Evolution strategies for robust optimization

Kruisselbrink, J.W.

Chapter 5

Optimization of Noisy Objective Functions

Optimizing systems or models of systems that exhibit noisy output is a common scenario for real-world optimization problems. Figure 5.1 illustrates such a scenario schematically, in which the system or (simulation) model produces an output that is noisy and where this noise in the output propagates to the objective and constraint functions. This chapter focuses on a restricted subclass of such problems, being unconstrained single objective real-parameter optimization problems in which the noise in the output propagates as additive noise in the objective function.

The intent of this chapter is to answer the following questions: 1) What is the goal of optimization when having noisy objective functions? 2) What is the effect of noise in the objective functions on Evolution Strategies? 3) How should Evolution Strategies, and in particular the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES, be adapted in order to deal with noisy objective functions?

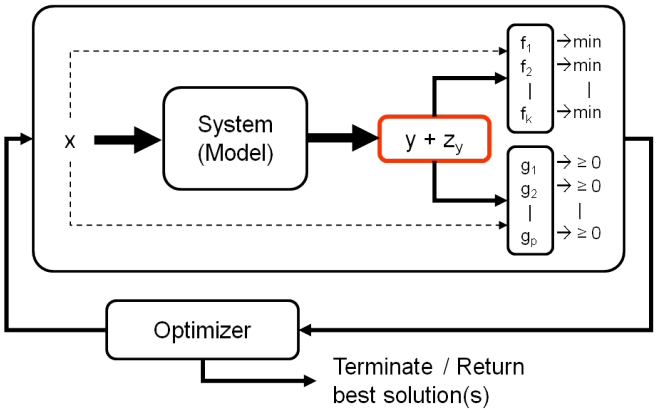

Figure 5.1: A typical robust optimization scenario: the system or model of the system for which an optimization problem needs to be solved produces a noisy output.

This chapter consists of two parts. In the first part (Section 5.1 and Section 5.2), the problem of noisy optimization and the effects of noise on Evolution Strategies are studied. Section 5.1 starts by providing a description of noisy objective functions and the goals of optimization in case of noisy objective functions. Section 5.2 studies the effects of noise in the objective functions on Evolution Strategies. In the second part of this chapter (Section 5.3 to 5.6), a number of noise handling techniques usable for Evolution Strategies are described and evaluated. Section 5.3 reviews some basic noise handling techniques, Section 5.4 reviews techniques known as adaptive averaging techniques, Section 5.5 briefly summarizes metamodel based noise handling techniques, and Section 5.6 provides a general discussion on noise handling techniques. Section 5.7 closes with a summary and discussion.

5.1 Noisy Objective Functions

The optimization problems considered in this chapter are unconstrained single-objective real-parameter optimization problems, with objective functions of the form

$$\tilde{f}(\mathbf{x}) = f(\mathbf{x}) + z(\mathbf{x}). \tag{5.1}$$

That is, the objective function $\tilde{f}(\mathbf{x})$ consists of a deterministic, noise-free part $f(\mathbf{x})$ and an additive stochastic part $z(\mathbf{x})$, which is a random variable indexed by space (i.e., it can be seen as a *random field* or *noise landscape*). In case of a *stationary* distribution, $z(\mathbf{x})$ are identically distributed for all $\mathbf{x} \in \mathbb{R}^n$, otherwise the noise is said to be *non-stationary*. Furthermore, the noise is *unbiased* if $\mathbf{E}[z(\mathbf{x})] = 0$ for all $\mathbf{x} \in \mathbb{R}^n$.

A common goal of optimization of noisy objective functions is to find optimal solutions for the deterministic part of the function $\tilde{f}(\mathbf{x})$. Hence, the underlying deterministic function $f(\mathbf{x})$ is considered to be the "true" objective function and the aim is to find optimal solutions for that underlying function despite the noisy evaluations. This view is considered explicitly in, e.g., [HB94, HNGK09], and is appropriate when the noise is due to measurement errors instead of being intrinsic to the system. This goal of optimization can be stated explicitly by denoting the objective function that is effectively seen as the objective function for optimization. This is called the *effective objective function* (denoted f_{eff}) which in this case is simply stated as

$$f_{\text{eff}}(\mathbf{x}) = f(\mathbf{x}). \tag{5.2}$$

An alternative goal, stated for example by Jin and Branke [JB05], is to find optimizers for the *expected objective function* (denoted f_{exp}), i.e.,

$$f_{\text{eff}}(\mathbf{x}) = f_{\text{exp}}(\mathbf{x}) = \mathbf{E}[\tilde{f}(\mathbf{x})]. \tag{5.3}$$

This aim is appropriate for systems with intrinsic noise. Although these two effective objective functions look very similar, it should be noted that these are only equivalent when the noise is unbiased.

Other goals for optimization of noisy objective functions are:

1. To optimize based on percentiles or the median.

2. To optimize based on a lower confidence bound, e.g.,

$$f_{\text{eff}}(\mathbf{x}) = \inf \left\{ a \in \mathbb{R} \mid P\left(\tilde{f}(\mathbf{x}) < a)\right) > p_\alpha \right\} \to \min, \tag{5.4}$$

   using an appropriate setting for the conflict level p_α, or

$$f_{\text{eff}}(\mathbf{x}) = \mathbf{E}[\tilde{f}(\mathbf{x})] - \omega\sqrt{\text{Var}[\tilde{f}(\mathbf{x})]}, \tag{5.5}$$

   using an appropriate weight ω.

3. To restate the optimization problem as a multi-objective problem, requiring optimization of the mean and minimization of the variance, i.e.,

$$\begin{aligned} {f_{\text{eff}}}^{(1)}(\mathbf{x}) &= \mathbf{E}[\tilde{f}(\mathbf{x})], & (5.6)\\ {f_{\text{eff}}}^{(2)}(\mathbf{x}) &= \text{Var}[\tilde{f}(\mathbf{x})] \to \min . & (5.7) \end{aligned}$$

4. To restate the optimization problem as a multi-objective problem, requiring optimization of the mean and optimization of the lower confidence bound, e.g.,

$$\begin{aligned} {f_{\text{eff}}}^{(1)}(\mathbf{x}) &= \mathbf{E}[\tilde{f}(\mathbf{x})], & (5.8)\\ {f_{\text{eff}}}^{(2)}(\mathbf{x}) &= P\left(\tilde{f}(\mathbf{x}) < T_{\text{crit}}\right) \to \min, & (5.9) \end{aligned}$$

   with T_{crit} being some critical level.

As pointed out by Sano and Kita [SK00], the latter two goals could be appropriate for optimization of investment, aiming to achieve high return and low risk solutions. However, note that for stationary noise, these alternatives do not effectively change the optimization goal as compared to the expected objective function. That is, in terms of optimization, these measures yield rankings amongst all solutions in the search space that are equivalent to the ranking based on the expected objective function.

The effective objective function states the optimization goal. However, it is obvious that it is impossible to precisely evaluate the effective objective functions stated above. Hence, for the evaluation of candidate solutions an alternative evaluation function $\hat{f}_{\text{eff}}(\mathbf{x})$ should be used that yields unbiased approximations of the effective objective functions. For instance, when using just one noisy evaluation for each candidate solution, one effectively uses $\hat{f}_{\text{eff}}(\mathbf{x}) = \tilde{f}(\mathbf{x})$, which yields unbiased approximations of the expected objective function. In literature, the step of explicitly stating an effective objective function is often omitted.

5.2 The Effects of Noise on Evolutionary Algorithms

The effects of noise on Evolution Strategies (and Evolutionary Algorithms in general) have been extensively researched over the past two decades. As noted by Beyer [Bey00], the commonly accepted viewpoint is that Evolutionary Algorithms are fairly robust against noise in the objective function based on the two empirical arguments that 1) evolution in nature is also highly influenced by noise, yet seems to work fine, and 2) in practice, Evolutionary Algorithms have shown to yield usable results for practical noisy optimization problems (that is, in the sense of melioration). Here it is considered that the fitness of each individual is determined by using one noisy evaluation (i.e., $\hat{f}_{\text{eff}}(\mathbf{x}) = \tilde{f}(\mathbf{x})$) that effectively approximates the expected objective function (i.e., $f_{\text{eff}}(\mathbf{x}) = f_{\text{exp}}(\mathbf{x})$).

In Evolutionary Algorithms, essentially only the selection operation is directly influenced by noise. Moreover, as noted by Heidrich-Meisner [HM11], for rank based selection, noise only affects selection when it changes the ranking among the individuals of the population.

The presence of noise does not necessarily have a negative impact on the performance of Evolutionary Algorithms. Noise has similar effects as the randomness that is intentionally included in commonly used selection methods, like the randomness in proportional selection and tournament selection in Genetic Algorithms [Bäc96], and this randomness can help to escape local optima. These alleged benefits of randomness induced by noise are supported by studies on theoretical cases in which adding a small noise signal to the original objective function yielded better convergence reliability [BH94] or even a higher convergence velocity [MNB08].

On the other hand, noise can also have harmful effects. The study of Beyer [Bey00] showed that for a simple quadratic function with stationary Gaussian noise, Evolutionary Algorithms fail to get infinitely close to the optimum. Instead, the population stagnates at a certain residual distance from the optimum. Given the general insight that the regions around the local optima of many continuous functions can be approximated by a quadratic model, similar effects can be expected for a wide range of noisy optimization problems.

An intuitive explanation of the reason why Evolutionary Algorithms are relatively good in dealing with noisy objective functions is that in the early stages of the evolution process the differences in fitness between all pairs of individuals are generally much bigger than the variations due to noise. Because of this, the selection mechanisms will keep a strong bias toward selecting the better solutions for reproduction, which is sufficient for Evolutionary Algorithms to progress. However, as the evolution proceeds and the population zooms in on an optimum, the differences in both the search space as well as the objective space will decrease, whereas the variance of the noise factor generally stays at the same order of magnitude. Hence, the signal to noise ratio decreases and in effect the bias toward selecting better solutions will decrease. Eventually, the selection process will degrade to uniform random selection.

