochastic nature or at least can be modeled as such. Moreover, this measure should be accompanied by an additional objective that also takes solution quality into account. In [JL02] and [JS03] methods are proposed that implement this measure in a multi-objective setting by combining it with the optimization of the expected objective function.

- **Sensitivity region / trust region:** Maximization with respect to the region around the candidate solutions in which the deviation in performance is still acceptable. This measure can be used when the solutions need to be as stable as possible and is also frequently accompanied by an additional objective that aims for the optimization of the initial objective function. Examples of approaches that adopt this aim can be found in [BA06] and [LAA05].

  Mathematically, in a more general sense, a robust objective function $f_{\text{eff}}(\mathbf{x})$ implementing this aim can be constructed as:

$$f_{\text{eff}}(\mathbf{x}) = \sup\{r \in \mathbb{R}_+ \,|\, B_r(\mathbf{x}) \subseteq L(\mathbf{x})\} \rightarrow \max, \tag{7.8}$$

  with

$$L(\mathbf{x}) = \{\mathbf{x}' \,|\, f(\mathbf{x}') \in [f(\mathbf{x}) - q, f(\mathbf{x}) + q]\}, \tag{7.9}$$

$$B_r(x) = \{\mathbf{x}' \in \mathbb{R}^n \,|\, d(\mathbf{x}, \mathbf{x}') < r\}. \tag{7.10}$$

  Here, $L(\mathbf{x})$ denotes the set of all points which are within the region of tolerance, and $B_r(\mathbf{x})$ denotes the set of points within radius r from the point $\mathbf{x}$ (hence, it aims to maximize a sphere-like region). The distance measure $d(\mathbf{x}, \mathbf{x}')$ should be proportional or related to the probability of obtaining $\mathbf{x}'$ from $\mathbf{x}$. For example, in Euclidean spaces, the Euclidean distance is a suitable distance measure for uncorrelated Gaussian noise, and for uncorrelated uniform noise, the Tchebychev distance is a suitable distance measure.

7.1.2 Robustness of Constraint Satisfaction

For constraints, the aim for robustness is different. Ideally constraints are always satisfied, however, this is not always possible. As noted by Samsatli et al. [SPS98] a distinction can be made a between *hard constraints* and *soft constraints*.

Hard constraints are the constraints that may not be violated under any circumstances, and should therefore hold under for all possible variations in $\eta_{\mathbf{x}}$, i.e.,

$$g_{\text{eff}}(\mathbf{x}) = \inf_{\mathbf{x}' \in \eta_{\mathbf{x}}} g(\mathbf{x}') \geq 0. \tag{7.11}$$

Also here, the η-neighborhood should be bounded in order to be of practical use.

Soft constraints are less strict and may occasionally be violated (as long as the probability of violation is small). These types of constraints can be accounted for in different ways:

- **Probabilistic constraints**: One possibility is to redefine a constraint function $g(\mathbf{x})$ in a probabilistic form, where the probability of satisfying the constraint should be above a certain threshold c, i.e.,

  $$g_{\text{eff}}(\mathbf{x}) = P\left[g(\mathbf{x} + \boldsymbol{\delta}) \geq 0 \,|\, \mathbf{x}\right] - c \geq 0. \tag{7.12}$$

  With $0 < c \leq 1$. Note that equation (7.11) can be defined in terms of equation (7.12) by setting $c = 1$.

- **Virtual bounds for uncertainty variations**: Soft constraints also arise in cases where the variation due to uncertainty is theoretically unbounded (e.g., Gaussian variation). In these cases it is theoretically not possible to require the satisfaction of each strict constraint, unless the constraint is independent of the particular uncertain variable. A solution to deal with theoretically unbounded variation in the design variables (besides using probabilistic constraints) is to assign virtual bounds to the uncertainty variations between which the uncertainty variations are practically always expected to be. For example: common practice for Gaussian uncertain variables is to assume $-6\sigma < \boldsymbol{\delta} < 6\sigma$ for every uncertain variable $\boldsymbol{\delta}$. This is also the general principle behind design for "six sigma" [KYG04]. The way in which this is modeled for a given constraint $g(\mathbf{x})$ is the following:

  $$g_{\text{eff}}(\mathbf{x}) = \inf_{\mathbf{x}' \in \eta'_{\mathbf{x}} \subset \eta_{\mathbf{x}}} g(\mathbf{x}') \geq 0, \tag{7.13}$$

  with $\eta'_{\mathbf{x}}$ being a proper bounded subset of the η-neighborhood of $\mathbf{x}$.

- **Transformation to objective functions**: Remodeling or transforming soft constraints as additional objective functions. Instead of requiring that a constraint is satisfied, one could also grade the degree by which a constraint is satisfied and use that as an additional objective function. For example, maximization of the satisfaction probability:

  $$f_{\text{eff}}(\mathbf{x}) = P\left[g(\mathbf{x} + \boldsymbol{\delta}) \leq 0 \,|\, \mathbf{x}\right] \rightarrow \max. \tag{7.14}$$

7.1.3 Multi-Objective Robustness Measures

When dealing with the trade-off between stability and quality, the robust optimization of one objective can also seen as a multi-objective task (see, e.g., [JS03]). Hence, individual objective

functions can be split up into multiple robust objective functions. By doing so, particularly the trade-off between stability and quality can be controlled. For constraints a translation of one constraint to multiple robust constraints is not reasonable. However, as shown in the previous section, certain soft constraints can be transformed into additional objective functions.

7.1.4 Robustness Transformations

Given the various different measures for robustness, an obvious next question is: does it really matter which robustness measure is used? Or, is it even worthwhile at all to use a robustness measure instead of performing an optimization on the objective functions in their normal form? Straightforwardly, the answer to these questions is: sometimes it matters and sometimes not; this depends on the optimization problem at hand. However, in order to give an example of how drastic the impact of choosing different robustness measures could be, we consider the following one-dimensional function and suppose that we want to minimize it over x in the interval $[0, 10]$:

$$\begin{aligned} f(x) &= 1 + f_1(x) + f_2(x), \\ f_1(x) &= \begin{cases} \left(\frac{x-4}{5}\right)^2 & , x < 4 \\ 1 & , \text{otherwise} \end{cases}, \\ f_2(x) &= -1.8 \cdot \exp\left(-\frac{(x-5)^2}{0.2}\right) - 2 \cdot \exp\left(-\frac{(x-7)^2}{0.1}\right). \end{aligned} \tag{7.15}$$

For this function, consider an uncertainty of $\delta \sim U(-\sigma_\epsilon, \sigma_\epsilon)$, $\sigma_\epsilon = 0.5$ of the design variable x and consider the following measures for robustness:

- optimize the expected objective function $f_{\text{exp}}(x)$,
- optimize the worst-case objective function $f_{\text{wc}}(x)$,
- optimize on the normal objective function $f(x)$.

Figure 7.3 shows the fitness landscapes of these three different measures. In this example both robust objective functions are very different from the original objective function. Moreover, the optimal solutions for all three measures lie in different parts of the search space. Hence, from this example it can be seen that considering uncertainty can change a problem very much. Although this does not have to be the case for every optimization problem, one should be aware of this, and realize that performing optimization without accounting for the uncertainty/noise in the input variables and/or environmental parameters might yield results of which the realizations are of very poor quality. For constraints, a similar message holds. That is, in some cases optimal solutions might be prone to feasibility failure when considering uncertainty in the design variables in which case it matters, yet in other cases it might not matter at all.

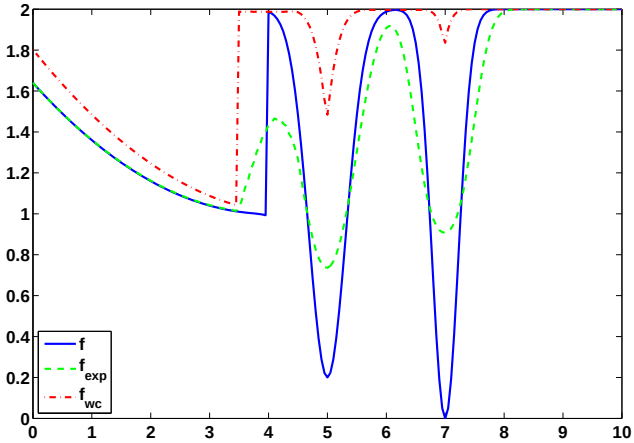

Figure 7.3: A one-dimensional objective function in which different measures of robustness produce different fitness landscapes, given uncertainty/noise in the design variables.

Besides the differences that occur due to the use of different robustness measures, another important factor is the magnitude of the anticipated variation. Small perturbations, for example, yield only small differences between the original objective function landscape and the effective objective function landscape, while large perturbations of the uncertainty/noise level may cause severe differences. Figure 7.4 illustrates how the original objective function landscape of the one-dimensional function Eq. 7.16 with input disturbances $\delta \sim \mathcal{U}(-\sigma_\epsilon, \sigma_\epsilon)$ changes for increasing values of σ_ϵ, considering the expected and worst-case objective function.

Mathematically, one can see the robustness measures described in the previous section as integral transformations (or filters) of the original objective functions and constraints. That is, a robustness measure is a transformation $T_{\text{eff}}(\mathbf{f}(\mathbf{x}), \mathbf{g}(\mathbf{x}))$ that maps the original objective and constraint functions $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ to new objective and constraint functions $\mathbf{f}_{\text{eff}}(\mathbf{x})$ and $\mathbf{g}_{\text{eff}}(\mathbf{x})$.

To view the problem of finding robust optima in terms of robustness transformations of the objective function landscape allows us to get some insight into the difference of normal optimization and the goal of finding robust optima. For example, consider the expected objective function transformation $f_{\text{exp}}(\mathbf{x})$ with Gaussian noise in the design variables. One can observe that this transformation is an integral transform with the original objective function convoluted with the Gaussian probability density function. More specific, the robustness transformation acts as a Gaussian filter or generalized Weierstrass transform (see, e.g., [Zay96]) from which we know that:

1. the effective objective function landscape should be smoother, hence easier for optimization since the spatial correlation is increased due to this smoothing operation,

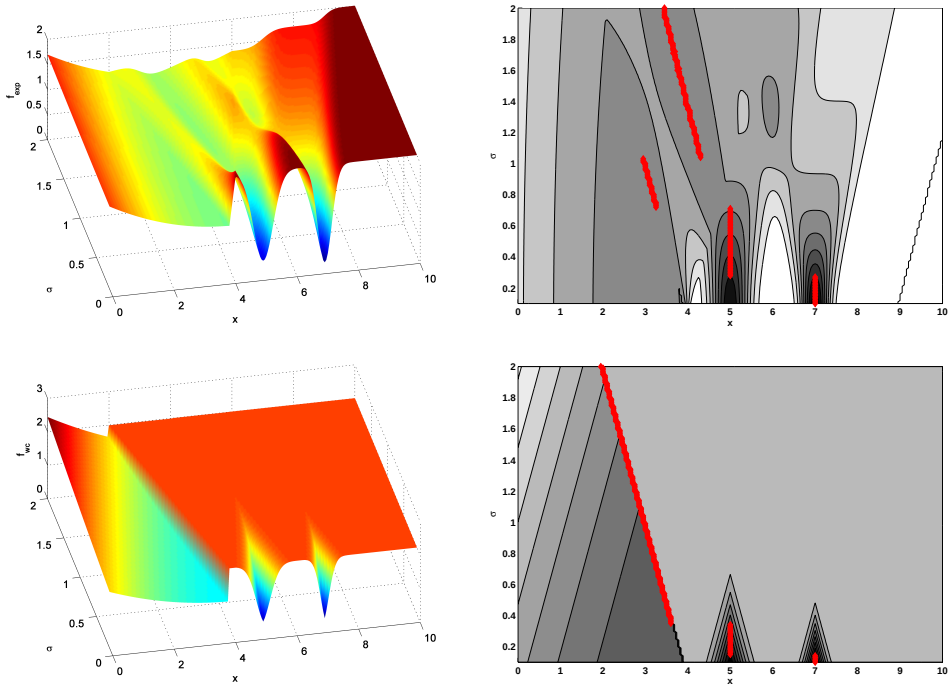

Figure 7.4: Illustration of the change of the effective objective function landscapes when varying the magnitude of the anticipated disturbances ($\sigma_\epsilon \in [0, 2]$) in the design variables. Top: the landscape dynamics for the expected solution quality. Bottom: the landscape dynamics for the worst-case solution quality. Left column: a surface plot of the landscape changing for different magnitudes of the input disturbances. Right column: contour plots showing also the locations of the optima for different uncertainty/noise magnitudes.

2. the extremal values (both minima and maxima) will lie between the extrema of the original objective function,

3. if the original objective function is an integrable function, then so is the effective objective function.

For other standard noise distributions besides Gaussian (e.g., uniform or Cauchy), these properties also hold. However, for different transformations than the expected objective function transformation, other properties might hold (e.g., the worst-case robustness measure).

Paenke et al. [PBJ06] classified the differences that could occur between the original objective function landscape and the effective objective function landscape in four categories:

- **Identical Optimum**: The original optimum and robust optimum are identical.
- **Neighborhood Optimum**: The original optimum and the robust optimum are located on the same hill (with respect to the original objective function landscape).
- **Local-Global-Flip**: An originally local optimum becomes the robust optimum.
- **Max-Min-Flip**: The robust optimum (for minimization) is located at a local maximum of the original objective function.

Figure 7.5 provides an example for each category of this classification.
As an alternative to this classification, we propose to classify robust optimizers by means of their explicit characteristics. This alternative classification is based on whether a robust optimizer is shifted with respect to some original global optimizer or whether it emerges anew:

Definition 7.1.2 (Shifted Robust Optimizer and Emergent Robust Optimizer)**:** Let σ denote a scaling of the magnitude of the input disturbance $\boldsymbol{\delta}$ and let $\mathbf{x}^*_\sigma$ denote a robust optimizer of $f_{\text{eff}\,\sigma}(\mathbf{x})$. That is, $f_{\text{eff}\,\sigma}(\mathbf{x})$ denotes the effective objective function for input perturbations $\boldsymbol{\delta}' = \sigma\boldsymbol{\delta}$. A robust optimizer $\mathbf{x}^*_{\sigma=1} = \mathbf{x}^*_{\text{eff}}$ is called a *shifted robust optimizer* if there exists a path $p : [0,1] \to X$ such that for all $t \in [0,1] : p(t) \in \text{argmin}_{\mathbf{x}\in X} f_{\text{eff}\,\sigma=t}(\mathbf{x})$, $p(0) = \mathbf{x}^*_{\sigma=0}$, and $p(1) = \mathbf{x}^*_{\sigma=1}$, where $\mathbf{x}^*_{\sigma=0} = \mathbf{x}^*$ is some optimizer for the original objective function $f(\mathbf{x})$. Otherwise, it is an *emergent robust optimizer*.

That is, if by gradually increasing the magnitude of the input disturbances the robust optimizer follows a connected path, it shifts. Otherwise, if a small increment of the noise magnitude leads to a completely different robust optimizer, it emerges anew, hence it is called emergent. The advantage of this classification is that it relates more closely to two major issues of finding robust optima: 1) accurately targeting robust optima (in case of shifts), and 2) targeting the right attractor regions in which a/the emergent robust optimizer is located (emergent robust optimizers). The first challenge relates to algorithms which are good in exploitation, the second challenge requires explorative strength.

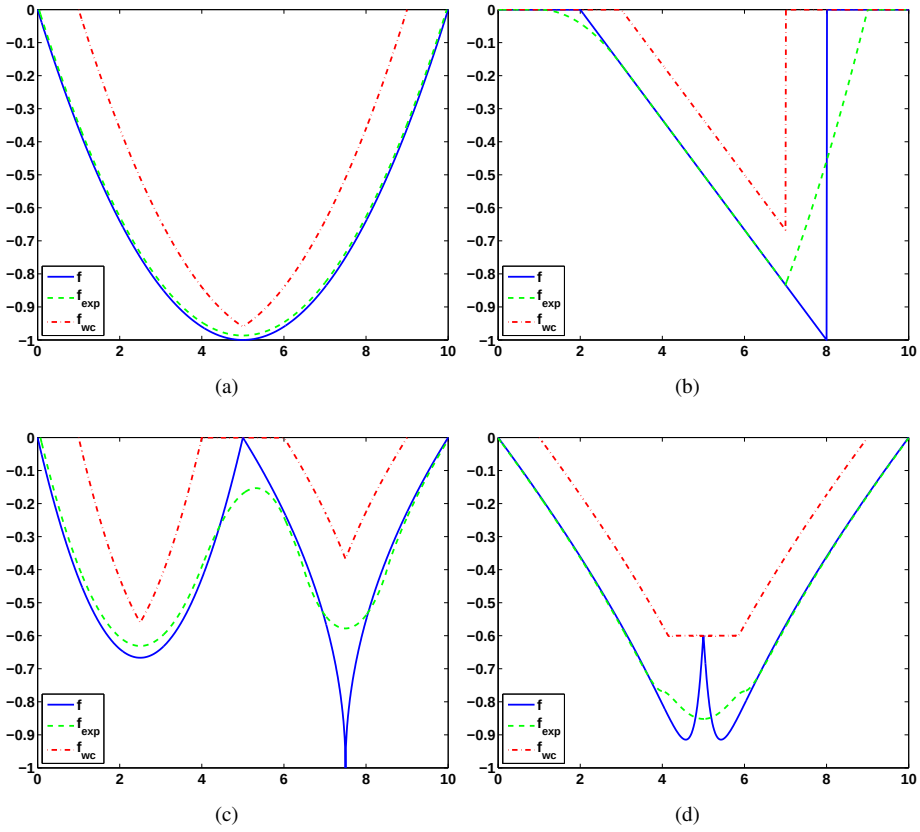

Figure 7.5: Illustrations of the four categories of differences in objective function landscape of the original objective function and the robust objective function according to Paenke et al. [PBJ06]: (a) identical optimum, (b) neighborhood optimum, (c) local-global-flip, (d) max-min-flip.

When we consider once again the function of Eq. 7.16 and look at the plots of Figure 7.4, we see in the right column that there are certain catastrophic events in which the optimum location "jumps" from one location to another. This happens three times for the expected objective function, and two times for the worst-case objective function. Besides these jumps, one can also identify shifts, for instance for the worst-case objective function, when $\sigma_\epsilon \gtrsim 0.4$. Moreover, besides these shifts and jumps, Sendhoff et al. [SBO04] show that input disturbances can even induce multimodality, meaning that an original optimizer might "split up" into multiple robust optimizers when increasing the noise strength. Lastly, it should be noted that two "robust optimizer branches" might also join when increasing the disturbance magnitude.

7.1.5 Computing the Robustness of Optima

The transformation of objective and constraint functions to functions that incorporate the notion of robustness is is the essence of the problem of finding robust optima. From this, we can identify two main issues that govern the problem of finding robust optima:

1. Choosing the appropriate robustness measure for the problem at hand.
2. Computing or approximating the robustness measure for candidate solutions.

The former is a part of the problem modeling, based on the preference of the user/expert and optimization problem context. The latter is a mathematical or algorithmic problem, which is challenging because exact computation of the solution quality for the robustness measures is often not possible.

In many cases it is either difficult or impossible to obtain exact derivations of the effective objective and constraint functions. Hence, alternative evaluation methods are needed that approximate the function values of the robustness measures, for instance, by sampling. Adopting the notation used in this work, we use $\hat{f}_{\text{eff}}(\mathbf{x})$ to denote an approximation function for the effective objective function $f_{\text{eff}}(\mathbf{x})$ and $\hat{g}_{\text{eff}}(\mathbf{x})$ to denote the approximation function for the effective constraint functions $g_{\text{eff}}(\mathbf{x})$.