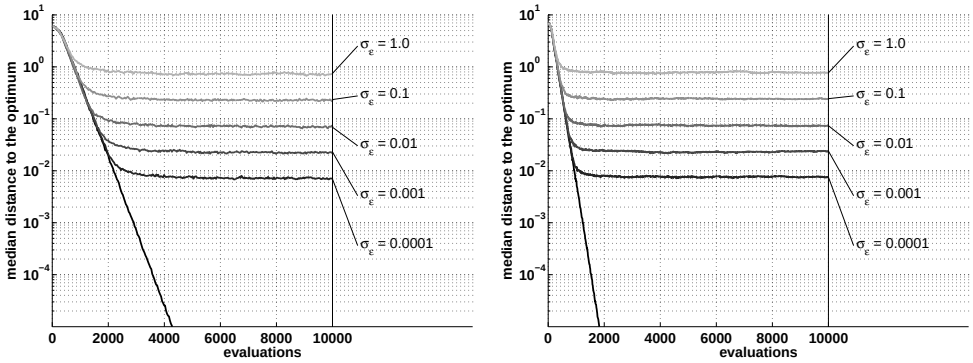

Figure 5.2: The convergence dynamics in terms of distance to the optimizer versus number of evaluations (median over 100 runs) of the $(5/2_{DI}, 35)$-σSA-ES (left) and the CMA-ES (right) on the noisy sphere problem for different noise levels ($\sigma_\epsilon = 0$, $\sigma_\epsilon = 0.0001$, $\sigma_\epsilon = 0.001$, $\sigma_\epsilon = 0.01$, $\sigma_\epsilon = 0.1$, and $\sigma_\epsilon = 1$).

Hence, although noise may be advantageous in some cases or at some stages of the optimization process, it can deteriorate the accurate localization of an optimum, and canonical Evolutionary Algorithms need to be equipped with noise handling mechanisms in order to enable them to locate optima of the expected objective function more precisely.

To illustrate the effect of noise on Evolution Strategies, and in particular on the variants that are the focus of this work, we set up the following experiment:

Experiment 5.2.1 (Performance of Evolution Strategies on the noisy sphere problem)**:** We perform 100 runs of a $(5/2_{DI}, 35)$-σSA-ES (see Section 4.2.2) and a CMA-ES (see Section 4.2.3) on the 10-dimensional noisy sphere problem (see Appendix A.1) with varying noise levels ($\sigma_\epsilon = 0$, $\sigma_\epsilon = 0.0001$, $\sigma_\epsilon = 0.001$, $\sigma_\epsilon = 0.01$, $\sigma_\epsilon = 0.1$, and $\sigma_\epsilon = 1$). Each run has a budget of $10,000$ function evaluations.

Figure 5.2 shows the results of Experiment 5.2.1 by means of the performance, measured in terms of the median distance to the optimizer, versus evaluations. The plots show that for both algorithms the performance in the early stages of the evolution is not affected by noise, yielding the same *progress rate* (that is, the improvement in the direction of the optimum) for each noise level. However, for each noise level, there is a specific point when the progress rate deteriorates and finally reaches zero. That is, the optimization process stagnates at a certain distance to the optimizer. The higher the noise level, the earlier the stagnation and the higher the distance to the optimizer at which the stagnation occurs.

For the $(1, \lambda)$-σSA-ES, the (μ, λ)-σSA-ES (without recombination) and the $(\mu/\mu, \lambda)$-σSA-ES, Beyer [Bey00] derived lower bounds for the residual distance R_∞ for the noisy sphere problem. For the (μ, λ)-σSA-ES (without recombination) on the noisy sphere problem it is

derived as

$$R_\infty \geq \frac{1}{2}\sqrt{\frac{\sigma_\epsilon N}{4\sqrt{\mu}c_{\mu,\lambda}}}\ ,\ c_{\mu,\lambda} \sim \sqrt{2\ln(\lambda/\mu)}, \tag{5.10}$$

with $c_{\mu,\lambda}$ being the so-called *progress coefficient* (see [Bey94] for the derivation of the progress coefficient). An approximation for the $(\mu/\mu,\lambda)$-σSA-ES, which most closely resembles the weighted recombination used in the CMA-ES, was obtained by Arnold and Beyer [AB02] as

$$R_\infty \simeq \frac{1}{2}\sqrt{\frac{\sigma_\epsilon N}{4\mu c_{\mu/\mu,\lambda}}}\ ,\ c_{\mu/\mu,\lambda} = \mathcal{O}(\sqrt{\ln(\lambda/\mu)}), \tag{5.11}$$

with $c_{\mu/\mu,\lambda}$ being derived in [Bey95, Bey96]. From this, it can be concluded that, in order to increase the convergence accuracy of Evolution Strategies, either the population size should be increased (that is, either μ, λ, or both) or the noise factor σ_ϵ should be decreased. Moreover, comparing Eq. 5.10 with Eq. 5.11, it can be concluded that for the noisy sphere problem, using (multi-) recombination improves convergence accuracy. Regarding the latter, Hammel and Bäck [HB94] reported that also two-parent recombination improves the performance of Evolution Strategies on noisy functions. Section 5.3 will discuss the technique of increasing the population size or decreasing the evaluation error as an active way of noise handling in more detail.

Finally, an issue specific for Evolution Strategies is the effect of noise on the adaptation of the strategy parameters (i.e., the stepsize, and for the CMA-ES also the update of the covariance matrix). Obviously, when the signal-to-noise ratio within a population becomes too small, the failures in selecting the fitter individuals will also affect the adaptation of the strategy parameters. However, especially when considering noise handling schemes it is important to know how the adaptation mechanisms of the strategy parameters are affected by noise, and, following that, at which noise ratio these adaptation mechanisms will yield inappropriate/counterproductive parameter settings.

Using the results of Experiment 5.2.1, Figure 5.3 shows the development of the stepsize for the noise level $\sigma_\epsilon = 1.0$ (both the mean performance and the development of a single run) compared to the stepsize development for the noise-free case. For the $(5/2_{DI},35)$-σSA-ES, the stepsize in each generation is the average stepsize of all selected parents, and for the CMA-ES, the plotted stepsize is the scaling factor of the mutations, σ (i.e., not accounting for the covariance matrix factor). It can be seen that also the stepsize stagnates at some level. Hence, the mutations remain fairly high although effectively the population does not get closer to the optimum. This behavior is comparable for both algorithmic schemes (although they use different stepsize adaptation mechanisms). Moreover, the single run dynamics show that the stepsize develops like a bounded random walk.

Not many studies exist on the adaptation of the stepsize in noisy scenarios. Two studies by Arnold and Beyer [AB04, AB08] consider the behavior of cumulative stepsize adaptation.

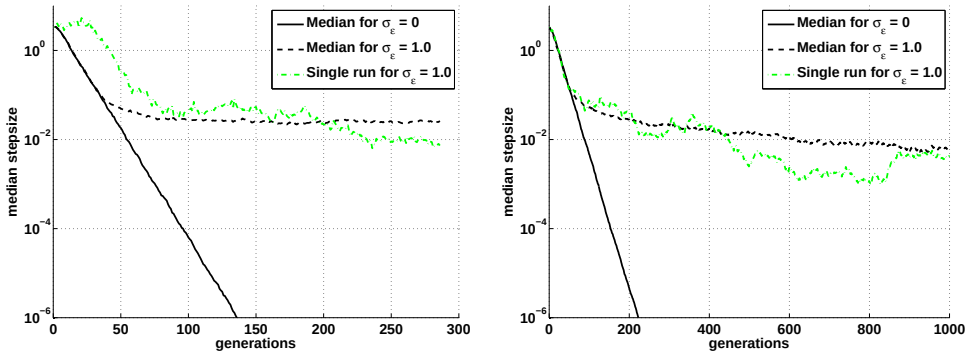

Figure 5.3: The stepsize development (both of a single run and the median over 100 runs) of the $(5/2_{DI}, 35)$-σSA-ES (left) and the CMA-ES (right) on the noisy sphere problem with noise level $\sigma_\epsilon = 0.1$ compared to the stepsize development on the noise-free sphere problem (median over 100 runs).

They conclude that noise affects the proper adaptation of the mutation strength as compared to the theoretical optimal mutation strength, and that this effect can be counteracted (again) by increasing the population size. Another interesting result is presented in [Bey00], showing an example where the failure of a proposed noise handling scheme is attributed primarily to a wrong adaptation of the stepsize.

In conclusion, we can summarize the findings from literature that are most interesting for practical application of Evolution Strategies on noisy objective functions:

- The robustness with respect to noisy objective functions is implicitly defined as the ability of Evolutionary Algorithms of finding high quality solutions with respect to the expected objective function value (i.e., $f_{\text{eff}} = f_{\text{exp}}$).
- As long as the noise level is small compared to the difference in objective function values of the individuals, noise does not affect the performance of Evolutionary Algorithms.
- Increasing the population size (μ, λ, or both) increases the convergence accuracy, meaning that the population will be able to converge closer to a local optimizer.
- Using any common type of recombination increases the ability to closer approximate the optimum on the noisy sphere problem.
- For adaptation of the strategy parameters, increasing the population size increases the reliability of the adaptation of the stepsize for cumulative stepsize adaptation.

5.3 Basic Noise Handling

As mentioned, Evolutionary Algorithms and Evolution Strategies are fairly robust against noise. However, at a certain point during the optimization when the signal-to-noise ratio

becomes too low, they will stagnate and not converge any closer to the optimum. When more accurate localization of an optimizer is required, additional measures are needed. This section will provide an overview of some basic techniques that can be used in the context of optimization of the expected objective function.