There are two types of approaches to determine the robust quality of a solution: analytical approaches and statistical approaches. Analytical approaches determine the robustness of a solution by means of analytical techniques, such as using (approximations of) first-order and second-order derivatives in order to obtain an accurate estimate of the robust quality. Statistical methods determine the robustness of a solution by means of statistical approximation. An obvious example is to approximate the expected objective function by means of Monte-Carlo integration. In any case, it is clear that in order to determine the robust objective function value for a given solution $\mathbf{x}$, information is needed about the quality of the solutions in the neighborhood of $\mathbf{x}$. The challenge when searching for robust optima is to obtain good approximations of the robust quality of candidate solutions as efficiently as possible in order to perform optimization on the transformed objective function landscape.

7.2 Strategies for Finding Robust Optima

From here on, we will focus on unconstrained single-objective optimization problems and consider as robustness measure the expected fitness in the light of a known uncorrelated multivariate probability density function of the disturbances, which is either completely uniform or completely Gaussian. In this section, we will summarize approaches that can be used for Evolution Strategies to solve such problems. These approaches are categorized as:

- Myopic approaches (Section 7.2.1)
- Single- and multi-evaluation approaches (Section 7.2.2, Section 7.2.3, and Section 7.2.4)
- Adaptive averaging approaches (Section 7.2.5 and Section 7.2.6)
- Archive based approaches (Section 7.2.7)
- Metamodeling approaches (Section 7.2.8)
- Niching approaches (Section 7.2.9)
- Overlap exploiting approaches (Section 7.2.10)

7.2.1 The Myopic Approach for Finding Robust Optima

The simplest approach for finding robust optima is doing nothing extra, but to rely on the inherent attraction of Evolutionary Algorithms for finding robust optima. Branke [Bra98] states: "The standard EA by nature favors hills over peaks, since usually the probability that some individuals of the initial population are somewhere in the basin of attraction of a hill is higher, also the average fitness of individual in the hill area is higher.". Combining this observation with the fact that for some problems, the original optimizer is also the robust optimizer (or at least is located close to the robust optimizer), it is reasonable to consider normal Evolutionary Algorithms as good alternatives for finding robust optima as well. In this work, we will call this approach the *myopic approach*.

Experiment 7.2.1 (The inherent attraction of robust peaks)**:** For testing the inherent attraction of robust peaks, we perform 1000 runs of a $(5/2_{DI}, 35)$-σSA-ES (see Section 4.2.2) and a CMA-ES (see Section 4.2.3) on a 10-dimensional instance of Branke's multipeak problem (see Appendix B.6). This problem consists of $2^n = 1024$ peaks varying in robustness. After each run, the peak score of the final solution is recorded (see Technical Note 7.1). For each run, an evaluation budget of $10,000$ function evaluations is considered.

Figure 7.7 shows histograms of the peak score of the final solution of 1000 optimization runs of the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES on a 10-dimensional instance of Branke's multipeak problem. From these plots, the inherent attraction of robust peaks is very apparent. The $(5/2_{DI}, 35)$-σSA-ES converges to the robust peak in more than 65% of the runs and

Technical Note 7.1: The Peak Score of Branke's Multipeak Problem

Branke's multipeak problem is described in detail in Appendix B.6 and visualized in Figure 7.6. In the uncertainty free case, the optima are located at $\{-1, 1\}^n$, where the quality of the local optima is ranked by the number of positive elements of the optimizer, i.e., the global optimum is located at $[1]^n$, the "worst" local optimum is located at $[-1]^n$, and peaks with the same number of positive and negative elements are equivalent (hence, there are n classes of equivalent peaks).
When considering the expected objective function for an input uncertainty $\boldsymbol{\delta} \sim \mathcal{U}(-\mathbf{1/2}, \mathbf{1/2})$, the order in peak quality is reversed. Hence, the most robust peak is located at $[-1]^n$, the least robust peak is located at $[1]^n$. We can construct a peak score ps that identifies the type of peak in which a solution $\mathbf{x}$ is located as

$$\mathrm{ps}(\mathbf{x}) = \sum_{i=1}^{n} \mathrm{signum}(x_i). \tag{7.16}$$

Here, $\mathrm{ps} = 0$ identifies the most robust peak and $\mathrm{ps} = n$ identifies the least robust, but highest peak. Using this peak score, we can analyze the attraction of the different peaks of myopic approaches and therefore the inherent attraction of robust peaks.

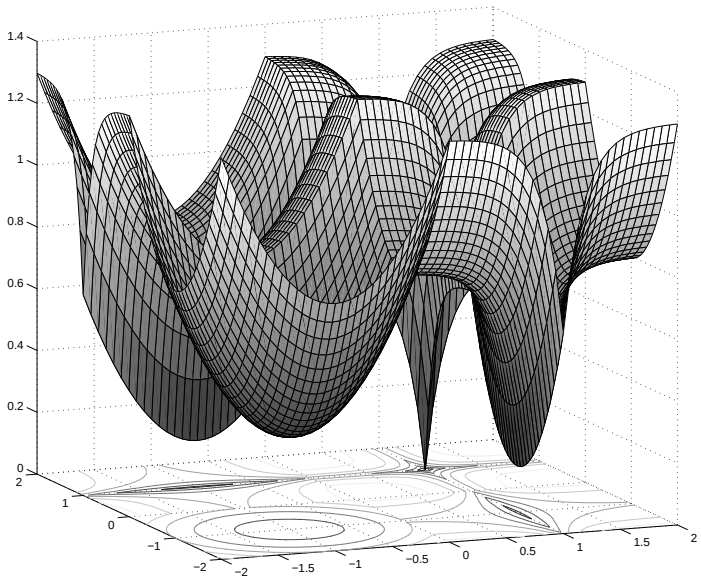

Figure 7.6: A plot of the objective function landscape of a two-dimensional instance of Branke's multipeak problem. This problem consists of 2^n peaks, varying in robustness.

shows a clear preference for the robust peak. The CMA-ES hits the robust peak in more than 30% of the runs and also shows a clear preference for the robust peak, but less clear than the $(5/2_{DI}, 35)$-σSA-ES.

In this small example, it can be seen clearly that the myopic approaches are biased towards

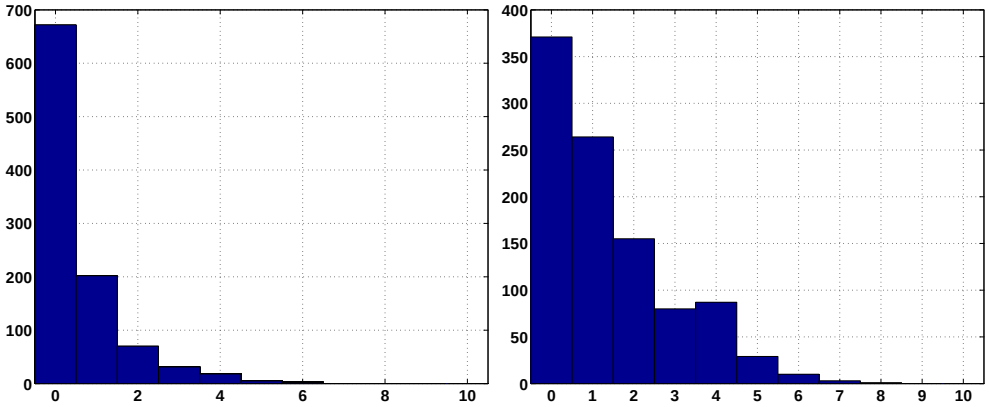

Figure 7.7: Histograms of the peak score of the final solution of 1000 optimization runs of the $(5/2_{DI}, 35)$-σSA-ES (left) and the CMA-ES (right) on a 10-dimensional instance of Branke's multipeak problem.

converging to the more robust peaks, making these approaches for the problem of identifying the more robust peaks already very promising. As an intuitive explanation of the inherent attraction of the more robust peaks as opposed to the sharper peaks, two factors play an important role: 1) the basin of attraction, and 2) the probability that a random point in the robust peak is better than a random point in the sharper peak. That is, the basin of attraction determines the probability of actually generating search points in a certain peak and the probability of a random solution in the robust peak being better than a random solution in the non-robust peak determines the possibility of the former solution to survive. For robust peaks, the latter probability can be assumed to be relatively high (i.e., the solutions in the neighborhood should be of relatively good quality, otherwise the peak is not robust). Hence, for Evolution Strategies, individuals located in these peaks have a high probability to survive and the population will be attracted to these peaks.

The tendency of myopic implementations to converge to the more robust peaks can be seen as undesirable from the perspective of classical optimization, but is advantageous when searching for robust optima. For this, myopic approaches could be serious alternatives for the problem of finding robust optima.

Experiment 7.2.2 (The robust precision of the myopic approach)**:** We perform 50 runs of a $(5/2_{DI}, 35)$-σSA-ES (see Section 4.2.2) and a CMA-ES (see Section 4.2.3) on a 10-dimensional instance of the Heaviside sphere problem (see Appendix B.2). The Heaviside sphere problem is a problem in which the robust optimizer is a shifted version of the uncertainty free optimizer. A two-dimensional instance of the Heaviside sphere problem is visualized in Figure 7.8. Each run has a budget of $10,000$ function evaluations.

Figure 7.9 shows the results of Experiment 7.2.2 in terms of the performance, measured in

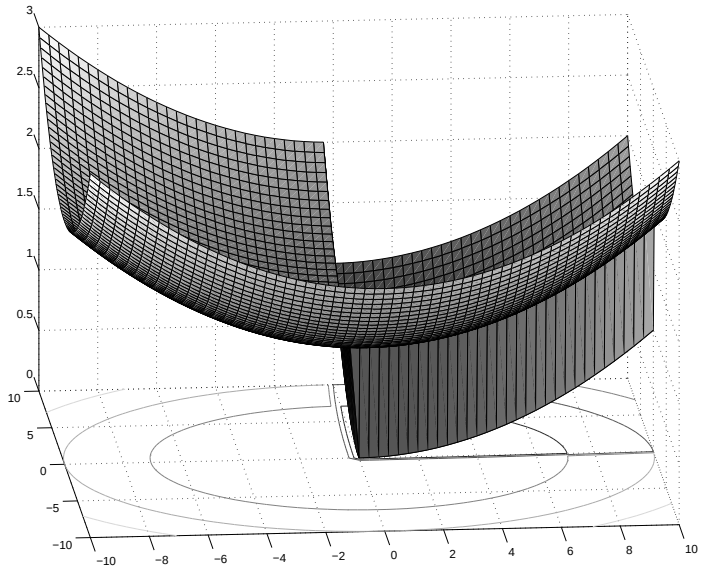

Figure 7.8: A plot of the objective function landscape of a two-dimensional instance of the Heaviside sphere problem. This problem consists one optimizer at the edge of the ridge. The robust optimizer will lie at a distance from the ridge.

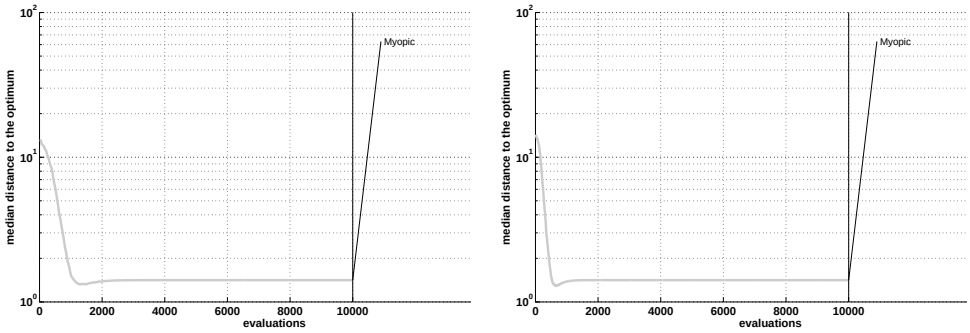

Figure 7.9: Results of Experiment 7.2.2. The performance of running canonical instances of the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES on a 10-dimensional instance of the Heaviside sphere problem. The performance is measured in terms of median distance to the optimum. Left column: the $(5/2_{DI}, 35)$-σSA-ES. Right column: the CMA-ES. The results are obtained using 50 runs for each scheme.

terms of median distance to the optimum. For this unimodal test problem with a shifted robust optimizer, it is apparent that myopic approaches will not always target the robust optimum. In this case, both algorithms target the uncertainty free optimum and not the robust optimum and find solutions that are sub-optimal with respect to the effective objective function. Moreover, in addition, even if myopic approaches get close to a robust optimizer, they are not able to detect this and can easily deviate from them again. In Figure 7.9 this "overshooting" behavior can be observed for both the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES, where the stagnation distance is higher than the closest distance, reached at approximately 1000 and 500 evaluations respectively.

From the two small experiments performed in this section, we observe that the robust regions of the search space have an inherent attraction on canonical Evolution Strategies. That is, blunt peaks are preferred over sharp peaks of a similar height. For the $(5/2_{DI}, 35)$-σSA-ES it seems that this attraction is more pronounced than for the CMA-ES. However, when the robust optimizer is a shifted version of the uncertainty free optimizer, myopic approaches will naturally fail to zoom in on the robust optimizer with arbitrary precision. The challenge of finding robust optima is therefore to adapt these such that they can accurately target (or zoom in on) robust optima, and improve their already present bias towards the robust peaks.

7.2.2 Single- and Multi-Evaluation Methods

A straightforward way to approximate the expected solution quality is to use Monte-Carlo integration. That is, approximating $f_{\exp}(\mathbf{x})$ for a candidate solution $\mathbf{x}$ as

$$\hat{f}_{\exp}(\mathbf{x}) = \sum_{i=1}^{m} f(\mathbf{x} + \mathbf{z}_i) \ , \ \mathbf{z}_i \sim \boldsymbol{\delta}, \tag{7.17}$$

with $\mathbf{z}_1, \ldots, \mathbf{z}_m$ being m disturbances sampled from the distribution of the anticipated input noise. This approach was proposed originally by Tsutsui et al. [TGF96] for $m = 1$ and shortly after that generalized for arbitrary m in, amongst others, [Bra98, WHB98, Tsu99]. An obvious effect of this evaluation method is that it introduces a measurement error of the expected objective function, i.e., noise. This error can be decreased by increasing the number of samples m. Tsutsui [Tsu99] distinguished between the extreme case of taking only one sample for the evaluation and the generalized version, naming the former the *Single Evaluation Mode* (SEM) and the latter the *Multi Evaluation Mode* (MEM). Here, we also adopt this terminology.

We observe a clear resemblance when comparing this approximation method to the approximation of the expected objective function for noisy objective functions of Eq. 5.12. Essentially there is no difference between the two as one could see the objective function value for each disturbance $f(\mathbf{x} + \mathbf{z}_i)$ as a noisy evaluation of the expected objective function at $\mathbf{x}$. Hence, the problem of finding robust optima can, in principle, be transformed into the problem of optimizing a noisy objective function, bearing in mind that the noise introduced by this

approximation method is non-stationary and, more importantly, biased with respect to the original objective function $f(\mathbf{x})$. Properties of noise introduced by input perturbations have been studied in, e.g., [BOS03, SBO04, BS06b].

For notational convenience, in this work we will include the number of samples in the identification of the particular MEM scheme. For instance, MEM5 will refer to the MEM evaluation scheme with $m = 5$ samples and MEM10 refers to the MEM evaluation scheme with $m = 10$ samples.

In order to obtain an insight into how the SEM and MEM evaluation scheme influence the performance of the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES and how they compare to canonical instances of these algorithms, we perform the following experiment:

Experiment 7.2.3 (Performance of the SEM/MEM evaluation scheme for finding robust optima)**:** We perform 50 runs of a $(5/2_{DI}, 35)$-σSA-ES (see Section 4.2.2) and a CMA-ES (see Section 4.2.3) using different evaluation schemes: the canonical evaluation scheme, the SEM evaluation scheme, the MEM5 evaluation scheme, and the MEM10 evaluation scheme. The experiments are performed on the 10-dimensional test problems: the Heaviside sphere problem (see Appendix B.2) and Branke's multipeak problem (see Appendix B.6). Each run has a budget of $10,000$ function evaluations.

The results of Experiment 7.2.3 are shown in Figure 7.10 by means of plots of the development of the median fitness, a posteriori approximated using Monte-Carlo integration with $m = 100$ samples. For the Heaviside sphere problem, the canonical implementations of the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES are clearly outperformed by the SEM/MEM evaluation schemes. That is, these canonical instances converge to the non-robust optimizer which is not the robust optimizer. Moreover, we observe the same effects of using more samples as can be seen with explicit averaging in case of noisy objective functions; using the SEM evaluation scheme yields a fast convergence, yet also stagnates at a certain distance to the optimizer. When using more samples for the MEM scheme, the convergence accuracy increases, but at the cost of slower convergence. For the sphere problem and for Branke's multipeak problem a major difference is the good performance of the myopic instances of the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES. For Branke's multipeak problem, this is surprising. Apparently, the bias of the myopic instances to converge to more robust peaks is already very high and combining this with the fact that the robust optimizer for this problem is a local optimizer of the original objective function, the canonical instances will converge with high precision and prove to be competitive alternatives.

Experiment 7.2.4 (The attraction of robust peaks)**:** For testing the attraction of robust peaks for the different MEM variants, we perform 500 runs of a $(5/2_{DI}, 35)$-σSA-ES (see Section 4.2.2) and a CMA-ES (see Section 4.2.3) on a 10-dimensional instance of Branke's multipeak problem (see Appendix B.6) using different evaluation schemes: the canonical evaluation scheme,

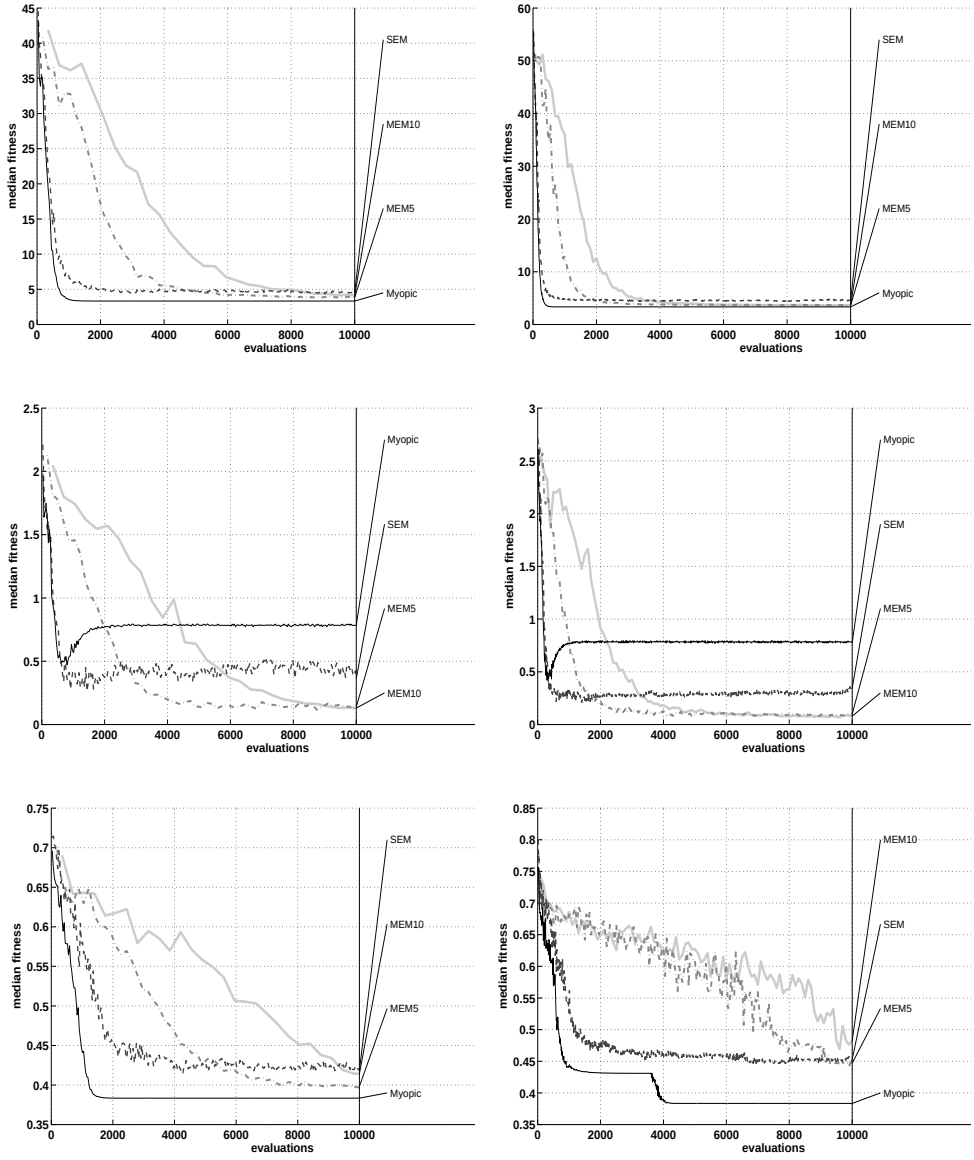

Figure 7.10: Results of Experiment 7.2.3. The performance of the $(5/2_{DI}, 35)$-σSA-ES (left column) and the CMA-ES (right column) using the SEM/MEM evaluation scheme with different sample sizes compared against canonical instances of the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES. The performance is measured in terms of median fitness approximated a posteriori using Monte-Carlo sampling with 100 samples. Top row: the results on the sphere problem. Middle row: the results on the Heaviside sphere. Bottom row: the results on Branke's multipeak problem. The results are obtained using 50 runs for each scheme.

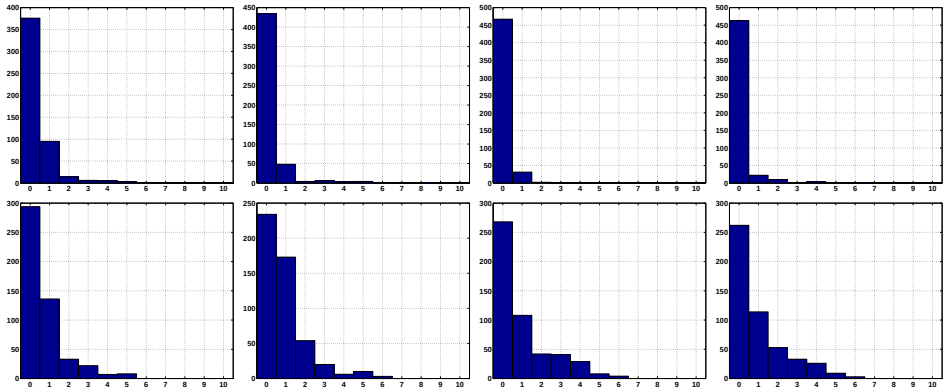

Figure 7.11: Histograms of the peak scores of the final solution of 500 optimization runs of the $(5/2_{DI}, 35)$-σSA-ES (top) and the CMA-ES (bottom) on a 10-dimensional instance of Branke's multipeak problem with varying evaluation schemes. From left to right: myopic, SEM, MEM5, and MEM10.

the SEM evaluation scheme, the MEM5 evaluation scheme, and the MEM10 evaluation scheme. For each run, an evaluation budget of $10,000$ function evaluations is considered.

The results of Experiment 7.2.4 are shown in Figure 7.11 by means of histograms of the peak scores of the final solution of all runs. For the $(5/2_{DI}, 35)$-σSA-ES it can be seen that the frequency of finding the most robust peak indeed slightly increases for MEM evaluation approaches. For the CMA-ES, almost the inverse seems to be happening. That is, comparing the myopic approach with the SEM approach for the CMA-ES, there is a huge drop in the frequency with which the robust optimizer is targeted. For the MEM5 and MEM10 approach, a slight improvement is obtained with respect to the SEM approach, but the bias for the myopic approach is still higher.

7.2.3 Reducing the Sampling Variance

For the MEM evaluation scheme, a modification proposed by Branke [Bra01] is to use Latin Hypercube Sampling (LHS) [MBC00] to obtain a well-distributed sample set of input disturbances in a box around the solution, and using these in a weighted way to determine the robustness, i.e.,

$$\hat{f}_{\exp}(\mathbf{x}) = \frac{\sum_{i=1}^{m} w(\mathbf{z}_i) f(\mathbf{x} + \mathbf{z}_i)}{\sum_{i=1}^{m} w(\mathbf{z}_i)}, \tag{7.18}$$

with $\{\mathbf{z}_1, \ldots, \mathbf{z}_m\} \sim \mathrm{LHS}(-\boldsymbol{\sigma}_\epsilon, \boldsymbol{\sigma}_\epsilon)$ being the set of sample points drawn using Latin Hypercube Sampling from an appropriately set hypercube $[-\boldsymbol{\sigma}_\epsilon, \boldsymbol{\sigma}_\epsilon]$ and $w(\mathbf{z}_i) \propto \mathrm{pdf}_{\boldsymbol{\delta}}(\mathbf{z}_i)$. In [Bra01], this modification was shown to yield a better convergence accuracy compared to the normal MEM evaluation scheme, attributed to a reduced sampling variance among the fitness estimates of the individuals in the population. Additionally, an extra gain in convergence

accuracy was reported when using the same set of disturbances for all individuals in the population.

For notational convenience, we will use the following notation to distinguish between the different variants of the SEM/MEM scheme. A subscript is used for indicating the sampling method when using Eq. 7.18, i.e., $\mathrm{MEM}_{\mathrm{MS}}$ when using Monte-Carlo sampling, and $\mathrm{MEM}_{\mathrm{LHS}}$ when using Latin Hypercube Sampling. Furthermore, the superscripts $^+$ and $^-$ are used to indicate whether or not the same disturbances are used for all the individuals in the current population, i.e., MEM^+ when using the same disturbances for all individuals in the population and MEM^- when resampling the disturbances for every individual in the population. When no subscript or superscript is used (as in Experiment 7.2.3), we assume that the evaluation follows Eq. 7.17 and different disturbance samples for every individual in the population.

In order to obtain an insight into how these different variants of the MEM scheme influence the performance of the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES, we perform the following experiment:

Experiment 7.2.5 (Performance of basic techniques for finding robust optima)**:** We perform 50 runs of a $(5/2_{DI}, 35)$-σSA-ES (see Section 4.2.2) and a CMA-ES (see Section 4.2.3) using the different variants of the MEM evaluation scheme, namely $\mathrm{MEM5}^-_{\mathrm{MC}}$, $\mathrm{MEM5}^+_{\mathrm{MC}}$, $\mathrm{MEM5}^-_{\mathrm{LHS}}$, and $\mathrm{MEM5}^+_{\mathrm{LHS}}$. The experiments are performed on the sphere problem (see Appendix B.1), the Heaviside sphere problem (see Appendix B.2), and Branke's multipeak problem (see Appendix B.6). Each run has a budget of $10,000$ function evaluations.

The results of Experiment 7.2.5 are shown in Figure 7.12 by means of plots of the development of the median fitness, a posteriori approximated using Monte-Carlo integration with 100 samples. For the sphere problem and for Branke's multipeak problem, the results confirm the results obtained by Branke [Bra01], i.e., using Latin Hypercube sampling yields better results than Monte-Carlo sampling and also using the same perturbations for all individuals in the current generations yields an additional gain in solution quality. The significance of these results can be read from the boxplots that are shown in Figure 7.13.

However, when looking at the results on the Heaviside sphere problem, remarkably different behavior can be observed. Here, especially for the CMA-ES, the LHS methods seem to be outperformed by the Monte-Carlo based methods and using the same perturbations for all individuals in the population yields a worse final solution quality. Looking at the convergence plot of the CMA-ES (Figure 7.12) we see that the schemes that reuse the same perturbations for all individuals in the current generation show divergent behavior for this particular test problem. The same divergent behavior is exhibited in the other schemes, but less drastically. For the $(5/2_{DI}, 35)$-σSA-ES, these effects are less severe. A possible explanation for this is that the two alterations of the MEM scheme exploit local symmetry, which is beneficial in case of the sphere and Branke's multipeak, but is harmful in case of the Heaviside sphere.

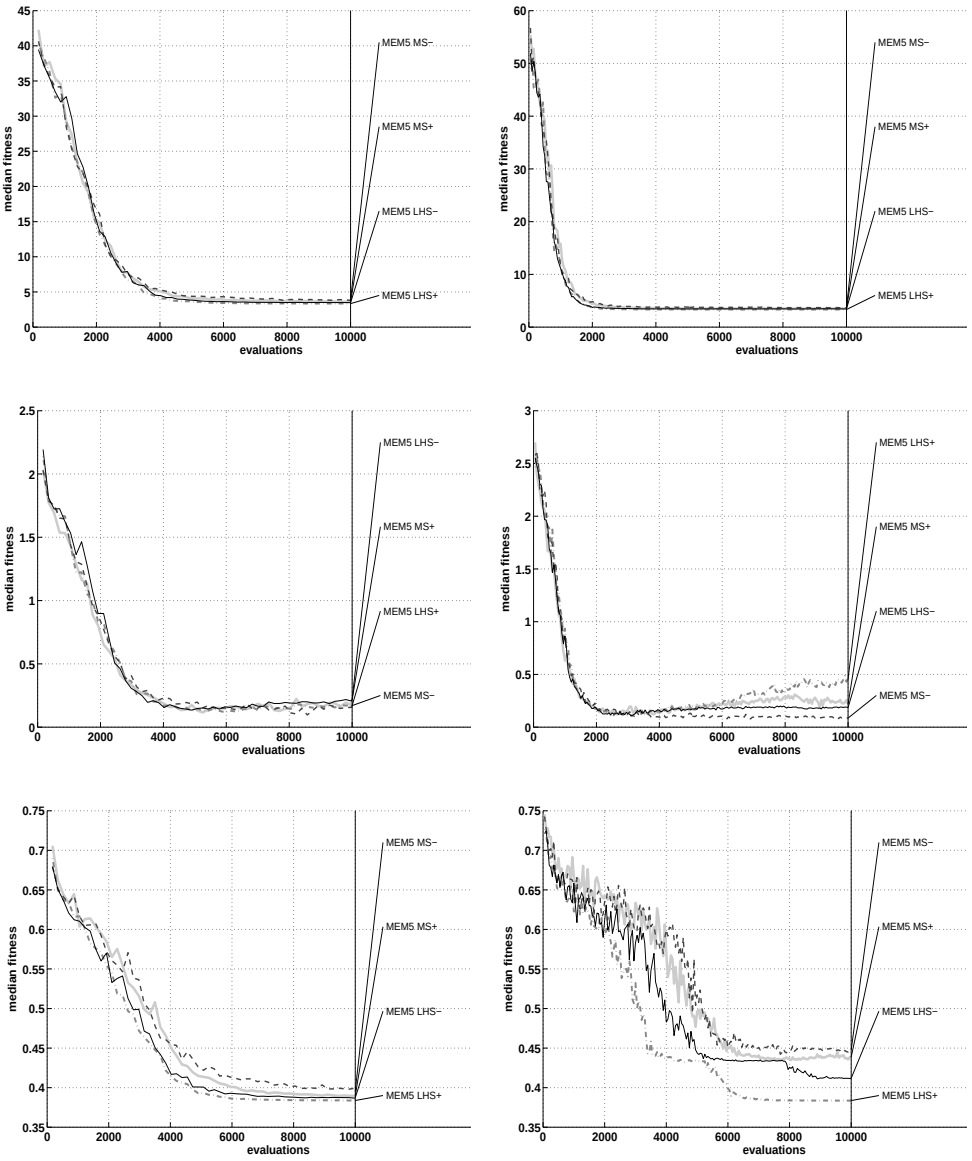

Figure 7.12: Results of Experiment 7.2.5. The performance of the $(5/2_{DI}, 35)$-σSA-ES (left column) and the CMA-ES (right column) using different variants of the MEM evaluation. The performance is measured in terms of median fitness approximated a posteriori using Monte-Carlo sampling with 100 samples. Top row: the results on the sphere problem. Middle row: the results on the Heaviside sphere problem. Bottom row: the results on Branke's multipeak problem. The results are obtained using 50 runs for each scheme.

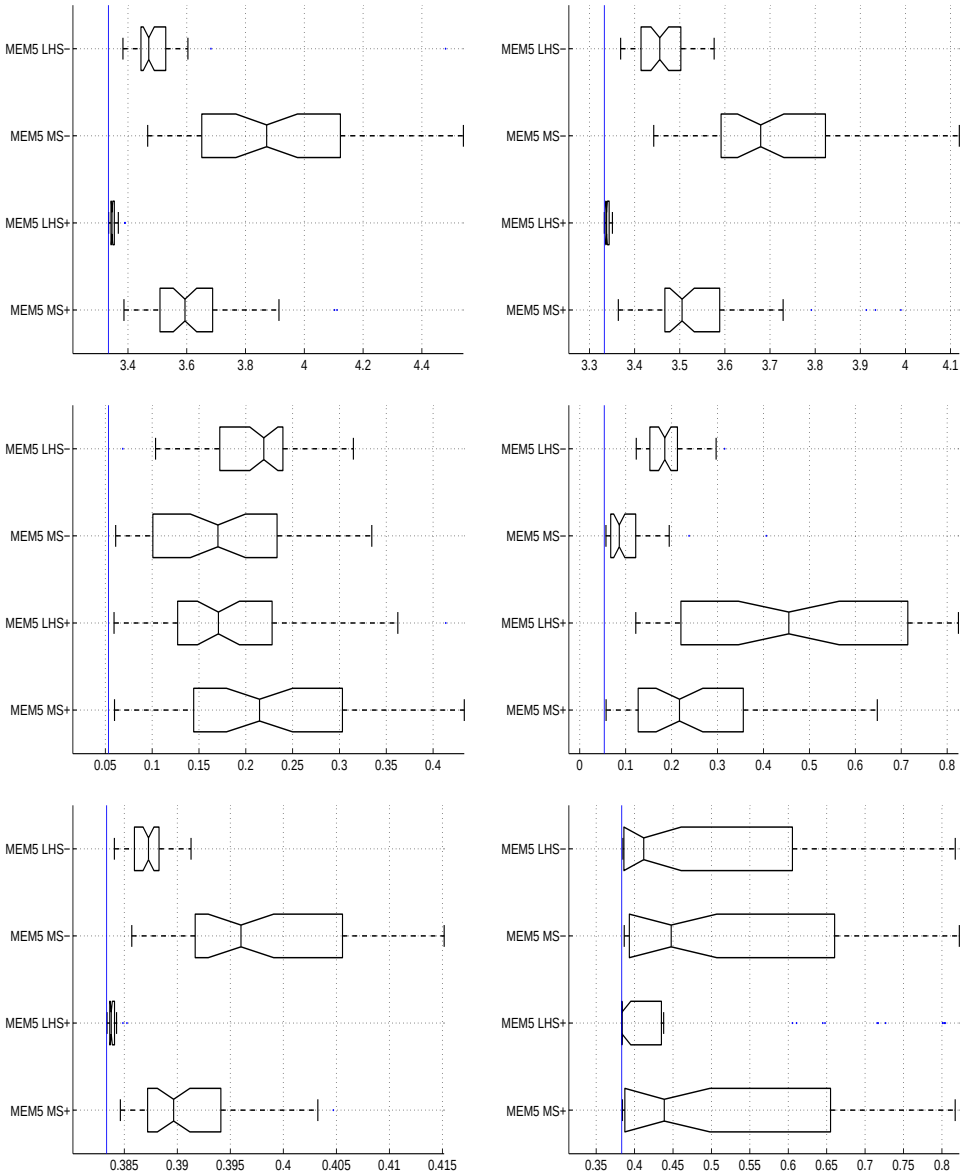

Figure 7.13: Results of Experiment 7.2.5. The final solution quality of the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES using different variants of the MEM evaluation. The solution quality is approximated a posteriori using Monte-Carlo sampling with 1000 samples. Top row: the results on the sphere problem. Middle row: the results on the Heaviside sphere problem. Bottom row: the results on Branke's multipeak problem. Left column: the $(5/2_{DI}, 35)$-σSA-ES. Right column: the CMA-ES. The results are obtained using 50 runs for each scheme.

From these results we can conclude that the proposed alternatives can be beneficial in cases where the objective function landscape is symmetric around the robust optimizer. However, when considering local discontinuities around the optimizer that shift the robust optimizer, they might yield divergent behavior. In the remainder of this chapter, we will consider the extremal cases of Monte-Carlo sampling without reusing the same disturbances for all individuals in the population (MEM5$^{-}_{\mathrm{MC}}$) and Latin Hypercube sampling in combination with using the same disturbances for all individuals in the population (MEM5$^{+}_{\mathrm{LHS}}$) for further investigation.

7.2.4 Implicit Versus Explicit Averaging for Finding Robust Optima

For noisy objective functions, we have observed that implicit averaging is competitive to or even better than explicit averaging. This is true when considering Gaussian additive noise. The MEM evaluation approach for finding robust optima is essentially generating noisy fitness evaluations, however, for smaller values of m, this noise cannot be assumed to be Gaussian. An interesting question is therefore whether implicit averaging is also applicable in these scenarios. In order to obtain an insight into the differences between implicit and explicit averaging in this scenario, we perform the following experiment:

Experiment 7.2.6 (Implicit versus explicit averaging for finding robust optima)**:** We perform 50 runs of a $(5/2_{DI}, 35)$-σSA-ES (see Section 4.2.2) and a CMA-ES (see Section 4.2.3) using two variants of the MEM evaluation scheme (using $m = 5$) and compare the results obtained with these two variants against two variants of the SEM scheme in which the population size is increased with a factor 5 (i.e., implicit averaging). The two MEM variants are MEM5$^{-}_{\mathrm{MC}}$ and MEM5$^{+}_{\mathrm{LHS}}$, and the two SEM variants are 5SEM^{-} and 5SEM^{+}. The experiments are performed on the sphere problem (see Appendix B.1), the Heaviside sphere problem (see Appendix B.2), and on Branke's multipeak problem (see Appendix B.6). Each run has a budget of $10,000$ function evaluations.

The results of Experiment 7.2.6 are shown in Figure 7.14 by means of plots of the development of the median fitness, a posteriori approximated using Monte-Carlo integration with 100 samples. From the results we see that, in general, the explicit resampling approaches outperform the implicit averaging schemes. That is, when not taking into account the divergent behavior obtained with the MEM5$^{+}_{\mathrm{LHS}}$ approach on the Heaviside sphere. Comparing this to the conclusion of the previous chapter, in which the implicit averaging scheme showed remarkably good performance, especially for the CMA-ES, this result is surprising. It is reasonable to assume that difference is caused by the fact that the noise that is introduced by using a SEM evaluation scheme is not Gaussian and non-stationary. However, this should be investigated in more detail.