5.3.1 Resampling

Resampling or *explicit averaging* is a straightforward approach to obtain better convergence accuracy by approximating f_{exp} as the sample mean over m samples

$$\hat{f}_{\text{exp}}(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^{m} \tilde{f}(\mathbf{x}). \tag{5.12}$$

For Gaussian noise $z(\mathbf{x}) \sim \mathcal{N}(0, \sigma_\epsilon^2)$, this yields an approximation error of

$$\bar{\sigma}_\epsilon = \sqrt{\text{Var}[\hat{f}_{\text{exp}}(\mathbf{x})]} = \sigma_\epsilon / \sqrt{m}. \tag{5.13}$$

Because the sample mean is an unbiased estimator of f_{exp}, this approach effectively reduces the noise level of objective function $\tilde{f}$ by a factor of $\sqrt{m}$, allowing for a closer convergence to the optimum. An obvious downside of this approach is that using multiple samples per fitness evaluation increases the computational effort by a factor m. Especially for limited evaluation budgets, determining an appropriate setting for m can be a tedious task.

5.3.2 Increasing the Population Size

As discussed in the previous section, for Evolution Strategies it has been observed that increasing the population size is also a way of improving the convergence accuracy. Interestingly, also in the context of Genetic Algorithms, this observation was made by Fitzpatrick and Grefenstette [FG88], and Miller and Goldberg [MG96] showed that for infinite population sizes, proportional selection is not affected by noise. Increasing the population size as an active way of noise handling is also referred to as *implicit averaging*, as opposed to the alternative of *explicit averaging* by means of resampling.

For Evolution Strategies, increasing both μ and λ can reduce the effects of noise. Considering that increasing μ does not increase the number of evaluations per generation, this may suggest that we have a free way of increasing the convergence accuracy. However, as noted in [HB94], the price of increasing μ is a lower selection pressure, which yields a lower convergence speed. Hence, increasing μ has an indirect effect on the convergence speed. Alternatively, one could increase both μ and λ. Although this does have a direct effect on the number of evaluations per generation, and yields a slower convergence speed, it leads to a higher convergence accuracy.

In order to obtain a clearer view on the effects of noise and different population sizes for the two particular schemes considered in this work, consider the following small experiment:

$(\mu/2_{DI}, \lambda)$-σ**SA-ES**	**CMA-ES**
$\lambda = 35$, $\mu = 5, 10, 15, 20, 25, 30$	$\lambda = 10$, $\mu = 3, 5, 7, 9$
$\lambda = 100$, $\mu = 15, 25, 35, 45, 55, 65$	$\lambda = 30$, $\mu = 10, 15, 20, 25, 30$

Table 5.1: The population size settings considered in Experiment 5.3.1.

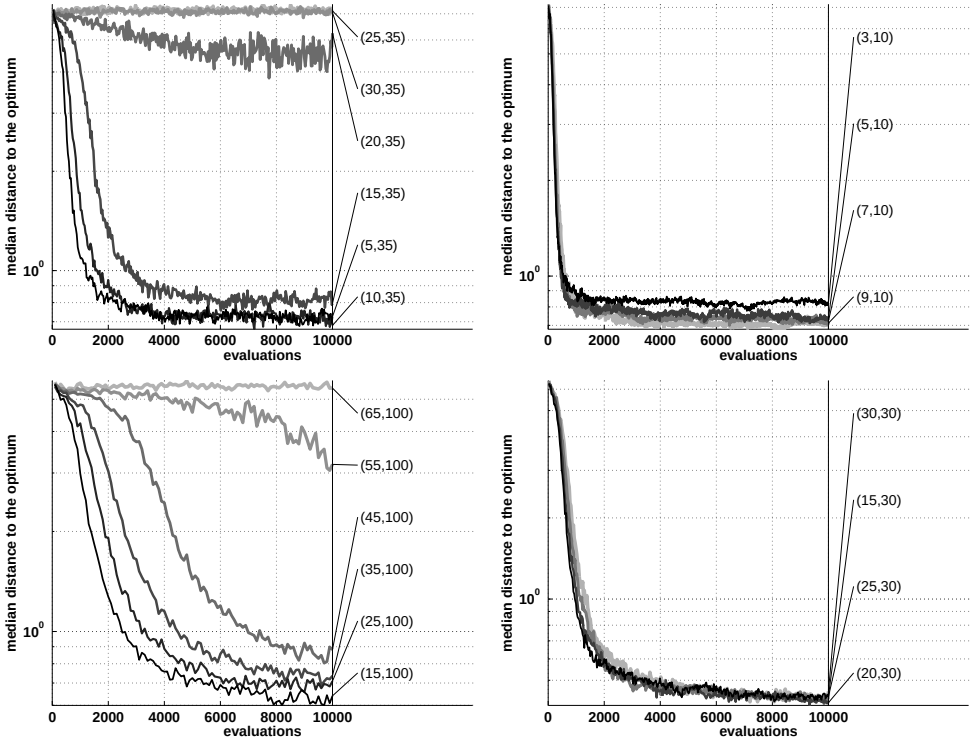

Figure 5.4: The performance of the $(\mu/2_{DI}, \lambda)$-σSA-ES (left) and the CMA-ES (right) on the noisy sphere problem for varying population sizes. The top row shows the default value for λ, varying μ, the bottom row shows a value of λ that is approximately three times the default value, again varying μ. The performance is measured in terms of distance to the optimizer versus evaluations (median over 100 runs).

Experiment 5.3.1 (Effect of higher population sizes on the noisy sphere problem)**:** We perform 100 runs of a $(\mu/2_{DI}, \lambda)$-σSA-ES (see Section 4.2.2) and a CMA-ES (see Section 4.2.3) on the 10-dimensional noisy sphere problem (see Appendix A.1). For each scheme, two settings of λ are considered, each with varying settings of μ (see Table 5.1). For each run, an evaluation budget of $10,000$ function evaluations is used.

Figure 5.4 shows the results of Experiment 5.3.1 by means of plots of the performance, measured in terms of the median distance to the optimizer, versus the number of evaluations. From these results, two conclusions can be drawn: First, increasing λ yields a higher convergence accuracy, but a lower convergence speed. This is indeed in line with what was expected.

Second, increasing μ can improve convergence quality, but for the $(\mu/2_{DI}, \lambda)$-σSA-ES, the convergence speed drastically decreases with increasing values for μ, whereas for the CMA-ES, $\mu \approx \lambda$ still seems to work and actually yields a good trade-off between convergence speed and convergence accuracy. The latter is a remarkable result, as setting $\mu = \lambda$ seemingly implies that there is no selection pressure. A simple explanation for why this works for the CMA-ES is that it uses weighted recombination, assigning a higher weight to the fitter individuals in the logarithmically weighted average (which can be seen as an implicit selection mechanism).

In [Bey00], it is recommended that for the $(\mu/\mu_I, \lambda)$-ES, given a fixed offspring number λ, the value of μ should be chosen such that $\mu = \lambda/2$. For the (μ, λ)-σSA-ES, in [HB94] it is recommended to set $\mu \approx 1/7$, whereas in [Bey00], a derivation based on Eq. 5.10 showed that it should be set to $\mu = \lambda/e$. Based on this, and on the results shown in Figure 5.4, we postulate that good alternative settings in case of 10-dimensional noisy objective functions with stationary noise, are: $\mu \approx \lambda/e$ for the (μ, λ)-σSA-ES and $\mu \approx \lambda$ for the CMA-ES (this should be investigated in more depth). For the setting of λ, there is an inherent trade-off between convergence speed and convergence accuracy, making it purely dependent on the available budget of function evaluations.

5.3.3 Implicit Averaging versus Explicit Averaging

Given the two techniques to increase convergence accuracy, implicit and explicit averaging, the question arises which one is better. In [BOS03], it is stressed that for the $(\mu/\mu_I, \lambda)$-ES, given a fixed noise strength, it is more efficient to increase the offspring number by a factor m instead of resampling the objective function m times. On the other hand, Hammel and Bäck [HB94] conclude exactly the opposite based on an experimental study on the (μ, λ)-ES (i.e., resampling is better than increasing the population size). In order to form a picture, we perform the following experiment:

Experiment 5.3.2 (Implicit versus explicit averaging)**:** For comparing implicit versus explicit averaging, we perform 100 runs of a $(\mu/2_{DI}, \lambda)$-σSA-ES (see Section 4.2.2) and a CMA-ES (see Section 4.2.3) on the 10-dimensional noisy sphere problem (see Appendix A.1) and on a multimodal problem; the 10-dimensional noisy Griewank problem (see Appendix A.6). We take for implicit averaging for the $(\mu/2_{DI}, \lambda)$-σSA-ES: a $(5, 35)$-, a $(25, 175)$-, and a $(50, 350)$-strategy, and for the CMA-ES: a $(5, 10)$-, a $(25, 50)$-, and a $(50, 100)$-strategy. For the resampling schemes we consider: $m = 1$ (i.e., no resampling), $m = 5$, $m = 10$. Evaluation budget for each run: $10,000$.

Figure 5.5 shows for Experiment 5.3.2 the convergence plots in terms of distance to the optimizer versus evaluations. Interestingly, we can observe a remarkable difference between the $(\mu/2_{DI}, \lambda)$-σSA-ES and the CMA-ES. For the $(\mu/2_{DI}, \lambda)$-σSA-ES, explicit resampling seems to yield better results than implicit averaging. That is, using larger population sizes seems to slow down the convergence. However, for the CMA-ES, increasing the population

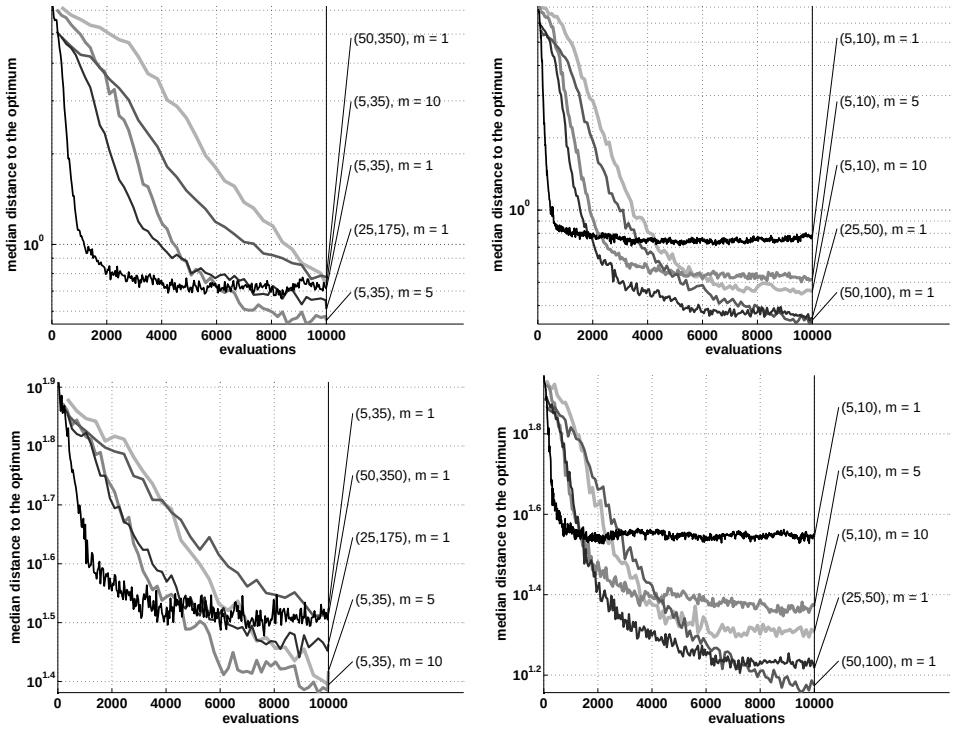

Figure 5.5: The performance of the $(\mu/2_{DI}, \lambda)$-σSA-ES (left) and the CMA-ES (right) on the noisy sphere problem (top row) and the noisy Griewank problem (bottom row). Comparing implicit versus explicit resampling. The performance is measured in terms of distance to the optimizer versus evaluations (median over 100 runs).

size seems to be most beneficial. Moreover, for both implicit averaging and explicit averaging, the CMA-ES obtains much better results than the $(\mu/2_{DI}, \lambda)$-σSA-ES. The results shown in Figure 5.5 can well explain the difference between the conclusions of Hammel and Bäck [HB94], and those of Beyer [BOS03].

5.3.4 Rescaled Mutations

An alternative approach of noise handling that does not require additional evaluations per generation is to use *rescaled mutations*. This technique was proposed originally by Rechenberg [Rec94] and further investigated by Beyer [Bey98, Bey00]. Using rescaled mutations is based on the slogan "*mutate large, but inherit small*" and it basically does just that. Instead of performing mutation in the normal way, a large mutation is applied to each individual (that is, large compared to the current stepsize) and selection is based on the fitness values of the offspring generated by these large mutations. However, after selection, instead of using the large mutation for the selected offspring, the mutation is rescaled to a small mutation. Hence, selection is based on large mutations, while small mutations in the same directions as the

successful large mutations are eventually used after selection.

As an example, consider the $(1, \lambda)$-strategy and let offspring i be obtained by $\mathbf{x}_i = \mathbf{x}_\mathrm{p} + \mathbf{z}_i$, $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$. Now, let $\mathbf{x}_{1:\lambda}$ denote the best offspring, which was generated by mutation $\mathbf{z}_{1:\lambda}$. Instead of using $\mathbf{x}_{1:\lambda}$ as parent for the next generation, the mutation is rescaled by a factor of $1/\kappa$, $\kappa \geq 1$. Hence, the parent for the next generation is computed as

$$\mathbf{x}_\mathrm{p} = \mathbf{x}_\mathrm{p} + \frac{1}{\kappa}\mathbf{z}_{1:\lambda}. \tag{5.14}$$

When assuming local or quasi linearity of the search space, the direction of the best large step should also be the best direction of improvement for a small step. Hence, by using large mutations for evaluations, differences between individuals become more apparent.

Although the idea behind using rescaled mutations is appealing and provides theoretically promising results for the noisy sphere problem, in practice, it does not yield convincing results, not even for the noisy sphere problem (see, e.g., [Bey00]). For the noisy sphere problem, this is attributed to a wrong adaptation of the stepsize (even after inclusion of fixes for the stepsize adaptation). For other problem types, one could argue that the assumed quasi linearity of the search space only holds very locally, i.e., only works for $\kappa \approx 1$. Based on these considerations, we can conclude that this noise handling technique is not off-the-shelf usable in practical scenarios.

5.3.5 Thresholding

Thresholding is a noise handling technique proposed by Markon et al. [MMA+01]. Thresholding is a simple technique, however, usable only for *plus*-selection strategies. The idea behind thresholding is that an offspring is only accepted to replace a parent if it is at least a constant $\tau > 0$ better. That is, in order to prevent the selection of outliers, offspring are required to be considerably fitter than their parents in order to be selected.

The study by Markon et al. [MMA+01] uses the $(1+1)$-ES as algorithmic basis and includes theoretical analysis of thresholding on the noisy sphere problem with Gaussian noise. For this setting, they provide a derivation on how to set τ when having an estimate of the noise strength and an estimate of the fitness difference with respect to the optimum. For this setting of τ, the $(1+1)$-ES with thresholding can converge arbitrarily close to the optimum.

Although the theory behind the study of Markon et al. [MMA+01] is sound and the results look promising, the gap with practical applicability is still considerable. Obtaining accurate estimates of the noise strength and the fitness difference with respect to the optimum is a tedious task in itself. Furthermore, the extension to multi-membered *comma*-strategies requires a number of adaptations. An approach that can be considered as a multi-membered extension of this approach will be described in Section 5.4.3.

5.4 Adaptive Averaging

Among the basic noise handling approaches discussed up to now, the two straightforward approaches of implicit and explicit averaging seem the most effective. However, both suffer from the problem that they only decrease the effect of noise, but do not eliminate it. Moreover, these techniques introduce a trade-off between using a small population/sample size that yields a high convergence speed, but a low convergence accuracy versus using a large population/sample size that yields a low convergence speed, but high convergence accuracy.

From this perspective it is desirable to have a method that adapts the intensity of the noise handling scheme such that convergence is maintained, but evaluations are not wasted. For implicit averaging, this means to start out with a small population size, and increase the population size when the population converges. For explicit averaging, this could be done in a similar way, and in addition one could distribute the sampling budget over the individuals in an efficient way. Methods that control the evaluation intensity are referred to as *adaptive averaging* methods. Several of such techniques have been proposed in the context of Evolutionary Algorithms. This section summarizes the most prominent ideas.

5.4.1 Duration Scheduling and Sample Allocation

Aizawa and Wah [AW93, AW94] proposed two adaptive resampling schemes for noisy objective functions in the context of Genetic Algorithms, based on two underlying scheduling problems that emerge when dealing with noisy objective functions:

- **Duration scheduling problem**: the problem of determining whether the quality of the objective function approximations of the individuals in the population is sufficient to end the current generation and use the current approximations for selection.

- **Sample allocation problem**: the problem of allocating evaluations (samples) to each individual in the population given a budget of evaluations such that it is most beneficial for the current generation.

The former emerges when premature convergence needs to be prevented while having no practical limitations on the evaluation budget. The latter emerges when evaluation is costly and it is important to spend the evaluations as effectively as possible within each generation. They proposed two separate approaches for these two scheduling problems.

Both the duration scheduling approach and the sample allocation approach are based on two assumptions: 1) the noise is stationary and has a Gaussian distribution, and 2) the "real" underlying objective function values of the individuals in a population are normally distributed. Based on these assumptions and on approximations of the parameters of both distributions (see Technical Note 5.1), a Bayesian approach is used to approximate the fitness of each individual (see Technical Note 5.2).

Technical Note 5.1: Estimating the Population and Noise Variance

When assuming that the noise in the objective functions is stationary and has a Gaussian distribution, $\mathcal{N}(0, \sigma_\epsilon^2)$, and the "real" underlying objective function values of the individuals in a population are also normally distributed, $\mathcal{N}(f_0, \sigma_0^2)$, then σ_0^2, f_0, and σ_ϵ^2 can be approximated as

$$\hat{\sigma}_\epsilon^2 = \frac{\sum_{i=1}^{\lambda}((m_i - 1) \cdot s_i^2)}{(\sum_{i=1}^{\lambda} m_i) - \lambda}, \; s_i^2 = \sum_{j=1}^{m_i}(\tilde{f}_{i,j} - \bar{f}_i)^2, \tag{5.15}$$

$$\hat{f}_0 = \frac{1}{\lambda} \cdot \sum_{i=1}^{\lambda} \bar{f}_i, \tag{5.16}$$

$$\hat{\sigma}_0^2 = \frac{1}{\lambda \cdot (\lambda - 1)} \left(\lambda \cdot \sum_{i=1}^{\lambda} (\bar{f}_i)^2 - \left(\sum_{i=1}^{\lambda} \bar{f}_i \right)^2 \right) - \frac{\hat{\sigma}_\epsilon^2}{m}, \tag{5.17}$$

where $m_1 = \ldots = m_\lambda = m$ is assumed.

Remark: An issue not mentioned in [AW94], but very relevant in practice, is that the estimate from Eq. 5.17 becomes unusable when the ratio $\sigma_0^2/(\sigma_\epsilon^2/m)$ becomes too small. That is, if the ratio between the population variance and the variance of the sample mean (σ_ϵ^2/m) becomes smaller, the estimate $\hat{\sigma}_0^2$ becomes less accurate and might even become negative (which, being a variance, is an unreasonable estimate). This should be accounted for in practical situations, because it harms the fitness estimates. Furthermore, it should be noted that this approach requires at least two samples for each individual in the population.

The **duration scheduling** approach aims to automatically adapt the number of samples used for resampling such that the Evolutionary Algorithm maintains progress. Primarily it uses a static incremental scheme, determining a budget t_k of samples available for generation k as

$$t_k = \lambda \cdot \left(m_0 + \left\lceil \gamma \sum_{i=1}^{k-1} t_i \right\rceil \right), \tag{5.22}$$

where λ is the population size, m_0 is the initial number of samples used for each individual, and γ is a heuristic parameter. Secondly, it uses the following bounds for the ratio between the effective variance of the noise (see Eq. 5.13) and the population variance as an indicator of whether the current generation can be terminated:

$$\delta_l \leq \frac{\sigma_\epsilon/\sqrt{m}}{\sigma_0} \leq \delta_u. \tag{5.23}$$

If this ratio is small enough (i.e., $< \delta_l$), no resampling is necessary and the resampling loop is terminated even if the current evaluation budget t_k is not spend entirely. When it is greater than δ_u, resampling is necessary and resampling is continued even if the evaluation budget t_k is exceeded. Here, σ_ϵ and σ_0 are estimated as described in Technical Note 5.1.