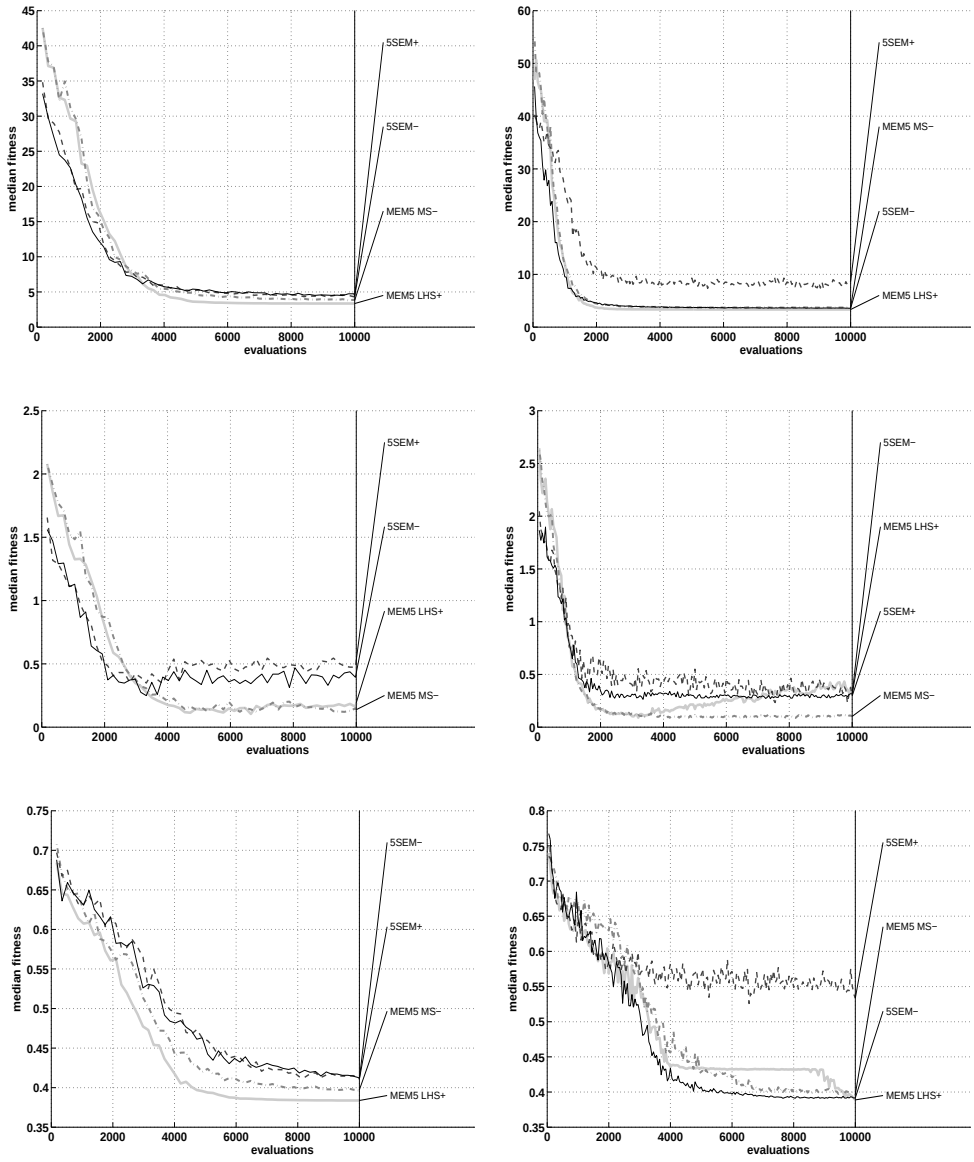

Figure 7.14: Results of Experiment 7.2.6. The performance, in terms of median fitness (a posteriori approximated using Monte-Carlo integration with 100 samples) of different variants of the SEM/MEM evaluation scheme, comparing implicit versus explicit averaging. Top row: the results on the sphere problem. Middle row: the results on the Heaviside sphere. Bottom row: the results on Branke's multipeak problem. Left column: the $(5/2_{DI}, 35)$-σSA-ES. Right column: the CMA-ES. The results are obtained using 50 runs for each scheme.

7.2.5 Adaptive Averaging for Finding Robust Optima

As mentioned in Section 7.2.2, using MEM evaluation methods for approximating the expected objective function is essentially the same as using explicit averaging for a noisy objective function. Moreover, also here we encounter the same trade-off between using few samples, yielding a low approximation accuracy or obtaining a high approximation accuracy at the cost of requiring many samples. A straightforward approach to deal with this problem is therefore to use the adaptive averaging techniques proposed in the context of noisy optimization for improving the convergence accuracy of schemes that aim to find robust solutions.

In [KRD+11] such an approach is followed. Here, the UH-CMA-ES proposed in [HNGK09] is adapted for resampling and the objective function is replaced by a $\mathrm{MEM}^{+}_{\mathrm{LHS}}$ evaluation scheme. The incorporation of this idea is straightforward: for any of the adaptive averaging techniques, a MEM evaluation scheme using m samples can be used instead of m times resampling the noisy objective function. The scheme that we will consider in this work as an implementation of this idea is shown in Algorithm 7.1. Here, the rank-change based adaptive averaging technique as described in Section 5.4.5 is modified for incorporation of any MEM evaluation scheme.

In order get an impression of how such a scheme performs in this setting, we consider the following experiment:

Experiment 7.2.7 (Performance of adaptive averaging for finding robust optima)**:** We perform 50 runs of a $(5/2_{DI}, 35)$-σSA-ES (see Section 4.2.2) and a CMA-ES (see Section 4.2.3) using a MEM evaluation scheme adopting the rank-change based adaptive averaging technique (see Section 5.4.5). The experiments are performed on the sphere problem (see Appendix B.1), the Heaviside sphere problem (see Appendix B.2), and on Branke's multipeak problem (see Appendix B.6). For the adaptive averaging schemes, the settings $\alpha = 1.2$ and $\theta = 0.6$ are used, according to [KRD+11]. Each run uses a budget of $10,000$ function evaluations.

The results of Experiment 7.2.7 are shown in Figure 7.15 by means of convergence plots of the median fitness (a posteriori approximated using Monte-Carlo integration with 100 samples). From the figure it can be seen that for the unimodal problems, the convergence speed in the early stages is much higher, however, for the sphere problem, the normal $\mathrm{MEM5}^{+}_{\mathrm{LHS}}$ yields a higher convergence accuracy than the adaptive averaging schemes. For the Heaviside sphere problem, the adaptive averaging approaches yield a higher convergence accuracy than their non-adaptive counterparts. For Branke's multipeak problem, using adaptive averaging seems not to be beneficial and might even be noted to be worse. The latter complies with the results reported in [KRD+11] that adaptive averaging seems only beneficial for improving the local convergence accuracy, but it does not increase the tendency to converge to the more robust peaks.

Algorithm 7.1: Rank-Change Based Adaptive Resampling for Finding Robust Optima

Procedure parameters: confidence level θ, averaging increment factor α

Procedure variables: sample size indicator m_{eval}, initialized at $m_{\text{eval}} = 2$

1. For all candidate solutions $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda$ obtain robustness estimates $\hat{f}_{\text{exp}1,\text{old}}, \ldots, \hat{f}_{\text{exp}\lambda,\text{old}}$ based on a MEM evaluation scheme using $m_1 = \lceil m_{\text{eval}}/2 \rceil$ perturbations for each individual

$$L^{\text{old}} = \{\hat{f}_{\text{exp}1,\text{old}}, \ldots, \hat{f}_{\text{exp}\lambda,\text{old}}\}. \tag{7.19}$$

2. Repeat step 1 using $m_2 = \lfloor m_{\text{eval}}/2 \rfloor$ perturbations for each individual and store the mean objective function values in the set $L^{\text{new}} = \{\hat{f}_{\text{exp}1,\text{new}}, \ldots, \hat{f}_{\text{exp}\lambda,\text{new}}\}$.

3. Compute the rank-changes $\Delta_1, \ldots, \Delta_\lambda$ using

$$\Delta_i = \text{rank}(L_i^{\text{new}}) - \text{rank}(L_i^{\text{old}}) - \text{signum}\left(\text{rank}(L_i^{\text{new}}) - \text{rank}(L_i^{\text{old}})\right). \tag{7.20}$$

4. Compute the uncertainty level based on the rank-changes

$$\begin{aligned} s \;=\; & \frac{1}{\lambda_{\text{reev}}} \sum_{i=1}^{\lambda_{\text{reev}}} (2|\Delta_i| \\ & -\Delta_\theta^{\lim}\left(\text{rank}\left(L_i^{\text{new}}\right) - \mathbb{I}\{L_i^{\text{new}} > L_i^{\text{old}}\}\right) \\ & -\Delta_\theta^{\lim}\left(\text{rank}\left(L_i^{\text{old}}\right) - \mathbb{I}\{L_i^{\text{old}} > L_i^{\text{new}}\}\right)), \end{aligned} \tag{7.21}$$

5. Update the sample size m_{eval} using the update rule

$$m_{\text{eval}} = \begin{cases} \alpha \cdot m_{\text{eval}} & , \text{if } s > 0 \\ m_{\text{eval}} & , \text{otherwise} \end{cases}. \tag{7.22}$$

6. Generate a ranking $\mathbf{x}_{1:\lambda}, \ldots, \mathbf{x}_{\lambda:\lambda}$ based on the combined fitness approximations

$$\hat{f}_{\text{exp}i} = \frac{\hat{f}_{\text{exp}1,\text{old}} + \hat{f}_{\text{exp}1,\text{new}}}{2}, \; i = 1, \ldots, \lambda. \tag{7.23}$$

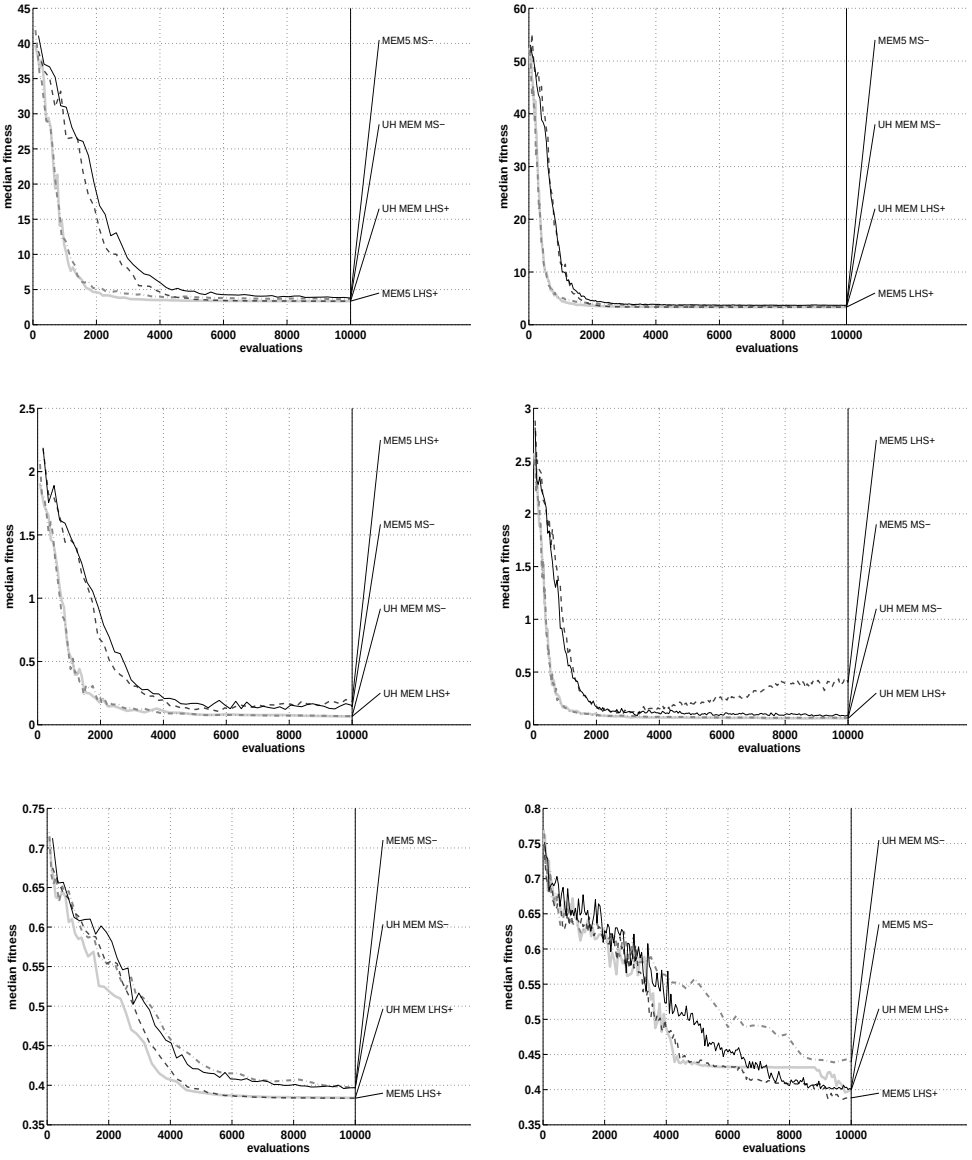

Figure 7.15: Results of Experiment 7.2.7. The performance, in terms of median fitness (a posteriori approximated using Monte-Carlo integration with 100 samples) comparing static explicit averaging versus adaptive averaging for finding robust optima. Top row: results on the Heaviside sphere. Bottom row: the results on Branke's multipeak problem. Left column: the $(5/2_{DI}, 35)$-σSA-ES. Right column: the CMA-ES. The results are obtained using 50 runs for each scheme.

7.2.6 Mutation as Robustness Tester

Beyer and Sendhoff [BS06a] also propose an adaptation of Evolution Strategies that can be classified as an adaptive averaging approach. However, it also introduces an alternative way of sampling for finding robust optima.

The focus in [BS06a] lies on optimization of the expected objective function with Gaussian input perturbations and they consider the SEM^- evaluation approach in the context of a $(\mu/\mu_I, \lambda)$-ES. For generating offspring, they suggest to include mutation itself as a robustness tester. That is, instead of generating an offspring $\mathbf{x}_\mathrm{o} = \mathbf{x}_\mathrm{p} + \mathbf{z}_\mathrm{o}$ and evaluating on $\mathbf{x}_\mathrm{o} + \mathbf{z}_\epsilon$, $\mathbf{z}_\epsilon \sim \mathrm{pdf}(\boldsymbol{\delta})$, they propose to interpret $\mathbf{z}_\epsilon + \mathbf{z}_\mathrm{o}$ as a mutation in its own right. In addition to this, they observe that selection tends to favor solutions with smaller disturbance vectors lengths $|\mathbf{z}_\epsilon|$. For this, they propose a control mechanism for adding input perturbations such that the variance amongst the μ selected parents is equal to the desired input perturbations. A third modification with respect to canonical Evolution Strategies is the inclusion of an adaptive averaging scheme. In order to prevent stagnation they propose an adaptive averaging technique that increases the population size when necessary (i.e., implicit averaging).

Technical Note 7.2 describes the mutation operator and the control mechanism for assuring that the measured disturbances of the μ selected offspring are distributed according to the targeted noise distribution $\mathrm{pdf}(\boldsymbol{\delta})$. As mentioned, the approach is including the input disturbances in the mutation operator, measuring the realized perturbation lengths of the selected individuals, and scaling the perturbation magnitude such that the realized perturbation length will approximate the desired input pertrubation length. Technical Note 7.3 describes the adaptive averaging approach suggested in [BS06a], which is based on increasing the population size with a certain factor whenever the uncertainty indicator $\overline{\Delta F}$ is below 0, which is checked every $\Delta g = n$ generations. The uncertainty indicator is based on tracking the improvement/decrement of the average population fitness of the select individuals.

The approach suggested in [BS06a] showed promising results on a number of test problems as compared to myopic approaches. The results were obtained using a $(\mu/\mu_I, \lambda)$-ES. For integration of this concept into a CMA-ES, it should take into account the full covariance matrix in the control mechanism for assuring that the measured disturbances of the selected offspring are equal to the targeted disturbances. Furthermore, note that the proposed scheme is specifically designed for SEM^- approaches. When using a SEM^+ or MEM^+, the observation of selection favoring smaller disturbance lengths does not hold anymore, because the disturbances are reused for all individuals in the population. One could think of using these approaches as an alternative fix to prevent selection from having this bias.

Technical Note 7.2: Mutation as Robustness Tester

Consider input perturbations $\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \mathbf{D})$, with $\mathbf{D} = diag((\sigma_\epsilon)_1^2, \ldots, (\sigma_\epsilon)_n^2)$. The mutation operator of Evolution Strategies can be adapted as

$$x_i = \langle x \rangle_i + \sqrt{\sigma_i^2 + \epsilon_i^2} \cdot \mathcal{N}(0,1)\ ,\ i = 1, \ldots, n, \tag{7.24}$$

where $\boldsymbol{\epsilon} = [\epsilon_1, \ldots, \epsilon_n]$ is not equivalent to the desired/targeted input noise $\boldsymbol{\sigma}_\epsilon = [(\sigma_\epsilon)_1, \ldots, (\sigma_\epsilon)_n]$, but is controlled such that the observed standard deviations $\mathbf{d} = [d_1, \ldots, d_n]$ of the μ selected individuals are close to $\boldsymbol{\sigma}_\epsilon$. This is accomplished by using the following update rule after every generation:

$$\epsilon_i = \epsilon_i \exp(\tau_\epsilon \cdot \text{sign}((\sigma_\epsilon)_i - d_i)), \tag{7.25}$$

where $\tau_\epsilon = c_x/3$ is a damping constant and $c_x = n^{-1}$ is a cumulation rate.

The observed standard deviations $\mathbf{d} = [d_1, \ldots, d_n]$ are measured over the main axes of μ selected individuals, which in this case are computed using a cumulated version of $d_i = \sqrt{\text{Var}[\{(\mathbf{x}_{1:\lambda})_i, \ldots, (\mathbf{x}_{\mu:\lambda})_i\}]}$:

$$d_i = \sqrt{\overline{x_i^2} - \overline{x}_i^2}, i = 1, \ldots, n, \tag{7.26}$$

with

$$\overline{x}_i = (1 - c_x)\overline{x}_i + c_x \langle x_i \rangle, \tag{7.27}$$

$$\overline{x_i^2} = (1 - c_x)\overline{x_i^2} + c_x \langle x_i^2 \rangle, \tag{7.28}$$

$$\langle x_i \rangle = \frac{1}{\mu} \sum_{m=1}^{\mu} (\mathbf{x}_{m:\lambda})_i, \tag{7.29}$$

$$\langle x_i^2 \rangle = \frac{1}{\mu} \sum_{m=1}^{\mu} (\mathbf{x}_{m:\lambda})_i^2. \tag{7.30}$$

Technical Note 7.3: Adaptive Population Size Control

As an uncertainty indicator, the average parental fitness change ΔF is given by

$$\Delta F = \langle F \rangle^{(g)} - \langle F \rangle^{(g-1)}, \tag{7.31}$$

where $\langle F \rangle^{(g)} = \frac{1}{\mu} \sum_{i=1}^{\mu} \hat{f}_{\text{eff}\,i:\lambda}$ for the current generation g. This indicator is smoothened as

$$\overline{\Delta F} = (1 - c_f)\overline{\Delta F} + c_f(\langle F \rangle^{(g)} - \langle F \rangle^{(g-1)}). \tag{7.32}$$

This indicator is checked every $\Delta g = n$ generations and when $\overline{\Delta F} \leq 0$, then the uncertainty treatment mechanism should be applied:

$$\mu = \lceil \alpha\mu \rceil, \tag{7.33}$$
$$\lambda = \lceil \mu/\eta \rceil, \tag{7.34}$$

with $\eta = \mu_0/\lambda_0$.

7.2.7 Archiving for Finding Robust Optima

Using resampling methods for approximating the robustness of candidate solutions can become computationally expensive. In an attempt to reduce the number of objective function evaluations, Branke [Bra98] suggested to evaluate each individual based on its own fitness and the fitness of the other individuals in the neighborhood. More specifically, the expected fitness of each individual $\mathbf{x}$ is approximated as the weighted mean of its own fitness and the fitness of previously evaluated neighboring points $\mathbf{x}'$ using

$$\hat{f}_{\text{exp}}(\mathbf{x}) = \frac{\sum_{\mathbf{x}'} w(\mathbf{x}' - \mathbf{x}) \cdot f(\mathbf{x}')}{\sum_{\mathbf{x}'} w(\mathbf{x}' - \mathbf{x})}, \tag{7.35}$$

where $w(\mathbf{x}'-\mathbf{x})$ is a weight function for which it should hold that $w(\mathbf{x}'-\mathbf{x}) \propto \text{pdf}_{\boldsymbol{\delta}}(\mathbf{x}'-\mathbf{x})$ (cf. Eq. 7.18). Although computing the distances between each pair $(\mathbf{x}, \mathbf{x}')$ introduces an overhead cost, in many cases it is fair to assume that the cost of evaluating one candidate solution is larger than this extra overhead cost. In [KEB10], this idea was extended by means of a method to obtain a more representative archive, which is also used in [SRS11].