Technical Note 5.2: Bayesian Fitness Approximation

Assume that the objective function value of candidate solution $\mathbf{x}_i$ is normally distributed, i.e.,

$$\tilde{f}_i \sim \mathcal{N}(f_i, \sigma_\epsilon^2). \tag{5.18}$$

Furthermore, assume that the "real" objective function values $f_1, \ldots, f_\lambda$ of the λ individuals within the population are normally distributed, $\mathcal{N}(f_0, \sigma_0^2)$, yielding for each individual i, a *prior distribution* $h(f_i) \sim \mathcal{N}(f_0, \sigma_0^2)$.

Let $\bar{f}_i$ be the mean of m_i fitness evaluations $(\tilde{f}_{i,1}, \ldots, \tilde{f}_{i,m_i})$ for candidate solution $\mathbf{x}_i$. By definition, we know:

$$p(\bar{f}_i \mid f_i) \sim \mathcal{N}(f_i, \sigma_\epsilon^2/m_i). \tag{5.19}$$

Using Bayes formula, we obtain a *posterior distribution* for individual i

$$h^*(f_i \mid \bar{f}_i) = \frac{p(\bar{f}_i \mid f_i) \cdot h(f_i)}{\int_{-\infty}^{\infty} p(\bar{f}_j \mid f_j) \cdot h(f_j) \cdot df_j}. \tag{5.20}$$

From this, the best estimator $\hat{f}_i$ for f_i with estimation error $\hat{\sigma}_i$ is given by

$$\hat{f}_i = \frac{m_i \cdot \bar{f}_i + \alpha \cdot f_0}{m_i + \alpha}\ ,\ \hat{\sigma}_i = \frac{\sigma_\epsilon^2}{m_i + \alpha}\ ,\ \alpha = \frac{\sigma_\epsilon^2}{\sigma_0^2}. \tag{5.21}$$

The values of σ_0^2, f_0, and σ_ϵ^2 can be estimated as described in Technical Note 5.1.

Regarding the implementation details of the duration scheduling approach, $\gamma = 5 \times 10^{-3}$, $\delta_l = 1.0$, $\delta_u = 4.0$ are used in [AW94]. The setting of m_0 is noted to be based on Eq. 5.23, however, the exact procedure is not described in [AW94]. Besides that, a pre-sampling step should be used to estimate σ_ϵ and σ_0, which are required within Eq. 5.23, but this pre-sampling step is not described explicitly. Also, no note is made of how often the estimates of σ_ϵ and σ_0 are updated. For the implementation of this scheme, these issues should be accounted for.

The **sample allocation** approach uses a fixed budget of objective function evaluations T every generation and aims to divide the evaluations such that it is most beneficial for selection. Technical Note 5.3 summarizes this sample allocation procedure[1]. The idea is to distribute the evaluation budget $T = (m_1 + \ldots + m_\lambda)$ in an optimal way among the individuals of the current generation. This can be accomplished by selecting the individual for resampling for which resampling will lead to the highest reduction of the following expected risk function:

$$\bar{R} = \sum_{i=1}^{\lambda} P_i \hat{\sigma}_i^2. \tag{5.24}$$

[1] The description in Technical Note 5.3 differs slightly from the description presented in [AW94], which is done for the sake of clarity.

Technical Note 5.3: Sample Allocation Procedure

1. Take a fixed number of samples (at least two[a]) for each individual in the population and set the evaluation counter $t = m_0 \cdot \lambda$.
2. For each individual $i = 1, \ldots, \lambda$ compute $\hat{\sigma}_i$ and P_i (or w_i as an approximation).
3. Take one additional sample for the individual that has the largest feedback value according to Eq. 5.28 and increase the evaluation counter $t = t + 1$.
4. Repeat step 2 to 4 until $t = T$.

[a] In [AW94] it is said to take one sample for each individual. However, at least two samples for each individual are needed to obtain estimates for σ_ϵ and σ_0.

Here, P_i is the probability that individual i is the best individual and $\hat{\sigma}_i^2$ is the estimated prediction error of individual i (see Eq. 5.21). Technical Note 5.4 describes the procedure to find the individual that contributes most to the risk function Eq. 5.24.

To summarize, Aizawa and Wah [AW94] proposed two adaptive resampling methods that are both based on a Bayesian approach for estimating the fitness that differs from the common way of doing explicit averaging. Furthermore, they introduced a way of measuring the selection uncertainty based on the ratio between the population variance and the approximation error Eq. 5.23, which is used in a duration scheduling approach. For in-generation allocation of additional fitness evaluations, a sample allocation scheme based on an expected risk function is proposed.

For applying the proposed techniques in practice, a few remarks are in place. First, it should be noted that for Evolution Strategies, using solely the Bayesian fitness approximation is not sensible, because it does not change the ranking amongst the individuals in the population as compared to explicit averaging. Furthermore, in [AW94] it is noted that the duration scheduling problem and sample allocation problem do not occur concurrently. However, this is debatable, because even if a sufficiently large evaluation budget is available for duration scheduling, it can still be desirable to spend the evaluations within a generation as effectively as possible. Lastly, no note is made of the possibility that the estimate $\hat{\sigma}_0^2$ (see Technical Note 5.1) can become too crude to be practical, or even negative (making it unusable).

5.4.2 Adaptive Resampling Based on the t-Test

Another class of adaptive averaging schemes is formed by approaches that apply reevaluation based on statistical testing of the ranking of the individuals. An obvious first choice is the t-test, which can be used for pairwise comparison of solutions. This approach is suitable when optimizing on the expected objective function, assuming that the noise on the objective

Technical Note 5.4: Minimization of the Expected Risk

Given Bayesian objective function approximations as described by Technical Note 5.2, then the probability P_i that individual i is the best individual is given by

$$P_i = \int_{-\infty}^{\infty} \left[\prod_{j \neq i} H^*(f_j \mid \bar{f}_j) \right] h^*(f_i \mid \bar{f}_i) df_i, \tag{5.25}$$

where, $H^*(f_j \mid \bar{f}_j)$ is the cumulative distribution function of $h^*(f_i \mid \bar{f}_i)$. Given that $\phi(.)$ and $\Phi(.)$ are the probability density function and the cumulative distribution function of the Gaussian distribution, respectively, and using the approximations of $\hat{f}_i$ and $\hat{\sigma}_i$, this becomes:

$$P_i = \int_{-\infty}^{\infty} \left[\prod_{j \neq i} \Phi\left(\frac{f_i - \hat{f}_j}{\hat{\sigma}_j} \right) \right] \phi\left(\frac{f_i - \hat{f}_j}{\hat{\sigma}_j} \right) df_i. \tag{5.26}$$

Alternatively, given that the computation of P_i requires numerical integration, a weight w_i can be used instead of P_i, in which each individual is only compared to the best (or second best if it is the best itself):

$$w_i = \Phi\left(\frac{\hat{f}_i - \hat{f}_j}{\sqrt{\hat{\sigma}_i^2 + \hat{\sigma}_j^2}} \right) \ , \ j = \begin{cases} k & , i \neq k \\ l & , i = k \end{cases}, \tag{5.27}$$

where k is the index of the best individual, and l is the index of the second best individual, based on the fitness approximations $\hat{f}_1, \ldots, \hat{f}_\lambda$.

Based on either P_i or w_i, the individual that should be selected for reevaluation in order to minimize the risk function of Eq. 5.24 is the individual i, computed as

$$\underset{i \in \{1,\ldots,\lambda\}}{\operatorname{argmax}} \left[P_i \frac{\hat{\sigma}_\epsilon^2}{(m_i + \alpha)^2} \right]. \tag{5.28}$$

Technical Note 5.5: Fitness Comparison Using the t-Test

Assume that the fitness of individual i is normally distributed, i.e.,

$$\tilde{f}_i \sim \mathcal{N}(f_i, \sigma_\epsilon^2). \tag{5.29}$$

Given two individuals $\mathbf{x}_i$ and $\mathbf{x}_j$, with mean fitness values $\bar{f}_i$ and $\bar{f}_j$ obtained through resampling using m_i and m_j samples respectively, sample variances s_i^2 and s_j^2, and suppose $\bar{f}_i \leq \bar{f}_j$. We can test the hypothesis

$$H_0 : \bar{f}_i \leq \bar{f}_j \tag{5.30}$$

against the hypothesis

$$H_1 : \bar{f}_i > \bar{f}_j \tag{5.31}$$

using a one-sided t-test to a significance α. Given the t-statistic

$$t_{ij} = \frac{\bar{f}_i - \bar{f}_j}{\sqrt{\frac{s_i^2}{m_i} + \frac{s_j^2}{m_j}}}, \tag{5.32}$$

we can reject H_0 if $t_{ij} > t_{(\alpha, 2m-2)}$. Here, $t_{(\alpha,2m-2)}$ is the t-distribution with $2m-2$ degrees of freedom, computed for α.

function is Gaussian. Approaches based on this statistical testing have been proposed in different studies in different settings [Sta98, CP04, KEB09a].

The t-test can be used to test whether or not the differences of the mean objective function values of two individuals is significant with a certain significance level. The approach to do so is briefly summarized in Technical Note 5.5. During the evaluation phase of the optimization, one can require for a one-sided t-test with a significance α between (certain/all) pairs of individuals, and continue resampling until this is achieved.

A first consideration that is relevant when following such an approach is whether or not a Bonferroni correction should be applied for multiple comparisons [Dun61]. The choice of applying a Bonferroni correction determines whether the statistics are based on comparison of separate pairs or on comparison of the full population. In [Sta98, CP04, KEB09a], the Bonferroni correction is not used. Secondly, note that instead of using a one-sided t-test as described in Technical Note 5.5, also a two-sided t-test could be used (see [KEB09a]). However, effectively this will not make much difference. That is, for both the one-sided as the two-sided t-test the t-statistic is the same, only the threshold value for a certain significance level α changes, yet this does not yield an essentially different indicator measure for resampling. Lastly, one has to decide which pairs of individuals are compared against each other. The approaches proposed in literature are:

- **Subsequent pair testing:** Based on a population sorted by fitness approximation

$\bar{f}_{1:\lambda}, \ldots, \bar{f}_{\lambda:\lambda}$, test for every subsequent pair of individuals $\{\mathbf{x}_{i:\lambda}, \mathbf{x}_{i+1:\lambda}\}, i = 1, \ldots, \lambda - 1$ whether the fittest is fitter with a significance α. For each individual belonging to a pair for which the significance is too low, take one additional sample. Repeat this loop until no more individuals need resampling. This method was studied in [KEB09a].

- **Test against best:** Sort the population by fitness approximation $\bar{f}_{1:\lambda}, \ldots, \bar{f}_{\lambda:\lambda}$, test every individual $\mathbf{x}_{i:\lambda}, i = 2, \ldots, \lambda$ against the best individual $\mathbf{x}_{1:\lambda}$ for a significance α. For each individual belonging to a pair for which the significance is too low, take one additional sample. Repeat this loop until no more individuals need resampling. This method was also studied in [KEB09a].

- **Pairwise tournaments:** In [CP04], pairwise comparison was used within tournament selection with tournament sizes of two. Within every tournament, resampling is continued until the fittest individual is better with a significance α.

For Evolution Strategies, using $(\mu\overset{+}{,}\lambda)$-selection, only the first two approaches are applicable.

Although the idea of using the t-test for adaptive resampling seems promising at first sight, empirical results in [CP04] and [KEB09a] show that these approaches come with serious problems. In the experiments of [KEB09a], different t-test based evaluation schemes are compared for a $(1, 10)$-σSA-ES on a 10-dimensional noisy sphere problem (with $\sigma_\epsilon = 0.1$) and an evaluation budget of $20,000$ function evaluations. These schemes are: subsequent pair adaptive resampling instances, test against best adaptive resampling instances, and a fixed sample size resampling method with a sample size tuned for this problem instance (which is $m = 50$). From the results, shown in Figure 5.6, it can be seen that tuning the parameter α is quite a tedious task. It should be strict enough in order to determine a correct ranking with sufficient confidence, but on the other hand, if it is too strict, samples might be wasted in assuring that pairs of individuals are actually different. The former leads to early convergence and the latter leads to slow convergence, as can be observed in the plots.

Moreover, a performance loss can even be observed when comparing the best adaptive resampling methods against the best fixed sample size resampling method. The latter can be attributed to an explosion of the number of samples required to achieve a certain significance level for pairs of individuals that happen to have objective function values that lie very close to each other (in the perspective of the overall differences between all λ offspring). These pairs require (unnecessarily) long resampling loops. An even more extreme scenario emerges when two individuals have exactly the same mean objective function value, which can cause (obviously undesirable) infinite evaluation loops. On objective function landscapes that contain many or large plateaus, this method is therefore likely to fail when this scenario is not accounted for.

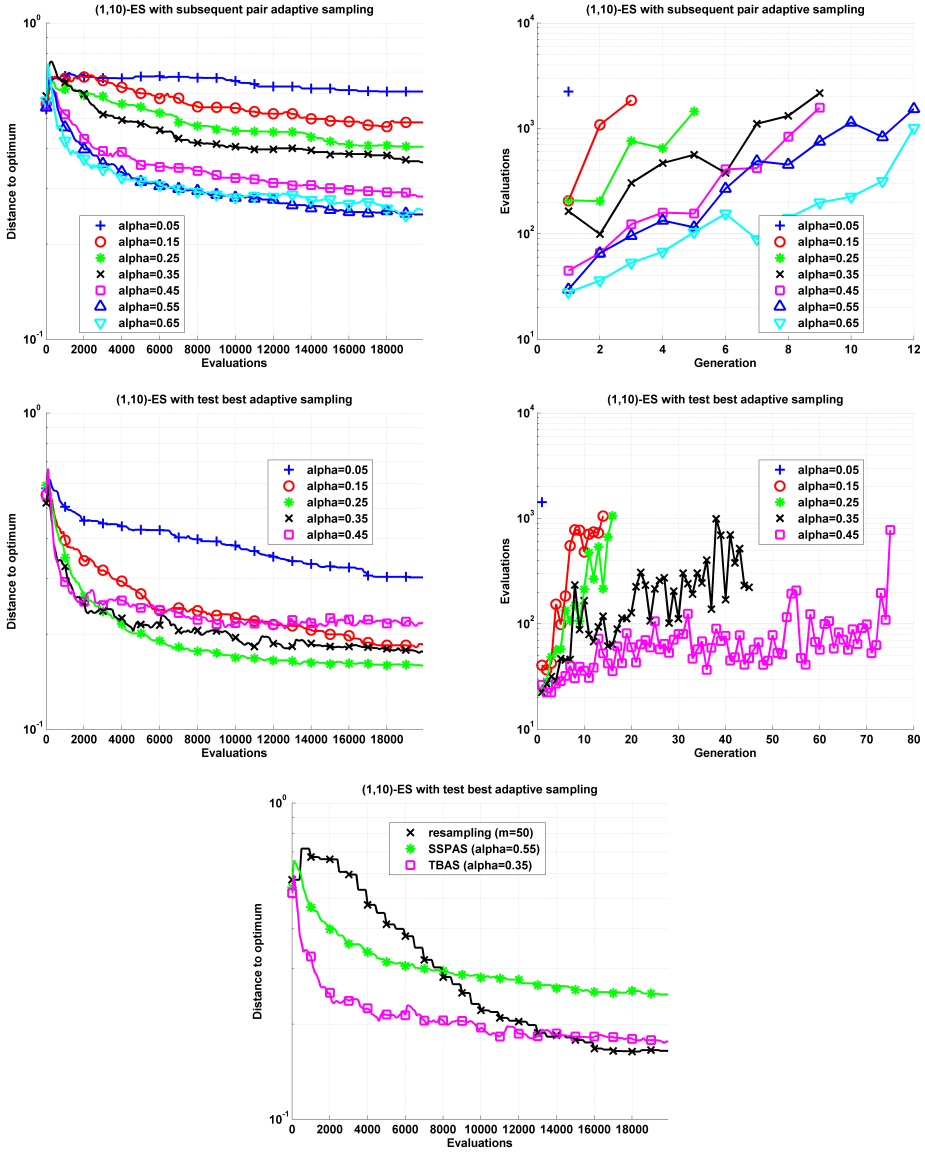

Figure 5.6: Results from [KEB09a]. Different instances of a $(1, 10)$-σSA-ES on a 10-dimensional noisy sphere problem, with $\sigma_\epsilon = 0.1$. Top row: the performance and required evaluations per generation of a $(1, 10)$-σSA-ES implementing the subsequent pair test adaptive sampling approach. Middle row: the performance and required evaluations per generation of a $(1, 10)$-σSA-ES implementing the test against best adaptive sampling approach. Bottom row: the performance of the best adaptive sampling instances compared to the performance when using a good fixed sample size approach. The results were obtained using 10 runs per algorithmic scheme, using an evaluation budget of $20,000$.

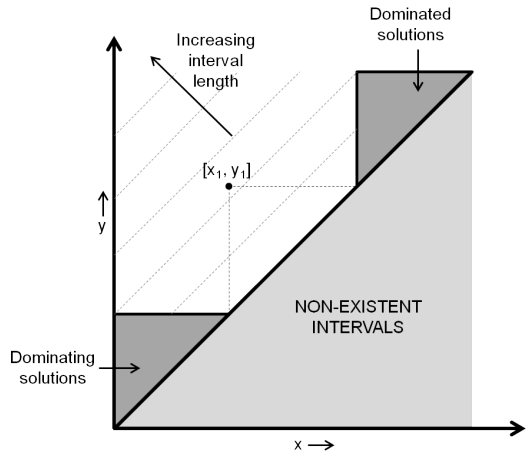

Figure 5.7: The dominance relationship for interval orders visualized geometrically. In this figure, the space of intervals is presented on a two-dimensional plane, divided into the regions of dominating intervals, dominated intervals, and incomparable intervals of the interval $[x_1, y_1]$. The line $x = y$ represents the intervals that are reduced to single points. From this figure, we see that decreasing the interval length decreases the distance to the line $x = y$, which increases the comparability.

5.4.3 Partial Order Based Adaptive Averaging

Rudolph [Rud01] proposed to actively consider the partial order that emerges when considering the noisy fitness of each candidate solution as an uncertainty interval, see Technical Note 5.6. For this, it should be assumed that the noise in the objective function is bounded within known intervals. Given this viewpoint, one could apply an Evolutionary Algorithm that selects based on the dominance relation that emerges from this partial order. Figure 5.7 visualizes the dominance relationship for interval orders. In this figure, the space of intervals is presented on a two-dimensional plane, divided into the regions of dominating intervals, dominated intervals, and incomparable intervals of the interval $[x_1, y_1]$. The line $x = y$ represents the intervals that are reduced to single points. From this figure, we see that decreasing the interval length decreases the distance to the line $x = y$, which increases the comparability.

Rudolph considered an Evolutionary Algorithm using an elitist selection strategy and for which it is guaranteed that every collection of offspring can be generated from any collection of parents. For such algorithms, it holds that for any finite search space with any noisy objective function with the noise bounded within the interval $[-a, a]$, the population will, with a probability 1, after a finite number of generations, enter a state in which all solutions have an objective function value that lies at most $3a$ away from the optimum (see [Rud01] for details).