A way in which such a scheme can be incorporated into an Evolutionary Algorithm is depicted in Technical Note 7.4. It is similar to a canonical evolution cycle, only a few extra steps are included involving the evaluation of candidate solutions. Note that this framework differs slightly from the description provided in [KEB10].

The evaluation procedure knows two stages. The first stage concerns updating the archive. An optional step is to evaluate each individual precisely. Also, it is checked whether the archive is representative enough to be used for obtaining approximations of the expected fitness. If not, then additional samples are chosen and added to the archive. The approach of Branke [Bra98] omits the latter step, but evaluates each individual precisely. The approach presented

in [KEB10] includes a method for checking the archive and selecting appropriate additional sample points which will be described below. In the second stage, appropriate archive points are selected for each individual and based on that approximations of the expected fitness are obtained following Eq. 7.35. An optional operation after each generation is to include a method to clean up the archive and prevent it to grow out of bounds.

The separation between the archive maintenance and evaluation (which is a difference with respect to the approach presented in [KEB10]) is to be recommended because during the first stage of this algorithm, the archive will be filled with samples that could be relevant for all individuals. Hence, in order to allow all individuals to benefit from the additional samples taken in the first stage, the evaluation stage is decoupled from the first stage. This should prevent the selection from being biased due to the order in which the individuals are handled.

A difference between the approach presented in [Bra98] and the alternative approach presented in [KEB10] is the way in which additional archive points are selected. In [Bra98] this is based on precise evaluation of the candidate solutions in the population, however, a major drawback of this is that the points in the archive are by no means guaranteed to be well-spread over the region of possible variation for each candidate solution. That is, the archive points are selected by the optimization algorithm, which yields a distribution of points that is dependent on the way in which subsequent candidate solutions are selected. Especially in focused searching strategies such as Evolution Strategies, this might lead to an archive that holds limited information about the (local) objective function landscape. In [KEDB10b], an approach is suggested to counter this drawback. The general idea of this approach is to generate a reference sample set of disturbances that should represent an ideal sample set. For each individual, additional sampling is determined by the reference set point that is least represented in the archive. Hence, additional sampling is done in the parts of the regions of uncertainty that are not underrepresented in the archive. By using a Latin Hypercube Sampling reference set, it can be assumed that the reference set is space filling and clustering is avoided.

Algorithm 7.2 describes the archive based selection method as proposed in [KEDB10b]. It takes as arguments a reference sample set $\mathbf{X}_{\mathrm{ref}} = \{\mathbf{x}_{\mathrm{r}}^{(1)}, \ldots, \mathbf{x}_{\mathrm{r}}^{(m)}\}$ of m samples, which should be the LHS reference set for an individual for which a robustness estimate is required, and the archive $\mathbf{A} = \{(\mathbf{x}_{\mathrm{a}}^{(1)}, f_{\mathrm{a}}^{(1)}), (\mathbf{x}_{\mathrm{a}}^{(2)}, f_{\mathrm{a}}^{(2)}), \ldots\}$ which contains design point / objective function value tuples.

For each reference sample $\mathbf{x}_{\mathrm{r}}$ in $\mathbf{X}_{\mathrm{ref}}$ the algorithm searches for the closest point in the archive. Then, for each of these selected archive points, it is checked whether the reference points for which they are selected are also the closest reference points. The archive points for which this is the case are then, together with their objective function values, added to the set $\mathbf{A}_{\mathrm{sel}}$ of selected archive points. For the archive points for which this is not the case one can conclude that the region around its reference point is underrepresented in the archive (making this reference point a good candidate for extra sampling). It is therefore added to the set $\mathbf{X}_{\mathrm{cand}}$

Technical Note 7.4: General Framework of an Archive Based Evolutionary Algorithm for Finding Robust Solutions

initialize parent population and archive
while not terminate **do**
  generate offspring from parents
  for each individual **do**
    (**optional: evaluate** individual and **add** to archive)
    if not enough samples **or** no appropriate sample set available **then**
      find appropriate additional sample points
      evaluate additional sample points and **add** to archive
    end if
  end for
  for each individual **do**
    select archive points for effective fitness approximation
    compute effective fitness approximation using the archive
  end for
  select best individual
  (**optional: clean up** archive)
end while

Algorithm 7.2: Reference Set Based Archive Selection for Finding Robust Optima

input: reference sample set $\mathbf{X}_{\mathrm{ref}}$, archive $\mathbf{A}$

output: selected archive points $\mathbf{A}_{\mathrm{sel}}$, sampling candidates $\mathbf{X}_{\mathrm{cand}}$

1: $\mathbf{A}_{\mathrm{sel}} \leftarrow \emptyset$
2: $\mathbf{X}_{\mathrm{cand}} \leftarrow \emptyset$
3: **for each** $\mathbf{x}_{\mathrm{r}} \in \mathbf{X}_{\mathrm{ref}}$ **do**
4: $\quad (\mathbf{x}_{\mathrm{a}}, f_{\mathrm{a}}) \leftarrow \{(\mathbf{x}_{\mathrm{a}}, f_{\mathrm{a}}) \in \mathbf{A} \mid \text{for all } (\mathbf{x}'_{\mathrm{a}}, f'_{\mathrm{a}}) \in \mathbf{A} : d(\mathbf{x}_{\mathrm{r}}, \mathbf{x}_{\mathrm{a}}) \leq d(\mathbf{x}_{\mathrm{r}}, \mathbf{x}'_{\mathrm{a}})\}$
5: $\quad$ **if** $(\text{for all } \mathbf{x}'_{\mathrm{r}} \in \mathbf{X}_{\mathrm{ref}} \ : \ d(\mathbf{x}_{\mathrm{r}}, \mathbf{x}_{\mathrm{a}}) \leq d(\mathbf{x}'_{\mathrm{r}}, \mathbf{x}_{\mathrm{a}}))$ **then**
6: $\quad\quad \mathbf{A}_{\mathrm{sel}} \leftarrow \mathbf{A}_{\mathrm{sel}} \cup \{(\mathbf{x}_{\mathrm{a}}, f_{\mathrm{a}})\}$
7: $\quad$ **else**
8: $\quad\quad \mathbf{X}_{\mathrm{cand}} \leftarrow \mathbf{X}_{\mathrm{cand}} \cup \{\mathbf{x}_{\mathrm{r}}\}$
9: $\quad$ **end if**
10: **end for**
11: **return** $(\mathbf{A}_{\mathrm{sel}}, \mathbf{X}_{\mathrm{cand}})$

of candidate sample points. In case that all reference points have been assigned archive points, then the reference point of the archive point / reference point pair that are located farthest apart from each other is selected as candidate for extra sampling[1]. This will assure that the archive is always updated in the region in which it needs the extra samples the most.

Figure 7.16 illustrates the scheme: Step **a)** displays archive points using the $\bullet$-symbol, the reference samples with the $\otimes$-symbol, and the box represents the η-neighborhood of the solution to be evaluated. In step **b)** for each reference point, the closest archive point is identified. In step **c)** it is checked whether the reference points are also the closest reference points of their selected archive points and in step **d)** the archive points for which this is the case are selected as archive points (solid circles) and the reference points for which this is not the case are selected as candidates for additional sampling (dashed circles).

The framework provided in Technical Note 7.4 combined with the archive selection scheme of Algorithm 7.2 yields an evaluation scheme for finding robust optima that can be integrated within any type of Evolutionary Algorithm. In this work, we adopt a slightly modified version of the scheme as considered in [KEDB10b]. It is presented in Algorithm 7.3. For

[1] This step is not shown in the algorithmic description of Algorithm 7.2.

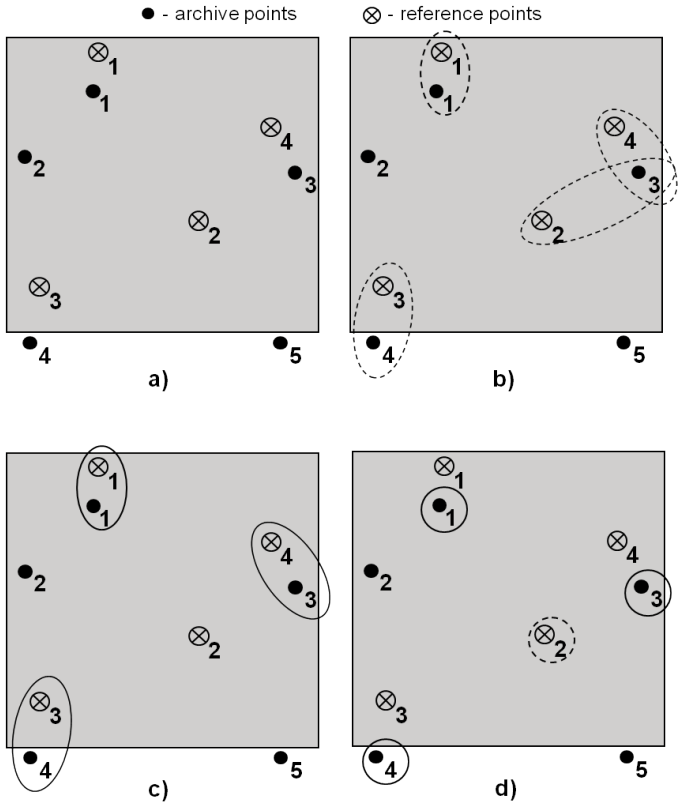

Figure 7.16: An illustration of the reference set based archive selection method proposed in [KEDB10b]. Step **a)**: the archive points are displayed with the ●-symbol, the reference samples with the ⊗-symbol, and the box represents the η-neighborhood of the to be evaluated solution. Step **b)**: for each reference point, the closest archive point is identified. Step **c)**: check whether the reference points are also the closest reference points of their selected archive points. Step **d)**: the archive points for which this is the case are selected as archive points (solid circles) and the reference points for which this is not the case are selected as candidates for additional sampling (dashed circles).

each individual, a reference set $\mathbf{X}_{\mathrm{ref}}$ of m samples is used to obtain a set of selected archive points $\mathbf{A}_{\mathrm{sel}} = \{(\mathbf{x}_{\mathrm{s}}^{(1)}, f_{\mathrm{s}}^{(1)}), (\mathbf{x}_{\mathrm{s}}^{(2)}, f_{\mathrm{s}}^{(2)}), \ldots\}$ and a set $\mathbf{X}_{\mathrm{cand}} = \{\mathbf{x}_{\mathrm{c}}^{(1)}, \mathbf{x}_{\mathrm{c}}^{(2)}, \ldots\}$ of suggested candidates for extra sampling. Then, in this variant, instead of all, only one of the suggested candidate points for extra sampling is evaluated and added to $\mathbf{A}$. An expected fitness approximation is thereafter generated by considering all archive points and using the weighted function as in Eq. 7.35.

Note that it is also possible to take for each individual a sample set using the reference set based archive selection method. However, it should be noted that this set should be selected anew in order to allow all individuals to use the same archive for evaluation. Using the complete archive for evaluation is justifiable when assuming that the archive will be locally well-spread (because additional samples will be taken in the least represented areas). An additional advantage of this is allows to take arbitrarily many samples for the evaluation (in the sense of adaptive averaging) and only requires a setting for m for the maintenance of the archive.

In [KEB10] an example of the behavior of this way of maintaining an archive is given by showing the archive development on a two-dimensional version of the Heaviside sphere (see Appendix B.2). Here, the archive based reference set selection scheme (named *ABRSS*) is compared against the archiving method as considered in [Bra98] (named *PROX*), both incorporated into a CMA-ES (see Section 4.2.3). Figure 7.17 shows, for a run of both the ABRSS scheme and the PROX scheme, the archive after 100, 200, and 300 generations respectively. Here, the small plots inside each plot are magnified versions on the interval $[0, 2]^2$ (i.e., the interval around the optimum). The asset of the archive maintenance scheme can be seen clearly; whereas the PROX method zooms in on a narrow region, the ABRSS scheme takes into account the whole region of uncertainty around the point on which it zooms in, leading to a well-spread set of archive points around the optimum (in effect generating more accurate fitness approximations). Although in higher dimensions it will take more time to build up a well-spread archive, the ABRSS scheme, in contrast to the PROX scheme, has the potential to do so.

Finally, in order to get an impression of the performance of this archive maintenance scheme on 10-dimensional problems, we consider the following experiment:

Experiment 7.2.8 (Performance of archive based evaluation for finding robust optima)**:** We perform 50 runs of a $(5/2_{DI}, 35)$-σSA-ES (see Section 4.2.2) and a CMA-ES (see Section 4.2.3) using the archive based reference set approach for assessment of the expected fitness as described in Algorithm 7.3. The experiments are performed on the sphere problem (see Appendix B.1), the Heaviside sphere problem (see Appendix B.2), and on Branke's multipeak problem (see Appendix B.6). For the archive based reference set selection, a sample size of $m = 2n = 20$ is used. Each run uses a budget of $10,000$ function evaluations.

Algorithm 7.3: Archive Maintenance in Evolutionary Algorithms for Finding Robust Optima

Procedure parameters: the number of samples used for generating the reference set m

Procedure variables: solution archive A

1: $\mathbf{P} \leftarrow$ *generate_initial_population*()
2: $A \leftarrow \emptyset$
3: **while** not terminate **do**
4: $\quad\mathbf{O} \leftarrow$ *generate_offspring*($\mathbf{P}$)
5: $\quad$**for each** $\mathbf{x} \in \mathbf{O}$ **do**
6: $\quad\quad\mathbf{X}_{\mathrm{ref}} \leftarrow$ *latin_hypercube_sampling*($\mathbf{x}, \boldsymbol{\sigma}_\epsilon, m$)
7: $\quad\quad(\mathbf{A}_{\mathrm{sel}}, \mathbf{X}_{\mathrm{cand}}) \leftarrow$ *reference_set_based_archive_selection*($\mathbf{X}_{\mathrm{ref}}, \mathbf{A}$)
8: $\quad\quad$**for a random** $\mathbf{x}_c \in \mathbf{X}_{\mathrm{cand}}$ **do**
9: $\quad\quad\quad f_{\mathrm{c}} \leftarrow$ *evaluate*($\mathbf{x}_{\mathrm{c}}$)
10: $\quad\quad\quad A \leftarrow A \cup \{(\mathbf{x}_{\mathrm{c}}, f_{\mathrm{c}})\}$
11: $\quad\quad$**end for**
12: $\quad$**end for**
13: $\quad$**for each** $\mathbf{x} \in \mathbf{O}$ **do**
14: $\quad\quad\hat{f}_{\mathrm{exp}}(\mathbf{x}) \leftarrow (\sum_{\{\mathbf{x}_{\mathrm{a}}, f_{\mathrm{a}}\} \in \mathbf{A}} w(\mathbf{x}_{\mathrm{a}} - \mathbf{x}) \cdot f_{\mathrm{a}}) / (\sum_{\{\mathbf{x}_{\mathrm{a}}, f_{\mathrm{a}}\} \in \mathbf{A}} w(\mathbf{x}_{\mathrm{a}} - \mathbf{x}))$
15: $\quad$**end for**
16: $\quad\mathbf{P} \leftarrow$ *select*($\mathbf{O}$)
17: **end while**

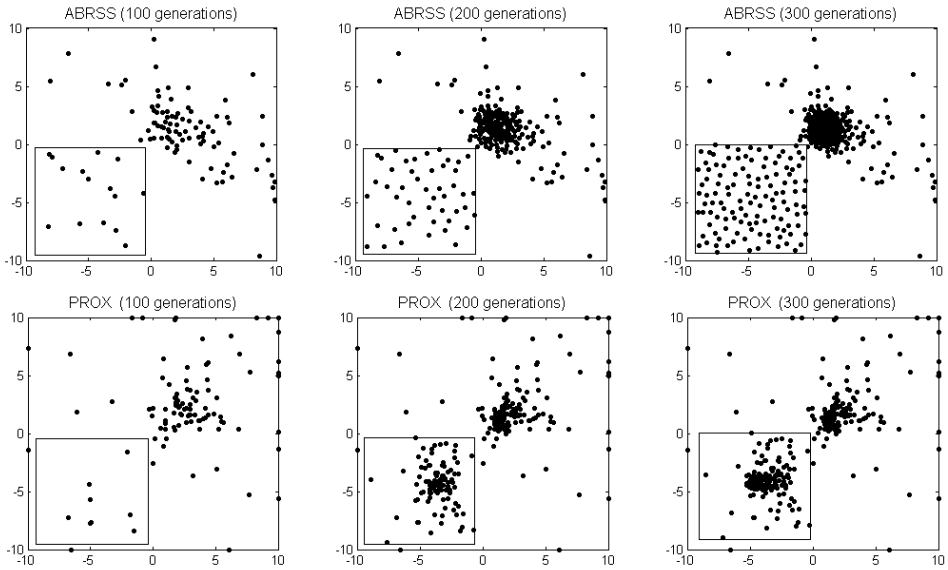

Figure 7.17: Archive after 100, 200, and 300 generations of the ABRSS scheme (top row) and PROX scheme (bottom row) on a two-dimensional instance of the heaveyside sphere. The zoom window magnifies the interval $[0, 2]^2$.

Figure 7.18 and Figure 7.19 show the results of Experiment 7.2.8. In Figure 7.18 the performance is shown in terms of median fitness (a posteriori approximated using Monte-Carlo integration with 100 samples) and in Figure 7.19 the final solution quality (approximated a posteriori using Monte-Carlo sampling with 1000 samples) is displayed. From the figures it can be seen that the methods incorporating the ABRSS scheme outperforms the $\mathrm{MEM}^{-}_{\mathrm{MS}}$ scheme on all runs. Comparing it, however, to the $\mathrm{MEM}^{+}_{\mathrm{LHS}}$ scheme, it is outperformed on the sphere problem, and on Branke's multipeak problem, but is clearly better on the Heaviside sphere. The results suggest that the ABRSS is indeed capable on zooming in on the robust optimum, but for 10-dimensional problem spaces, it might take some while for the archive to fill. When the objective function landscape is symmetric around the (robust) optimum, the $\mathrm{MEM}^{+}_{\mathrm{LHS}}$, that exploits such symmetry, will be able to zoom in on the robust optimum more quickly. The convergence speed obtained by using the ABRSS scheme (see Figure 7.19) is much higher than when using the MEM schemes.

In conclusion, using an archive of previously evaluated solutions provides a way for efficiently using objective function evaluations. By using the archive maintenance approach as proposed in [KEB10], it can be assured that the archive will be updated in the places where it is underrepresented. This approach is suitable especially for low-dimensional search spaces (dimension $\lesssim 10$). The higher the dimensionality of the search space, the longer it will take before the archive is filled sufficiently well to be representative. Regarding the computational

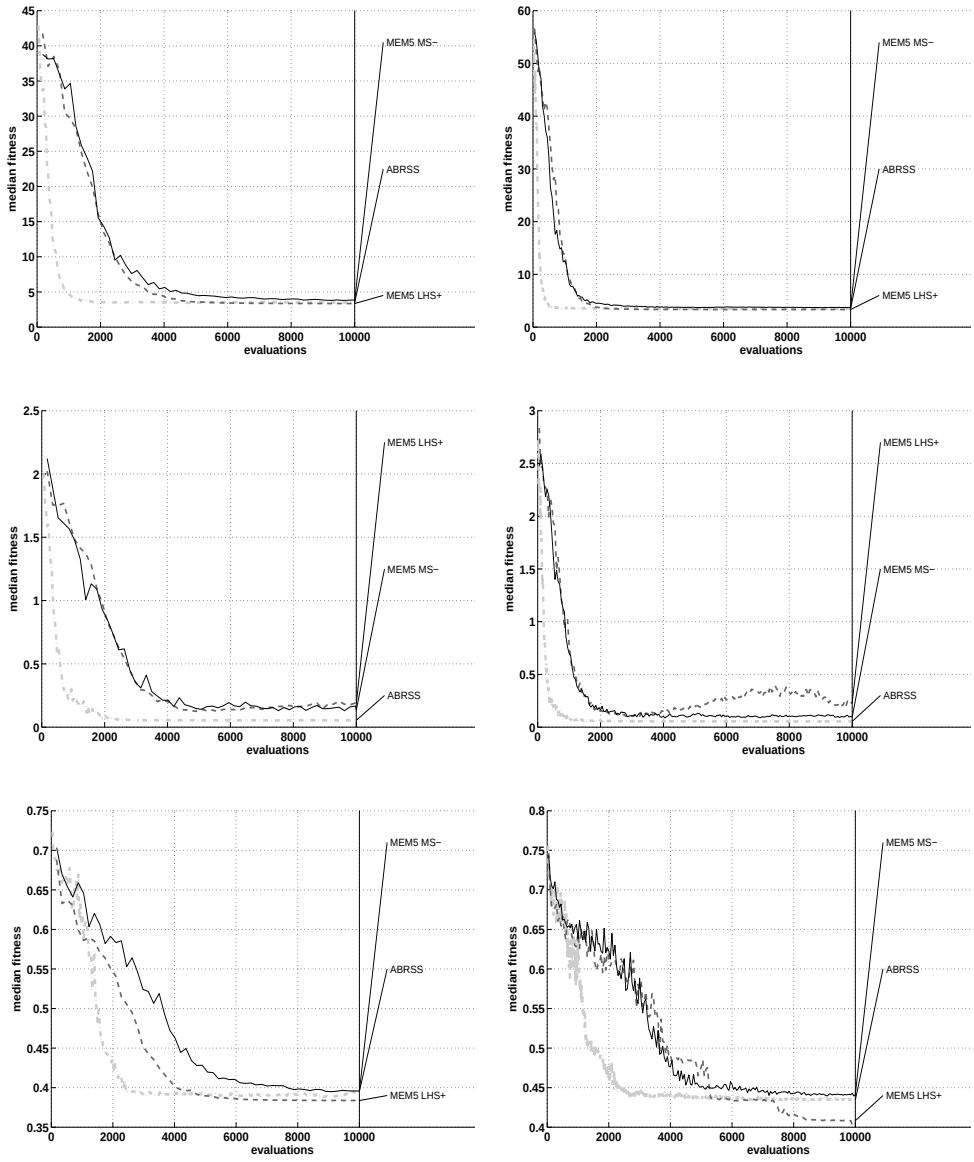

Figure 7.18: Results of Experiment 7.2.8. The performance, in terms of median fitness (a posteriori approximated using Monte-Carlo integration with 100 samples) of the ABRSS evaluation scheme incorporated into the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES, compared against two variants of the MEM evaluation schemes. Top row: results on the sphere. Middle row: results on the Heaviside sphere. Bottom row: the results on Branke's multipeak problem. Left column: the $(5/2_{DI}, 35)$-σSA-ES. Right column: the CMA-ES. The results are obtained using 50 runs for each scheme.

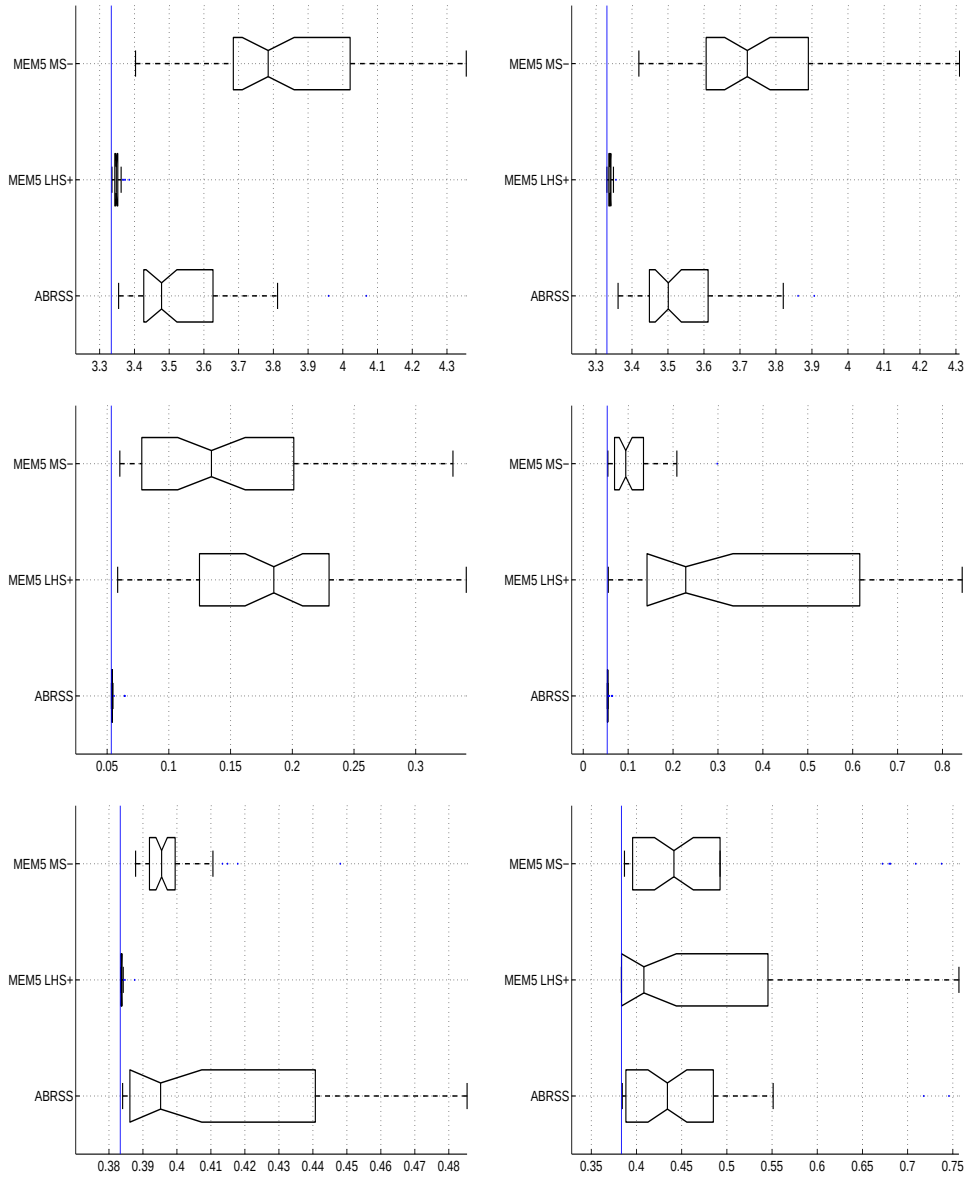

Figure 7.19: Results of Experiment 7.2.8. The final solution quality of the ABRSS evaluation scheme incorporated into the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES, compared against two variants of the MEM evaluation schemes. The solution quality is approximated a posteriori using Monte-Carlo sampling with 1000 samples. Top row: the results on the sphere problem. Middle row: the results on the Heaviside sphere problem. Bottom row: the results on Branke's multipeak problem. Left column: the $(5/2_{DI}, 35)$-σSA-ES. Right column: the CMA-ES. The results are obtained using 50 runs for each scheme.

cost of using the archive maintenance approach as considered in this section, it should be noted that a serious additional overhead is introduced by following this method. It is therefore specifically useful when the (computational) cost of an objective function evaluation is high.

7.2.8 Metamodeling for Finding Robust Optima

Metamodeling approaches are closely related to archive based approaches and are also particularly useful when objective function evaluations are expensive. The general idea is to construct (local) approximation models (or metamodels) of the objective function for which evaluation is cheap and obtain accurate robustness approximations based on these metamodels. In the context of classical optimization using Evolutionary Algorithms, metamodeling techniques know many applications (for an overview, see, e.g., [Jin05]) and it is also applied in the context of noisy optimization (see Section 5.5). In the context of finding robust optima, for which the demand for objective function evaluations is high, the application possibilities are apparent. Approaches that are proposed in this context are, e.g., [ONL06, PBJ06, PL09, KEDB10a].

The way in which metamodeling techniques can be integrated into a scheme for finding robust optima depends on a number of matters, i.e.,

- the modeling assumptions and modeling approach used for constructing the metamodels,
- the way in which archive points are selected for constructing metamodels,
- the frequency of the model updates / regenerations and the usage of the metamodels,
- the robustness measure and approximation method used on the metamodels.

Studying and testing all possible configurations for these different issues is obviously infeasible. Regarding the modeling assumptions and the modeling approach, this depends largely on the problem at hand and the same holds for the robustness measure and the approximation method. As an indication of the variety of possibilities, Paenke et al. [PBJ06] consider linear interpolation, quadratic interpolation, linear regression, and quadratic regression, Ong et al. [ONL06] consider radial basis functions, and Poles and Locison [PL09] consider polynomial models. The way in which the archive is maintained and the frequency of the model updates are more general choices. For the former, choosing additional sample points is much related to the discussion of the Section 7.2.7. The quality of a metamodel depends largely on the available sample set, which should contain "sufficient" information for generating an accurate model (see Section 5.5.3 for a discussion). As an indication for the possibilities regarding the frequency of the model updates and the usage of the metamodels, four possibilities studied by Paenke et al. [PBJ06] are:

- **Singe model**: generate a metamodel for each individual separately.

- **Nearest model**: construct a metamodel for each individual separately, but use for each sample point the closest model to estimate its objective function value.
- **Ensemble**: construct a metamodel for each individual in the population and use the ensemble of metamodels of all individuals in a weighted way to obtain objective function value approximations.
- **Multiple models**: generate a separate metamodel for each sample point that is to be evaluated.

Technical Note 7.5 shows a (possible) general framework of how metamodeling techniques can be integrated into an Evolutionary Algorithm for finding robust optima. For each offspring, a suitable set of archive points is selected which can be used to construct a local metamodel. In case such a suitable set of archive points does not exist, additional samples are taken, evaluated on the original objective function and also added to the archive. Thereafter, the fitness is determined by obtaining a robustness approximation based on the local metamodel or (optional) on the ensemble of metamodels of all offspring.

Note that the framework of Technical Note 7.5 much resembles the framework shown in Technical Note 7.4 on page 165. In fact, it can be argued that the archiving method of Section 7.2.7 is a simple form of metamodeling.

In this section, we will restrict ourselves to one configuration of the framework of Technical Note 7.5. This configuration is similar to the one considered in [KRD$^+$11] and serves as an example case to represent the metamodeling based approaches. This approach uses ordinary Kriging as metamodeling technique, which is briefly summarized in Appendix C. For maintaining an archive, the archive maintenance method as presented in Section 7.2.7 is used. A separate model is constructed for each individual in the population (i.e., the single model is followed). The robustness measure that will be adopted is the expected fitness, approximated using a $\mathrm{MEM}^+_{\mathrm{LHS}}$ evaluation scheme. The implementation of this specific configuration is described in Algorithm 7.4.

Experiment 7.2.9 (Performance of Kriging based evaluation for finding robust optima)**:** We perform 50 runs of a $(5/2_{DI}, 35)$-σSA-ES (see Section 4.2.2) and a CMA-ES (see Section 4.2.3) using the Kriging based approach for assessment of the expected fitness as described in Algorithm 7.4. The experiments are performed on the sphere problem (see Appendix B.1), the Heaviside sphere problem (see Appendix B.2), and on Branke's multipeak problem (see Appendix B.6). For building the Kriging metamodels, a sample size of $n_{\mathrm{krig}} = 2n = 20$ is used, and for the approximation for the expected fitness, a $\mathrm{MEM}50^+_{\mathrm{LHS}}$ approach is followed. Each run uses a budget of $10,000$ function evaluations.

Figure 7.20 and Figure 7.21 show the results of Experiment 7.2.9. In Figure 7.20 the performance is shown in terms of median fitness (a posteriori approximated using Monte-Carlo

Technical Note 7.5: General Framework of a Metamodel Assisted Evolutionary Algorithm for Finding Robust Optima

initialize parent population
initialize archive
while not terminate **do**
  generate offspring
  for each offspring **do**
    (**optional: evaluate** offspring and **add** to archive)
    select archive points for metamodel construction
    if no representative set of samples available **then**
      get extra sample points
      evaluate the extra sample points
      add the extra sample points to the archive
      construct a local metamodel for the current individual
    end if
  end for
  for each offspring **do**
    evaluate robust fitness using the (ensemble of) local metamodel(s)
  end for
  select best offspring as new parent population
  (**optional: clean up** archive)
end while

Algorithm 7.4: Kriging Based Evolutionary Algorithm for Finding Robust Optima

Procedure parameters: number of samples used for metamodel construction n_{krig}, number of samples used for estimating the effective fitness m

Procedure variables: solution archive A

1: $\mathbf{P} \leftarrow$ *generate_initial_population*()
2: $A \leftarrow \emptyset$
3: **while** not terminate **do**
4: $\quad\mathbf{O} \leftarrow$ *generate_offspring*($\mathbf{P}$)
5: $\quad$**for each** $\mathbf{x}_{\mathrm{o}} \in \mathbf{O}$ **do**
6: $\quad\quad\mathbf{X}_{\mathrm{ref}} \leftarrow$ *latin_hypercube_sampling*($\mathbf{x}_{\mathrm{o}}, \boldsymbol{\sigma}_{\epsilon}, n_{\mathrm{krig}}$)
7: $\quad\quad(\mathbf{A}_{\mathrm{sel}}, \mathbf{X}_{\mathrm{cand}}) \leftarrow$ *reference_set_based_archive_selection*($\mathbf{X}_{\mathrm{ref}}, \mathbf{A}$)
8: $\quad\quad$**for a random** $\mathbf{x}_{\mathrm{c}} \in \mathbf{X}_{\mathrm{cand}}$ **do**
9: $\quad\quad\quad f_{\mathrm{c}} \leftarrow$ *evaluate*($\mathbf{x}_{\mathrm{c}}$)
10: $\quad\quad\quad A \leftarrow A \cup \{(\mathbf{x}_{\mathrm{c}}, f_{\mathrm{c}})\}$
11: $\quad\quad\quad A_{\mathrm{sel}} \leftarrow A_{\mathrm{sel}} \cup \{(\mathbf{x}_{\mathrm{c}}, f_{\mathrm{c}})\}$
12: $\quad\quad$**end for**
13: $\quad\quad\hat{f}_{\mathbf{x}_{\mathrm{o}}}(\mathbf{x}) \leftarrow$ *calibrate_kriging*($\mathbf{A}_{\mathrm{sel}}$)
14: $\quad\quad\tilde{f}_{\mathbf{x}_{\mathrm{o}}} \leftarrow \hat{f}_{\mathrm{eff}}(\hat{f}(\mathbf{x})_{\mathbf{x}_{\mathrm{o}}}, \mathbf{x}_{\mathrm{o}})$
15: $\quad$**end for**
16: $\quad\mathbf{P} \leftarrow$ *select*($\mathbf{O}$)
17: **end while**

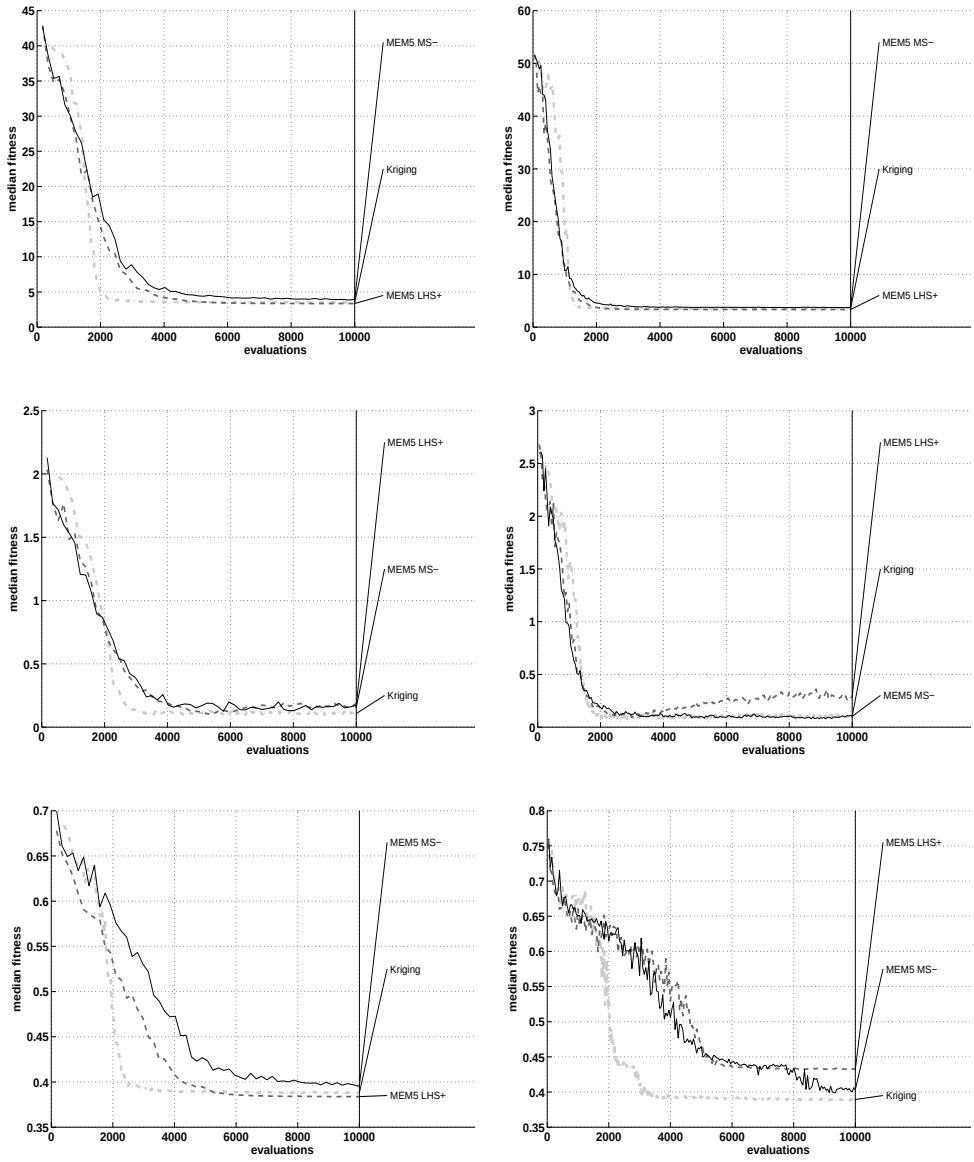

Figure 7.20: Results of Experiment 7.2.9. The performance, in terms of median fitness (a posteriori approximated using Monte-Carlo integration with 100 samples) of the Kriging based evaluation scheme incorporated into the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES, compared against two variants of the MEM evaluation schemes. Top row: results on the sphere. Middle row: results on the Heaviside sphere. Bottom row: the results on Branke's multipeak problem. Left column: the $(5/2_{DI}, 35)$-σSA-ES. Right column: the CMA-ES. The results are obtained using 50 runs for each scheme.