This scheme can be extended as an adaptive averaging scheme by using the sample mean of m evaluations. For this, confidence intervals can be generated that are stricter than the original bounds, which will lead to more accurate convergence precision. For this, an adaptive resampling technique can be devised as follows: run the Evolutionary Algorithm described

Technical Note 5.6: Partial Orders Induced by Noise

For the set of intervals $\mathcal{F} = \{[x_1, x_2] \subset \mathbb{R} : x_1 \leq x_2\}$, one can introduce a strict partial order $\prec$

$$[x_1, x_2] \prec [y_1, y_2] \quad \text{iff} \quad x_2 < y_1, \tag{5.33}$$

which can be extended to a partial order by adding the relations

$$[x_1, x_2] = [y_1, y_2] \quad \text{iff} \quad x_1 = y_1 \text{ or } x_2 = y_2, \tag{5.34}$$

$$[x_1, x_2] \preceq [y_1, y_2] \quad \text{iff} \quad [x_1, x_2] \prec [y_1, y_2] \vee [x_1, x_2] = [y_1, y_2], \tag{5.35}$$

Although strictly speaking it is sufficient to consider only a strict partial order, for compliance with literature, we discuss the method in the context of partial orders. The pair $(\mathcal{F}, \preceq)$ is called a *partially ordered set* (*poset*). Furthermore, given two elements $x, y \in \mathcal{F}$, x is said to dominate y iff $x \prec y$ and both elements are said to be incomparable (denoted $x||y$) iff $x \not\prec y$ and $y \not\prec x$ [a].

In the context of noisy objective functions, one can view the noisy fitness value $\tilde{f}(\mathbf{x})$ of an individual $\mathbf{x} \in \mathcal{X}$ as an element of the interval $[f(\mathbf{x}) - a, f(\mathbf{x}) + a]$. Hence, given a noisy evaluation $\tilde{f}(\mathbf{x})$, then the true fitness $f(\mathbf{x})$ of $\mathbf{x}$ should lie within the random interval $[\tilde{f}(\mathbf{x}) - a, \tilde{f}(\mathbf{x}) + a]$.

When assuming that all feasible solutions $\mathbf{x} \in \mathcal{A}$ have been evaluated once, one can define the set of minimal elements in the set of evaluated objective function values as

$$\tilde{\mathcal{F}}^* = \{\tilde{f}(\mathbf{x}^*) \,|\, \mathbf{x}^* \in \mathcal{A} \text{ and } \nexists \mathbf{x} \in \mathcal{A} : [\tilde{f}(\mathbf{x}) - a, \tilde{f}(\mathbf{x}) + a] \prec [\tilde{f}(\mathbf{x}^*) - a, \tilde{f}(\mathbf{x}^*) + a]\}. \tag{5.36}$$

For this, Rudolph [Rud01] showed that

$$\max\{\tilde{\mathcal{F}}^*\} \leq f^* + 3a, \tag{5.37}$$

where f^* denotes the optimum of the "true" objective function. That is, all solutions in $\tilde{\mathcal{F}}^*$ lie at most $3a$ from the optimum.

[a] By abuse of notation, the notion of incomparability also includes equality.

above until the population consists of all non-dominated solutions and no improvements have been found for a number of generations. At that point, increase the sample size to tighten the confidence intervals and increase convergence accuracy.

This approach is different from other noise handling approaches in that it is the only approach that actively considers the noisy objective function values in the context of a partial order based on confidence intervals. However, comparing it to the thresholding approach (see Section 5.3.5), there is some overlap. The elitist thresholding scheme maintains non-dominated solutions in a similar way, by only accepting solutions that are a factor of τ better. In a similar fashion one could construct confidence bounds that have a similar effect as using this threshold.

The algorithm proposed in [Rud01] is not straightforwardly incorporable in the algorithmic schemes considered in this work. This is because it considers an elitist evolution loop that differs from the general evolution loop of the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES, and because it is based on the assumption that the noise is strictly bounded within known intervals $[-a, a]$. In order to test the idea of using the non-dominance relation amongst intervals, we consider the adaptive averaging procedure of Algorithm 5.1. This procedure, which replaces the evaluation procedure of the canonical $(5/2_{DI}, 35)$-σSA-ES and CMA-ES, counts the number of non-dominated solutions based on Gaussian confidence intervals (using confidence level δ) that are constructed from a number of samples $\lfloor m_{\text{eval}} + 1 \rfloor$. If this number reaches the parental population size μ (corresponding to a parental population that contains only non-dominated solutions), then the number of samples used for the next generation is increased with a factor α_m. The ranking amongst the individuals is based on the mean objective function values. When there are at most μ non-dominated solutions, then all of them will be selected when selecting based on the mean objective function values.

To gain insight in the behavior of this evaluation scheme we perform the following experiment:

Experiment 5.4.1 (Performance of poset based adaptive averaging on the noisy sphere problem)**:** We perform 10 runs of a $(5/2_{DI}, 35)$-σSA-ES (see Section 4.2.2) incorporating the evaluation procedure of Algorithm 5.1 (named PUH-$(5/2_{DI}, 35)$-σSA-ES) on the 10-dimensional noisy sphere problem (see Appendix A.1). We take parameter setting $\alpha = 1.5$ and use varying $\delta = 0.1, 0.3, 0.5$. As benchmark, we include a $(5/2_{DI}, 35)$-σSA-ES using a fixed sample size resampling scheme with $m = 50$. Each run uses a budget of $100,000$ objective function evaluations.

The results of Experiment 5.4.1 are presented in Figure 5.8 and Figure 5.9. Figure 5.8 shows the convergence dynamics of the three instances of this adaptive averaging scheme and as a benchmark the convergence dynamics of a $(5/2_{DI}, 35)$-σSA-ES using a fixed sample size resampling scheme with $m = 50$. Figure 5.9 shows the single run and average dynamics of the three instances of this adaptive averaging scheme. The left column shows the distance to the optimizer versus the number of generations, the middle column the development of sample

Algorithm 5.1: Poset Based Adaptive Averaging

Procedure parameters: confidence level δ, 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 $m = \lfloor m_{\text{eval}} + 1 \rfloor$ noisy objective function evaluations
$$\tilde{f}_{i,j} = \tilde{f}(\mathbf{x}_i)\ ,\ i = 1, \ldots, \lambda\ ,\ j = 1, \ldots, m. \tag{5.38}$$
2. For each individual $\mathbf{x}_i$, compute the mean objective function value $\bar{f}_i$, the sample variance s_i^2, and confidence bound $[\bar{f}_i - c_i, \bar{f}_i + c_i]$, with:
$$c_i = c_i^{\text{Gaussian}} = \frac{s_i}{\sqrt{m}} \Phi^{-1}\left(\frac{1+\delta}{2}\right). \tag{5.39}$$
3. Compute the number of non-dominated solutions as
$$\#\text{nds} = |\{i \in \{1, \ldots, \lambda\} | \nexists j \in \{1, \ldots, \lambda\} : f_j + c_j < f_i - c_i\}|\,. \tag{5.40}$$
4. Update the sample size m_{eval} using the update rule
$$m_{\text{eval}} = \begin{cases} \alpha \cdot m_{\text{eval}} & , \text{if } \#\text{nds} \geq \mu \\ m_{\text{eval}} & , \text{otherwise} \end{cases}. \tag{5.41}$$
5. Generate a ranking $\mathbf{x}_{1:\lambda}, \ldots, \mathbf{x}_{\lambda:\lambda}$ based on the sample means $\bar{f}_1, \ldots, \bar{f}_\lambda$.

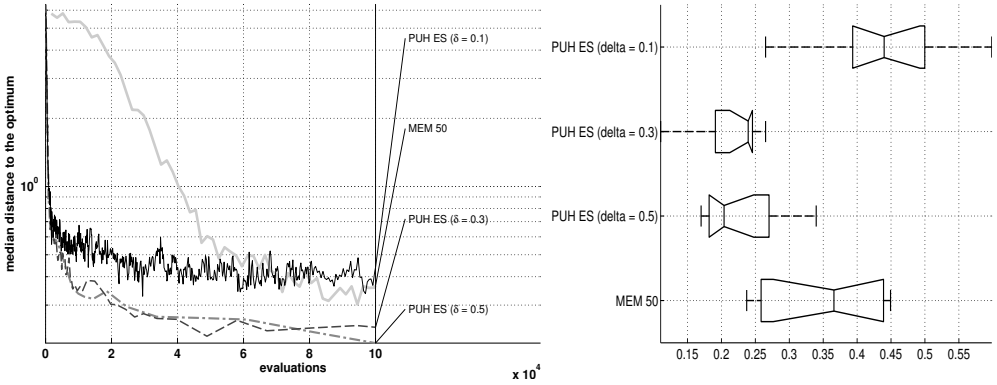

Figure 5.8: Left: The convergence dynamics (median over 10 runs) of the PUH-$(5/2_{DI}, 35)$-σSA-ES on the noisy sphere problem using $\alpha = 1.5$ and using varying $\delta = 0.1, 0.3, 0.5$, compared against a fixed sample size resampling scheme with $m = 50$. Right: boxplots of the final solution quality.