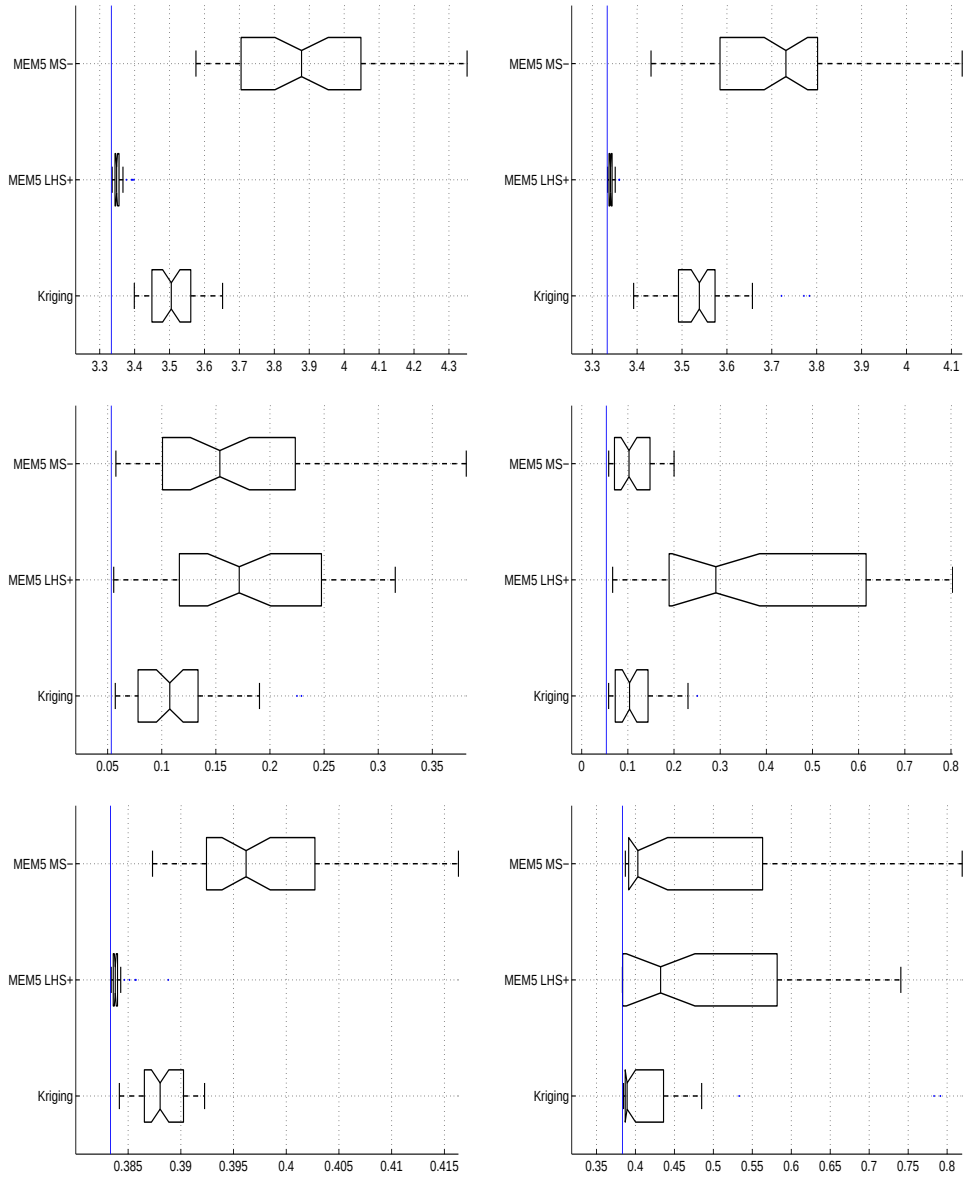

Figure 7.21: Results of Experiment 7.2.8. The final solution quality of the Kriging based evaluation scheme incorporated into the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES, compared against two variants of the MEM evaluation schemes. The solution quality is approximated a posteriori using Monte-Carlo sampling with 1000 samples. Top row: the results on the sphere problem. Middle row: the results on the Heaviside sphere problem. Bottom row: the results on Branke's multipeak problem. Left column: the $(5/2_{DI}, 35)$-σSA-ES. Right column: the CMA-ES. The results are obtained using 50 runs for each scheme.

integration with 100 samples) and in Figure 7.21 the final solution quality (approximated a posteriori using Monte-Carlo sampling with 1000 samples) is displayed. From the figures it can be seen that we can draw similar conclusions as for the ABRSS approach of the previous section. The methods incorporating the Kriging scheme outperforms the $\mathrm{MEM}^{-}_{\mathrm{MS}}$ scheme on all runs except for one where it is comparable. However, comparing it to the $\mathrm{MEM}^{+}_{\mathrm{LHS}}$ scheme, it is outperformed on the sphere problem and on Branke's multipeak problem, but is clearly better on the Heaviside sphere. The convergence speed obtained by using the Kriging based scheme (see Figure 7.21) is in the early stages a bit slower which is due to the fact that the archive needs to fill. Once the archive is filled, the Kriging based approach quickly zooms in and catches up with the MEM schemes.

In conclusion, we note that metamodeling can be successfully employed within Evolutionary Algorithms for finding robust optima. As the cost of constructing metamodels is high, such methods are only applicable if the evaluation cost of the original objective function is higher than the cost of constructing the metamodel. When this is the case, performing sampling on the metamodels is a promising way to get estimate for the effective fitness. A downside of metamodeling based approaches is that when it is too expensive to use all archive points for constructing the metamodels (e.g., as in Kriging), a subset selection should be made. This introduces an algorithm parameter that highly affects the accuracy of the metamodels. That is, when using a limited number of samples for metamodel construction, the metamodel itself is limited with respect to the prediction accuracy. In this sense, modeling approaches that use iterative updates are favorable, as those should be able to use the complete archive.

Additionally, an issue worthwhile mentioning is the possibility of performing exact robustness analysis on the metamodels instead of using sampling methods. That is, the metamodels themselves are not black-box functions, but have an explicit mathematical description. For some of these techniques, explicit derivation of the effective objective function measures might be possible. Poles and Lovison [PL09] already present such an approach in the context of polynomial models.

7.2.9 Niching for Finding Robust Optima

All techniques that have been discussed up to now have been noted to work well with respect to zooming in on the robust optimizer in case it is shifted, but do not noticeably improve the capabilities of Evolution Strategies to target the more robust peaks. This observation has been made, for instance, in [KEB10] in the context of archiving and in [KRD+11] in the context of adaptive averaging. Niching techniques are designed to improve the explorative behavior of Evolutionary Algorithms by actively separating sub-populations in different parts of the search space. For finding robust optima, such techniques could also be incorporated, with the specific goal in mind to find the more robust parts of the search space. Preliminary results, providing a proof of concept for this idea, are presented by Tsutsui and Ghosh [TG97]. In this study,

the sharing scheme of Goldberg and Richardson [GR87] is used in combination with a SEM evaluation approach. However, despite promising results, no further studies exist that include niching techniques for finding robust optima.

A straightforward incorporation of niching into a scheme for finding robust optima is, for instance, to use a SEM/MEM evaluation scheme and include a basic niching approach, e.g., as proposed by Shir et al. [SB09]. Algorithm 7.5 describes the general framework of this niching approach, in which the evaluation procedure should simply be a SEM/MEM approach when aiming to find robust optima.

In this framework, q niches are maintained, which are in the simplest case q parent individuals which are separated in the search space (i.e., the niche leaders or alpha individuals). Every generation λ offspring are generated for each niche separately. After that, the full population of offspring is evaluated, and based on the fitness, the dynamic peak set, which is the set of new niche leaders for the next generation, is selected by means of the dynamic peak identification procedure (see Algorithm 7.6). Each individual in the DPS forms a separate niche and inherits the strategy parameters from its parent, which are updated according to the normal update rules. Finally, if there are fewer than q niches, new niches are created randomly.

The dynamic peak identification algorithm, as described in Algorithm 7.6 works by sorting the individuals by fitness, and then considering them in that order (i.e., fitter individuals are treated earlier). The best individual is always a niche leader and will always form a niche. The other individuals will form a niche only if they are not within a radius ρ of an already formed niche and if there are not yet q niches. This way, a set of at most q individuals is selected, which are the niche leaders for the next generation.

Regarding the two important parameters ρ and q: the number of desired niches is a design choice and the niche radius is recommended to be set relative to the number of niches,

$$\rho = \frac{\sqrt{\sum_{i=1}^{n}((\mathbf{x}_\mathrm{u})_i - (\mathbf{x}_\mathrm{l})_i)^2}}{2\sqrt[n]{q}}. \tag{7.36}$$

Although the approach sounds reasonable, a simple experiment on a two-dimensional instance of Branke's multipeak problem shows that it is more complicated than this:

Experiment 7.2.10 (Dynamics of niching for finding robust optima)**:** We consider four two-dimensional instance of Branke's multipeak problem (see Appendix B.6) in which the anticipated input uncertainties vary. The noise levels are: $\boldsymbol{\delta} = \mathbf{0}$ (no noise), $\boldsymbol{\delta} \sim \mathcal{U}(-\mathbf{0.05}, \mathbf{0.05})$ (low noise), $\boldsymbol{\delta} \sim \mathcal{U}(-\mathbf{0.1}, \mathbf{0.1})$ (medium noise), $\boldsymbol{\delta} \sim \mathcal{U}(-\mathbf{0.5}, \mathbf{0.5})$ (default noise). On these problems we run a normal CMA-ES (see Section 4.2.3) incorporating a SEM^- evaluation scheme and a niching based CMA-ES, which uses $q = 4$ niches, one parent per niche (i.e., $\mu = 1$), $\lambda = 10$ offspring per niche, and also the SEM^- evaluation scheme. For both schemes one run is performed on each problem instance.

Figure 7.22 shows the results of Experiment 7.2.10 in terms of distance to the robust optimizer

Algorithm 7.5: Niching Based Evolutionary Algorithm for Finding Robust Optima

Procedure parameters: Number of niches q, niching radius ρ

1: **initialize** parent population
2: **while** not terminate **do**
3: **for** each niche **do**
4: **generate** λ offspring from the current niche
5: **end for**
6: **evaluate** offspring
7: **compute** the Dynamic Peak Set (DPS) of the population
8: **for each** individual in the DPS **do**
9: **set** the current individual as niche
10: **inherit** the strategy parameters from the parent
11: **update** the strategy parameters of the current niche
12: **end for**
13: **if** $|\mathrm{DPS}| < q$ **then**
14: **generate** new niche from scratch
15: **end if**
16: **end while**

Algorithm 7.6: Dynamic Peak Identification (DPI)

Procedure parameters: Number of niches q, niching radius ρ

Sort the population according to fitness: $P = \{\mathbf{x}_{1:q\lambda}, \ldots, \mathbf{x}_{1:q\lambda}\}$
$i \leftarrow 1$
$numpeaks \leftarrow 0$
$DPS \leftarrow \emptyset$
while $numpeaks < q$ **and** $i \leq q \cdot \lambda$ **do**
  $is\_niche \leftarrow \text{true}$
  for each $\mathbf{x}_{\text{niche}} \in DPS$ **do**
    if $||\mathbf{x}_{\text{niche}} - \mathbf{x}_{i:q\lambda}|| < \rho$ **then**
      $is\_niche \leftarrow \text{false}$
      break
    end if
  end for
  if is_niche **then**
    $DPS \leftarrow DPS \cup \{\mathbf{x}_{i:q\lambda}\}$
    $numpeaks \leftarrow numpeaks + 1$
  end if
  $i \leftarrow i + 1$
end while
return DPS

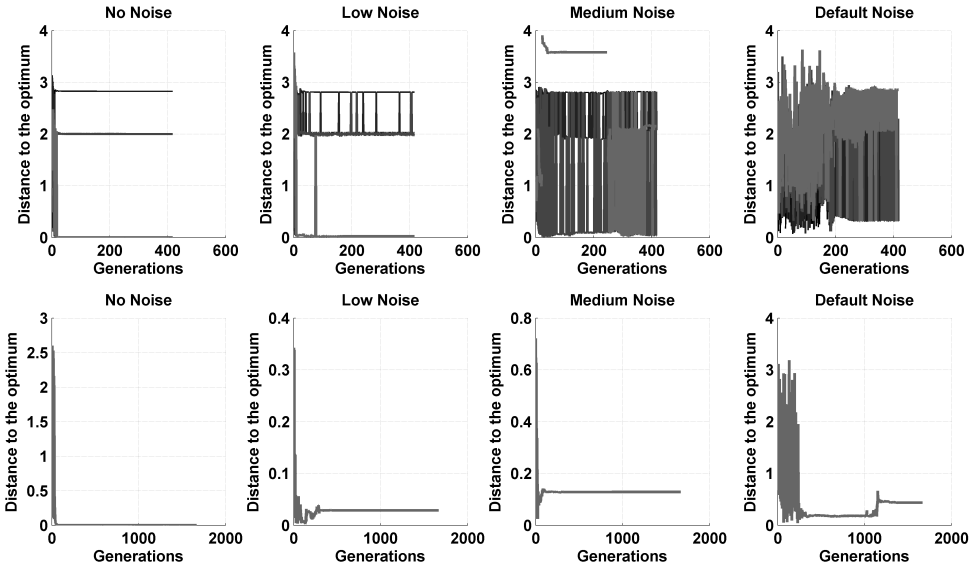

Figure 7.22: A single run of the niching approach incorporating a SEM evaluation scheme (top) and a normal MEM approach. The plot shows the results in terms of distance to the robust optimizer at $[-1]^n$.

at $[-1]^n$. From this plot it can be seen that when there is no noise, the niching approach easily settles in the four optima that exist in two dimensions. However, when increasing the anticipated input disturbances, the niching approach will encounter more and more difficulties for forming the niches. Hence, the niche formation process seems to become unstable in this particular scenario. An explanation for why this instability can arise is that the four peaks lie very close to each other. At times, neighboring niches might be destroyed when an individual in one niche with a randomly high fitness lies to close to the other niche. This way, niches can be destroyed and prevent each other to settle in a peak.

From the small preliminary study, we can conclude that although including niching techniques might be a good idea, using straightforward niching implementations in combination with SEM/MEM evaluation may cause undesirable instabilities. For the niching approach considered in this section it would be interesting to study its behavior first on noisy optimization problems and find out how it can be stabilized before including it in the context of finding robust optima. When this is achieved, an interesting idea with respect to the niche radius is to link the niche radius to the magnitude of the input uncertainty. That is, the input uncertainties provide an indicator that fits naturally for specifying the niche radius. The latter remains a parameter that depends on the problem at hands.

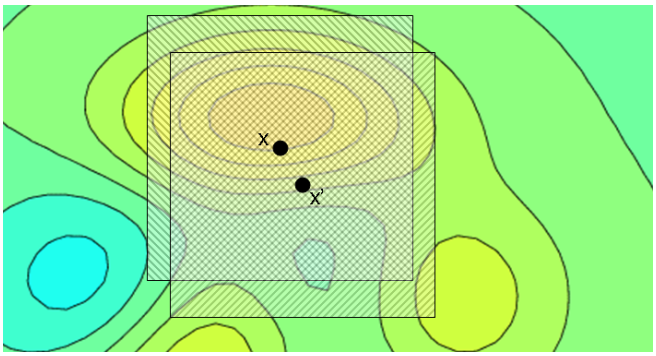

Figure 7.23: Overlap sketch: two solutions $\mathbf{x}$ and $\mathbf{x}'$ around which square regions are drawn indicating the regions of uncertainty $\eta_{\mathbf{x}}$ and $\eta_{\mathbf{x}'}$.

7.2.10 Exploiting Overlap

The approaches that have been discussed so far aim to obtain as accurate as possible fitness approximations using as few as possible objective function evaluations. In [KEDB10b], a different approach is followed.

An important observation when aiming to find robust optima is that sampling regions (i.e., η-neighborhoods) of different candidate solutions in (successive) populations are often overlapping, particularly in later stages of the optimization where the population focuses on a single region of the search space. Hence, the evaluations made in overlapping regions can be used at the same time for evaluating the robustness of different candidate solutions. Or, in some cases, it is even possible to discard a large part of the sampling space when comparing solutions. Consider, for example, the scenario displayed in Figure 7.23, where two solutions $\mathbf{x}$ and $\mathbf{x}'$ are compared given a uniform distribution of the input noise. Here, it suffices to sample the non-intersecting regions, because the contribution of the intersecting region to the expected fitness is the same for both solutions. Instead of trying to acquire an estimate of the effective fitness for each candidate solution separately, alternatively one could compare solutions based on how they relate in their effective fitness; therewith exploiting the overlap of their η-neighborhoods.

Consider the special case of comparing two candidate solutions $\mathbf{x}$ and $\mathbf{x}'$ on their expected fitness based on a uniform perturbation $\boldsymbol{\delta} \sim \mathcal{U}(-\mathbf{l}, \mathbf{l})$. By normalizing the search space we can transform the sampling intervals of the independent input variables such that they have the same width l. For this scenario (illustrated in Figure 7.24), the intersection region A of $\eta_{\mathbf{x}}$ and

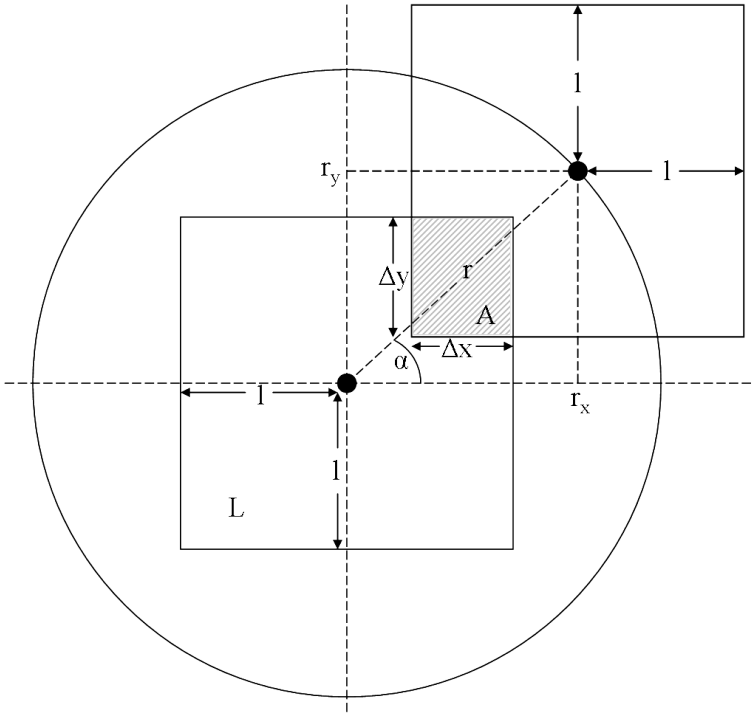

Figure 7.24: A two-dimensional sketch of the overlap of the regions of uncertainty $\eta_{\mathbf{x}}$ and $\eta_{\mathbf{x}'}$ of the two solutions $\mathbf{x}$ and $\mathbf{x}'$.

$\eta_{\mathbf{x}'}$ contributes the same to the expected fitness of both solutions, i.e.,

$$f_{\exp_{\mathbf{x}}} = \int_{\tilde{\mathbf{x}}\in\eta_{\mathbf{x}}\setminus A} f(\tilde{\mathbf{x}})\ \mathrm{p}_{\mathbf{x}}(\tilde{\mathbf{x}})\ d\tilde{\mathbf{x}} + \int_{\tilde{\mathbf{x}}\in A} f(\tilde{\mathbf{x}})\ \mathrm{p}_{\mathbf{x}}(\tilde{\mathbf{x}})\ d\tilde{\mathbf{x}}, \tag{7.37}$$

$$f_{\exp_{\mathbf{x}'}} = \int_{\tilde{\mathbf{x}}\in\eta_{\mathbf{x}'}\setminus A} f(\tilde{\mathbf{x}})\ \mathrm{p}_{\mathbf{x}'}(\tilde{\mathbf{x}})\ d\tilde{\mathbf{x}} + \int_{\tilde{\mathbf{x}}\in A} f(\tilde{\mathbf{x}})\ \mathrm{p}_{\mathbf{x}'}(\tilde{\mathbf{x}})\ d\tilde{\mathbf{x}}, \tag{7.38}$$

$$f_{\exp_{\mathbf{x}}} - f_{\exp_{\mathbf{x}'}} = \int_{\tilde{\mathbf{x}}\in\eta_{\mathbf{x}}\setminus A} f(\tilde{\mathbf{x}})\ \mathrm{p}_{\mathbf{x}}(\tilde{\mathbf{x}})\ d\tilde{\mathbf{x}} - \int_{\tilde{\mathbf{x}}\in\eta_{\mathbf{x}'}\setminus A} f(\tilde{\mathbf{x}})\ \mathrm{p}_{\mathbf{x}'}(\tilde{\mathbf{x}})\ d\tilde{\mathbf{x}}, \tag{7.39}$$

where $\mathrm{p}_{\mathbf{x}}(\tilde{\mathbf{x}}) \sim \mathrm{pdf}(\mathbf{x}+\boldsymbol{\delta})$ and $\mathrm{p}_{\mathbf{x}'}(\tilde{\mathbf{x}}) \sim \mathrm{pdf}(\mathbf{x}'+\boldsymbol{\delta})$. Hence, for comparisons, A offers no relevant information. Moreover, when $\mathbf{x}$ and $\mathbf{x}'$ are located close to each other, A will be large compared to $\eta_{\mathbf{x}}\setminus A$ and $\eta_{\mathbf{x}'}\setminus A$ and for an evaluation method based on sampling in $\eta_{\mathbf{x}}$ and $\eta_{\mathbf{x}'}$, the probability of sampling within $\eta_{\mathbf{x}}\setminus A$ and $\eta_{\mathbf{x}'}\setminus A$ will be small. Given the surface area (or volume) $|A|$ of A, the probability that one uniformly drawn random sample in $\eta_{\mathbf{x}}$ hits $\eta_{\mathbf{x}}\setminus A$ is

$$P(\text{sample not in } A \mid \text{sample in } \eta_{\mathbf{x}}) = 1 - \frac{|A|}{|\eta_{\mathbf{x}}|} = 1 - \frac{|A|}{2^n l^n}. \tag{7.40}$$

Let X denote the discrete random variable for the number of samples n until the first sample in $\eta_{\mathbf{x}}\setminus A$ is obtained. For a pure Monte-Carlo sampling scheme, this leads to the following

expected number of samples needed to obtain one sample in $\eta_{\mathbf{x}}$ that is not in A

$$E(X) = \sum_{n=1}^{\infty} n \cdot P(X = n) = \frac{1}{1 - \frac{|A|}{\eta_{\mathbf{x}}}} = \frac{2^n l^n}{2^n l^n - |A|}. \tag{7.41}$$

Obviously, the same reasoning can be followed for $\eta_{\mathbf{x}'}$. Provided that there is an overlap in the regions of uncertainty, $|A|$ can be computed as

$$|A| = \prod_{i=1}^{n} (2l - |x_i - x_i'|)\,. \tag{7.42}$$

Or, alternatively, in two dimensions the following expression can be derived for $|A|$, given that $\mathbf{x}$ and $\mathbf{x}'$ lie at a distance r from each other and make an angle α with respect to the x-axis

$$|A| \quad = \quad \begin{cases} (2l - r_x) \cdot (2l - r_y) & \text{, if } r_x \leq 2l \text{ and } r_y \leq 2l \\ 0 & \text{, otherwise} \end{cases}. \tag{7.43}$$

Here, $r_x = |r \cos \alpha|$ and $r_y = |r \sin \alpha|$. From this, and assuming a periodicity of $\pi/2$ for α (i.e., $\alpha \in [0, \pi/2]$), an expression can be derived for the expected number of samples in $\eta_{\mathbf{x}}$, m, needed in order to hit the area $\eta_{\mathbf{x}} \backslash A$ at least c times

$$m \quad = \quad \begin{cases} 4cl^2 / \left(2lr\left(\cos\alpha + \sin\alpha\right) - r^2 \cos\alpha \sin\alpha\right) & \text{, if } r_x \leq 2l \text{ and } r_y \leq 2l \\ c & \text{, otherwise} \end{cases} \tag{7.44}$$

By expressing r in terms of l, substituting $r = kl$, this can be simplified to

$$m \quad = \quad \begin{cases} 4c / \left(2k\left(\cos\alpha + \sin\alpha\right) - k^2 \cos\alpha \sin\alpha\right) & \text{, } k\cos\alpha \leq 2 \text{ and } k \sin\alpha \leq 2 \\ c & \text{, otherwise} \end{cases} \tag{7.45}$$

The derivation above is still dependent on the angle α between $\mathbf{x}$ and $\mathbf{x}'$. It is desirable to have an approximation independent of α. This yields a general approximation for the required number of samples m needed for two individuals at a distance r (still using $r = lk$, i.e., $k = r/l$) to have at least c samples in the regions $\eta_{\mathbf{x}}$ and $\eta_{\mathbf{x}'}$ respectively. For this, we look at the upper bound of $|A|$ with respect to α and note that for $|A|$ to be maximized, $\alpha = 0$ or $\alpha = \pi/4$

$$\begin{aligned} |A| \quad &= \quad \begin{cases} \max\{4l^2 - 2kl^2, 4l^2 - 2\sqrt{2}kl^2 + \frac{1}{2}k^2l^2\} & \text{, if } k \leq 2\sqrt{2} \\ 0 & \text{, otherwise} \end{cases} \\ &= \quad \begin{cases} 4l^2 - 2\sqrt{2}kl^2 + \frac{1}{2}k^2l^2 & \text{, if } k \leq 4\left(\sqrt{2} - 1\right) \\ 4l^2 - 2kl^2 & \text{, if } 4\left(\sqrt{2} - 1\right) < k \leq 2\sqrt{2}\,. \\ 0 & \text{, otherwise} \end{cases} \end{aligned} \tag{7.46}$$

This upper bound of A can be used to approximate (the upper bound of) m

$$m = \begin{cases} 8c/\left(4\sqrt{2}k - k^2\right) & \text{, if } k \leq 4\left(\sqrt{2}-1\right) \\ 2c/k & \text{, if } 4\left(\sqrt{2}-1\right) < k \leq 2\sqrt{2} \\ 0 & \text{, otherwise} \end{cases}. \tag{7.47}$$

For general n-dimensional cases, $|A|$ can be assumed to be maximized in the cases equivalent to the two-dimensional cases of $\alpha = 0$ and $\alpha = \pi/4$. Following this, and again using the substitution $r = kl$, we obtain

$$|A| = \begin{cases} 2^n l^n \max\left\{\left(1 - \frac{1}{2}k\right), \left(1 - \frac{1}{2\sqrt{n}}k\right)^n\right\} & \text{, if } k \leq 2\sqrt{n} \\ 0 & \text{, otherwise} \end{cases}, \tag{7.48}$$

which can be used to obtain an expression for the expected number of samples in $\eta_{\mathbf{x}}$ needed to hit $\eta_{\mathbf{x}} \backslash A$ at least c times

$$m = \begin{cases} c \max\left\{\frac{2}{k}, 1/\left(1 - \left(1 - \frac{1}{2\sqrt{n}}k\right)^n\right)\right\} & \text{, if } k \leq 2\sqrt{n} \\ c & \text{, otherwise} \end{cases}. \tag{7.49}$$

It is clear that when using normal sampling approaches, many samples are practically wasted when the distance between two solutions becomes small. Figure 7.25 shows the number of required samples m needed to hit $\eta_{\mathbf{x}} \backslash A$ at least once versus k for $n = 10^0, 10^1, \ldots, 10^6$. Interestingly, the plots are not much different for all values of n. For $k \lesssim 1.7$ the term $2/k$ becomes the determining factor and the other term leads to $m \approx 1$, even for $n = 10^6$. Hence, the following rule-of-thumb can be used to indicate the growth of m relative to k

$$m = \begin{cases} \frac{2c}{k} & \text{, if } k \leq 1.7 \\ c & \text{, otherwise} \end{cases}. \tag{7.50}$$

In [KEDB10b], a rejection based sampling procedure is proposed to obtain m samples in $\eta_{\mathbf{x}} \backslash \eta_{\mathbf{x}'}$. This approach is described in Algorithm 7.7 and simply works by uniformly sampling in $\eta_{\mathbf{x}}$ and rejecting samples in $\eta_{\mathbf{x}} \cap \eta_{\mathbf{x}'}$. Although the practical viability of this scheme is limited, it can be incorporated into a simple (1+1)-Evolution Strategy to test the approach of exploiting overlap. Additionally, besides the fact that we can either avoid the region A in case of uniform noise, or at least reuse the samples for the evaluation of $\mathbf{x}$ and $\mathbf{x}'$ in case of other noise distributions, there is also a symmetry in the regions $\eta_{\mathbf{x}} \backslash A$ and $\eta_{\mathbf{x}'} \backslash A$. Given a perturbation $\mathbf{x}_\mathrm{p} \sim U(-\mathbf{l}, \mathbf{l})$ we note that

$$\mathbf{x} + \mathbf{x}_\mathrm{p} \in \eta_{\mathbf{x}} \backslash A \Leftrightarrow \mathbf{x}' - \mathbf{x}_\mathrm{p} \in \eta_{\mathbf{x}'} \backslash A. \tag{7.51}$$

Hence, when using the rejection based sampling algorithm of Algorithm 7.7, it is only required to obtain one set of samples for every pair $\mathbf{x}$ and $\mathbf{x}'$, because $\mathbf{X}'$ can be deduced from $\mathbf{X}$.

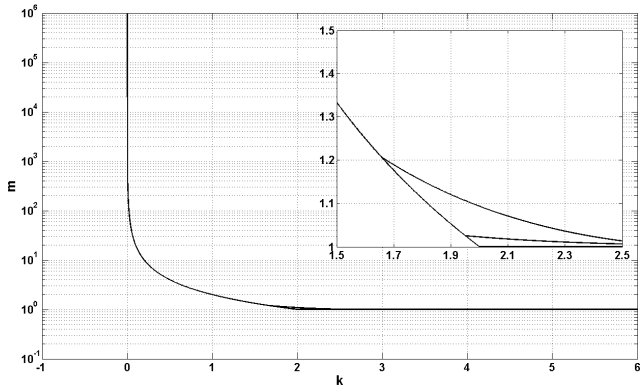

Figure 7.25: The number of samples needed to obtain at least 1 sample in the region $\eta_{\mathbf{x}} \backslash A$, plotted against the normalized distance k between $\mathbf{x}$ and $\mathbf{x}'$ for $n = 10^0, 10^1, \ldots, 10^6$ with a zoom on the interval $k \in [1.5, 2.5]$.

Algorithm 7.7: Rejection Based Uniform Sampling in $\eta_{\mathbf{x}} \backslash \eta_{\mathbf{x}'}$

input: η-neighborhoods $\eta_{\mathbf{x}}$ and $\eta_{\mathbf{x}'}$

output: a set $\mathbf{X}$ of m uniformly drawn samples in $\eta_{\mathbf{x}} \backslash \eta_{\mathbf{x}'}$

1: $\mathbf{X} \leftarrow \emptyset$
2: **while** $|\mathbf{X}| < m$ **do**
3: $\quad \mathbf{x}_{\mathrm{s}} \sim U(\eta_{\mathbf{x}})$
4: $\quad$ **if** $\mathbf{x}_{\mathrm{s}} \notin \eta_{\mathbf{x}'}$ **then**
5: $\quad\quad \mathbf{X} \leftarrow \mathbf{X} \cup \{\mathbf{x}_{\mathrm{s}}\}$
6: $\quad$ **end if**
7: **end while**
8: **return** $\mathbf{X}$

As a proof of concept, in [KRD+11], two versions of the (1+1)-ES are compared on the sphere problem (see Appendix B.1): one incorporating a $\mathrm{MEM}^{+}_{\mathrm{MS}}$ scheme (and reevaluation of the parent) and one using the rejection based sampling approach for comparing the parent and the offspring. In both schemes the sample size was set to $m = 2n = 20$ and an evaluation budget of $2 \cdot 10^6$ was used. The results of one run of both schemes are shown in Figure 7.26.

From the convergence plots (the top left plot for the normal sampling approach and the top

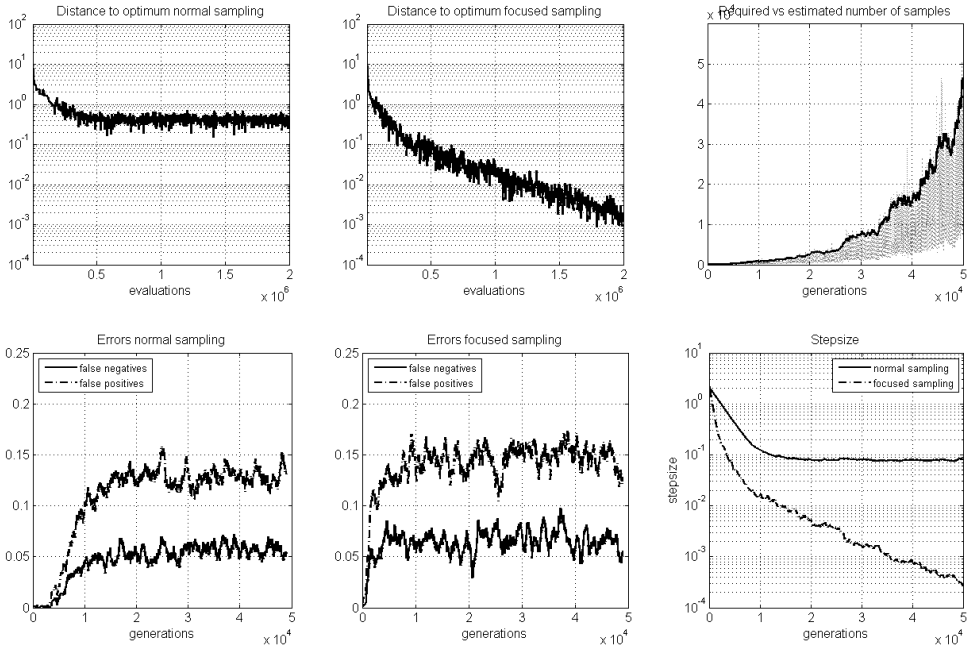

Figure 7.26: Left column: distance to the optimum and ordering error frequency of the normal sampling approach. Middle column: distance to the optimum and ordering error frequency of the focused sampling approach. Top right: the required number of samples and an upper bound estimator for the required number of samples of the focused sampling run. Bottom right: the stepsize development for both approaches.

right plot for the rejection based sampling approach) it can be seen that while the normal sampling stagnates after about $5 \cdot 10^5$ evaluations, there is no sign of stagnation for the approach implementing the focused sampling. Hence, although it still uses $m = 20$ samples for each evaluation, the rejection based sampling approach remains making progress. Supported also by the stepsize plots (bottom right), these empirical results suggest linear generation- and evaluation-wise convergence for this particular problem.

The error frequency plots (bottom left for the normal sampling approach and bottom right for the rejection based sampling approach) show the frequency of false negatives and false positives versus the number of generations (these frequencies are computed over a window of 1000 generations). For both approaches, the error frequencies stay at the same levels at a certain point in time. For the normal sampling approach, this can be related to the stagnation of the stepsize (i.e., the error rate is coupled to the noise ratio, which is directly coupled to the stepsize). However, for the rejection based sampling approach the error rates stay at the same levels even with the stepsize decreasing.

The number of samples that required in order to obtain m samples in $\eta_{\mathbf{x}} \backslash A$ is shown in the top right plot. Note that the implementation used for these experiments uses the simple

rejection method of Algorithm 7.7. The thick solid line shows for every generation the expected upper bound of the number of samples, computed using the simple rule-of-thumb of Eq. 7.50, with $r = \sigma\sqrt{n}$. This plot shows how this rule-of-thumb accurately determines the upper bound for the required number of samples, but also that in many cases fewer samples suffice.

In conclusion, the ideas proposed in [KEDB10b] provide an alternative view on robustness evaluation where the focus is not on trying to obtain accurate robustness estimates for individual candidate solutions, but rather to compare the solutions of (successive) populations with respect to robustness.

The experiments on the $(1{+}1)$-ES show how the proposed idea can successfully be integrated in common optimization algorithms. However, a step is still to be made to also include this concept in population based schemes that work with sets of candidate solutions rather than two. An implementation for tournament selection can easily be derived, but for $(\mu/\rho\overset{+}{,}\lambda)$-selection, more sophisticated schemes are required. For $(\mu/\rho\overset{+}{,}\lambda)$-schemes, samples of overlapping regions can at least be reused when evaluating candidate solutions, but specifically targeting non-overlapping regions will become computationally more expensive. An idea that requires further development is to replace the rejection based sampling by Gibbs sampling [CG92, GW92]. This would decrease the computational complexity of the sampling method tremendously, making this approach from that perspective viable.

Finally, two issues that should still be addressed are: 1) the way in which this sampling strategy can be adopted in cases of other input noise distributions, and 2) the applicability of this method in case there are disturbance-free design variables. In the former case, the overlapping region may still be relevant for comparing two candidate solutions. In the latter case, there will never be an overlap, hence, in principle this method cannot be applied.

7.3 Summary and Discussion

This chapter has discussed the problem of finding robust optima. It has been shown how these types of problems may be modeled within the optimization problem statement and how different robustness measures can be derived from this. The problem of finding robust optima is therefore reformulated into the problem of optimization of the effective objective functions, while satisfying the effective constraints.

The effective objective and constraint functions can be seen as transformations of the original objective and constraint functions. However, as precise evaluation of these functions is often impossible, approximation methods are required. The algorithmic design aspect is therewith reduced to efficient approximation or comparison of candidate solutions with respect to the effective objective and constraint functions.

In the second part of this chapter, techniques have been reviewed that can be used within Evolution Strategies to find robust optima. The particular focus of this chapter was on single objective optimization of the expected objective function.

The myopic approach (Section 7.2.1) is the strategy which simply uses canonical instances of the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES. This view relies on the observation that Evolutionary Algorithms in practice already tend to converge to the more robust peaks. It has been shown to be quite effective for identifying the more robust peaks, but it fails in case of shifted robust optimizers.

When actively accounting for robustness, using Monte-Carlo integration methods is the most straightforward way to evaluate the effective fitness of candidate solutions. Compared to myopic approaches, these evaluation approaches (named SEM when using only one sample and MEM when using multiple samples) allow for a closer convergence to shifted robust optimizers. Additionally, for MEM evaluation approaches one could use the same sample perturbations for all individuals in the population and base the Monte-Carlo integration on a sample set obtained with Latin Hypercube sampling. These modifications seem to be beneficial when the region around the robust optimum is symmetric, but can yield divergent behavior when sharp ridges cause the robust optimizer to be shifted. A brief comparison of MEM evaluation approaches with m samples to the alternative of using SEM evaluation and increasing the population size with the same rate m (i.e., implicit averaging) shows that for finding robust optima, explicit resampling seems to work better.

Similar to noisy objective functions, adaptive averaging techniques can also be used in the context of finding robust optima. These techniques provide a way of omitting the problem of determining an appropriate sample size m and in theory allow Evolution Strategies to zoom in on the optimizer with arbitrary precision, although in practice convergence can be slow.

For objective functions for which evaluations are expensive, archive based approaches and metamodeling approaches can be used for finding robust optima. These techniques aim to efficiently use an archive of previous evaluations to serve as samples for robustness evaluation. Although the overhead cost of these methods is considerable, these techniques provide promising results compared to standard MEM approaches. A difficulty is, however, that especially in higher dimensional search spaces it takes a while before the archive is filled and can effectively be used for robustness estimation or metamodeling.

When aiming to actively improve the capabilities of targeting the more robust peaks, it has been suggested to use niching techniques. However, although this idea sounds promising, in this chapter it has been observed that a straightforward implementation of niching with a MEM evaluation approach tends to become unstable. In this context it should be accounted for that objective function evaluations are noisy, hence, the niching technique should be robust against noise. The niching technique considered in Section 7.2.9 is, without modifications, not suitable for this.

Lastly, an alternative evaluation scheme has been considered in which it is suggested to compare pairs of individuals with respect to their robustness rather than trying to obtain accurate robustness estimates. By exploiting the overlap in the regions of uncertainty (or η-neighborhoods) it is in some cases possible to drastically improve the convergence accuracy

of MEM evaluation approaches (Section 7.2.10). However, this approach has been formulated so far only for simple $(1+1)$-schemes and requires further study to be applied in practical scenarios.

The techniques for finding robust optima that are discussed in this chapter represent the main classes of approaches that can be followed. Based on this review, the myopic approach, the $\mathrm{MEM}^{+}_{\mathrm{LHS}}$ and $\mathrm{MEM}^{-}_{\mathrm{MS}}$, the adaptive averaging approach, the archive based approach, and the metamodeling approach are feasible approaches for practical scenarios. The question that remains is which of these techniques can best be used in practice. In Chapter 8, we will study this question.