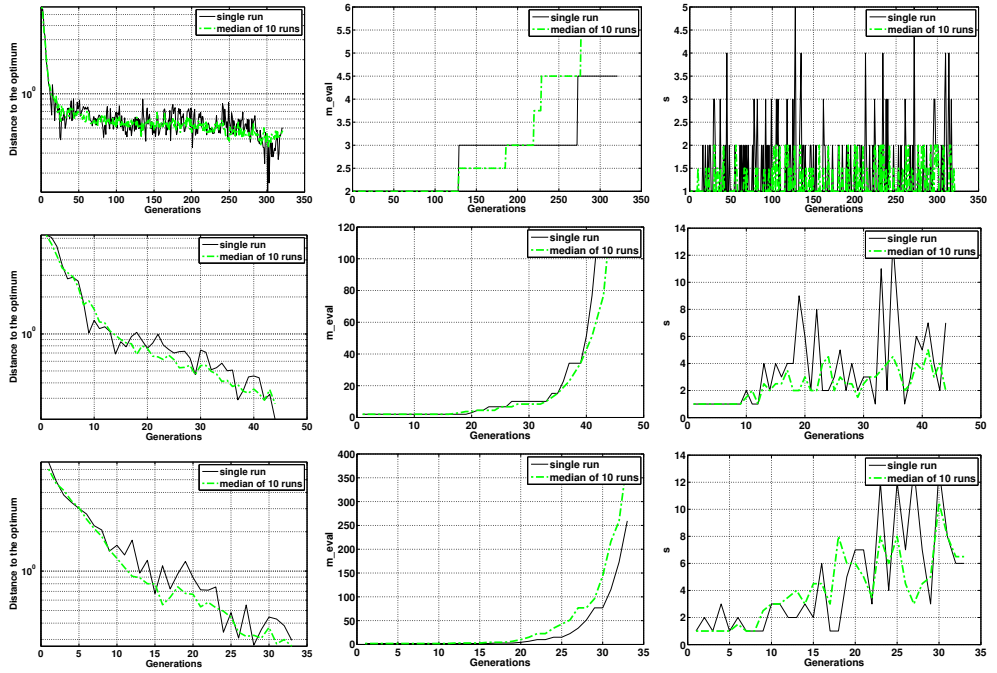

Figure 5.9: The dynamics (median over 10 runs) of the PUH-$(5/2_{DI}, 35)$-σSA-ES on the noisy sphere problem, using different confidence levels $\delta = 0.1$ (top row), $\delta = 0.3$ (middle row), $\delta = 0.5$ (bottom row). Left: the distance to the optimizer versus number of generations. Center: the development of m_{eval}. Right: the development of the uncertainty level (i.e., the number of non-dominated solutions in the offspring population).

size parameter m_{eval}, and the right column the development of the uncertainty indicator (i.e., the number of non-dominated solutions).

In Figure 5.8 we observe promising convergence behavior of this scheme for all considered settings of δ when comparing it to the fixed sample size resampling scheme. Also the convergence plots of Figure 5.9 show promising behavior, however, we observe a big difference between the three PUH-$(5/2_{DI}, 35)$-σSA-ES variants. When δ is small (e.g., $\delta = 0.1$), corresponding to having loose confidence bounds, the uncertainty level stays low for a large number of generations. Yet, from the convergence plot of $\delta = 0.1$, we also see that the hovering behavior that is due to noise already occurs after approximately 20 generations. On the other hand, the convergence plots for higher values of δ do not show this hovering behavior, but also complete much less generations based on the same evaluation budget. The latter is due to a much faster growing uncertainty level, yielding an exponentially growing sample size m_{eval}. Here, we observe a typical dilemma of adaptive averaging techniques, which is to find a good balance between requiring a high selection accuracy that yields a good progress each generation and accepting inaccuracies in the selection that yields slower generation-wise convergence, but which allows for completing much more generations with the same evaluation budget. In this small experiment setup, $\delta = 0.3$ seems to be the most promising choice.

From these results, we can conclude that using the concept of dominance based on partial orders on uncertainty intervals seems indeed a viable way to do uncertainty handling. However, the results do not show whether this approach can outperform a well chosen fixed sample size resampling scheme on a fixed evaluation budget. Also a more fine-grained tuning of this approach remains to be done.

5.4.4 Selection Through Racing

Heidrich-Meisner and Igel [HMI09a] suggest the use of so-called *Hoeffding* and *Bernstein Races* [MM94, MM97] for handling noisy fitness evaluations. They proposed their approach in the context of policy learning and incorporated it in the CMA-ES. In [HMI09a], it is stated that the goal of the noise handling scheme is to ensure with a given confidence that the μ selected individuals from the population are indeed the μ best. To achieve this, they 1) control the overall number of evaluations, and 2) control the distribution of evaluations among the individuals in the population. Note that this two-phase distinction is identical to the distinction between duration scheduling and sample allocation made by Aizawa and Wah [AW94].

The evaluation/selection procedure as proposed in [HMI09a] uses confidence bounds based on Hoeffding's or Bernstein's inequality as described in Technical Note 5.7. The underlying assumption of this approach is that the measured objective function values $\tilde{f}_{i,j}$ of a candidate solution $\mathbf{x}_i$ are bounded within known bounds $[a, b]$. Given this assumption, Hoeffding's or Bernstein's inequality can be used to construct confidence bounds for the estimate of the sample mean of each individual in the population when having multiple objective function

Technical Note 5.7: Hoeffding and Bernstein Bounds

Given m noisy evaluations of individual i, $\tilde{f}_{i,1}, \ldots, \tilde{f}_{i,m}$ and the sample mean $\bar{f}_i$ of these m samples. Furthermore, assume that the fitness value is almost surely bounded within the interval $[a, b]$, i.e., $\Pr(\tilde{f}_i \in [a, b]) \approx 1$. Using Hoeffding's inequality we can state that

$$\Pr\left(\left|\bar{f}_i - \mathbf{E}[\tilde{f}_i]\right| \geq R\right) \leq 2\exp\left(-\frac{2R^2 m}{(b-a)^2}\right). \tag{5.42}$$

Using this, we can state that with a probability of at least $1-\delta$ it holds that

$$\left|\bar{f}_i - \mathbf{E}[\tilde{f}_i]\right| \leq (b-a)\sqrt{\frac{\ln\frac{2}{\delta}}{2m}}. \tag{5.43}$$

A more general bound can be obtained by using the *empirical Bernstein bound*, which uses the empirical standard deviation, obtained through $\hat{\sigma}_i^2 = \frac{1}{m}\sum_{j=1}^{m}\left(\tilde{f}_{i,j} - \bar{f}_i\right)^2$. For this, it holds with a probability of $1-\delta$ that

$$\left|\bar{f}_i - \mathbf{E}[\tilde{f}_i]\right| \leq \hat{\sigma}_i\sqrt{\frac{2\ln\frac{3}{\delta}}{m}} + \frac{3(b-a)\ln\frac{3}{\delta}}{m}. \tag{5.44}$$

Hence, using the Hoeffding or the Bernstein inequality, we can compute a confidence interval $[\bar{f}_i - c_i, \bar{f}_i + c_i]$, with

$$c_i^{\text{Hoeffding}} = (b-a)\sqrt{\frac{\ln\frac{2}{\delta}}{2m}}, \tag{5.45}$$

$$c_i^{\text{Bernstein}} = \hat{\sigma}_i\sqrt{\frac{2\ln\frac{3}{\delta}}{m}} + \frac{3(b-a)\ln\frac{3}{\delta}}{m}. \tag{5.46}$$

Technical Note 5.8: Selection Through Races

Given for each individual $i = 1, \ldots, \lambda$ object variables $\mathbf{x}_i$, absolute lower and upper bound a and b of the fitness values, a maximal evaluation budget per individual m_{limit}, the number of to be selected individuals μ, and required confidence level δ.

1. Let $S = \emptyset$, $D = \emptyset$, and $U = \{1, \ldots, \lambda\}$ be respectively the set of selected, discarded, and undecided individuals. Evaluate each individual once, $\tilde{f}_{i,1} = f(\mathbf{x}_i)$ for $i = 1, \ldots, \lambda$, initialize the lower and upper bounds for each individual, $LB_i = a$, $UB_i = b$ for $i = 1, \ldots, \lambda$, initialize the sample counter $m = 1$, and let $u_m = |U|$.

2. Set $m = m + 1$, and let $u_m = |U|$. Then, for each of the undecided individuals $i \in U$, reevaluate once $\tilde{f}_{i,m} = f(\mathbf{x}_i)$ and recompute the sample mean $\bar{f}_i = \frac{1}{m} \sum_{j=1}^{m} \tilde{f}_{i,j}$.

3. For each undecided individual $i \in U$, compute a new confidence interval $\left[\bar{f}_i - c_i, \bar{f}_i + c_i, \right]$ using the Hoeffding or Bernstein bound requiring a confidence of $1 - \delta/n_b$, $n_b = \sum_{j=1}^{m-1} u_j + (m_{\text{limit}} - m + 1)u_m$ [a]. Update the lower and upper bound based on the new confidence interval: $LB_i = \max\{LB_i, \bar{f}_i - c_i\}$, $UB_i = \min\{UB_i, \bar{f}_i + c_i\}$.

4. For each of the undecided individuals $i \in U$:

   - If $|\{j \in U | LB_i < UB_j\}| \geq \lambda - \mu - |D|$, then individual i is probably among the best μ, so add it to the set of selected individuals $S = S \cup \{i\}$ and remove it from the set of undecided individuals $U = U \backslash \{i\}$.
   - If $|\{j \in U | UB_i < LB_j\}| \geq \mu - |S|$, then individual i is probably not among the best μ, so add it to the set of discarded individuals $D = D \cup \{i\}$ and remove it from the set of undecided individuals $U = U \backslash \{i\}$.

5. Repeat step 3 to 5 until $|S| = \mu$ or $m = m_{\text{limit}}$.

6. Adapt m_{limit}, if $|S| = \mu$ then $m_{\text{limit}} = \max\{3, (1/\alpha) \cdot m_{\text{limit}}\}$, else $m_{\text{limit}} = \min\{m_{\max}, \alpha \cdot m_{\text{limit}}\}$. Use $\{\bar{f}_1, \ldots, \bar{f}_\lambda\}$ as the fitness values for selection and m_{limit} as the updated evaluation limit.

[a] In order to assure for n estimated intervals a $1 - \delta$ confidence, we require a confidence of $1 - \delta/n$ for each individual interval (Boole's inequality). Given that $n_{b,i}$ denotes the total number of computed intervals for individual i after the full evaluation loop, then there are $n = n_{b,1} + \ldots + n_{b,\lambda}$ estimated bounds in total. As a (worst-case) estimate for n, $n_b = \sum_{j=1}^{m-1} u_j + (m_{\text{limit}} - m + 1)u_m$ can be used.

evaluations.

A procedure named *race* (see Technical Note 5.8) is proposed that incorporates an in-generation resampling loop that uses these confidence bounds. At the start of the racing procedure, a confidence bound is computed for each individual. Thereafter, within the resampling loop, the individuals that are marked *undecided* are reevaluated and their confidence bounds are updated. An individual is qualified as undecided when it does not belong to the μ best (*selected*) or the $\lambda - \mu$ worst (*discarded*) individuals. By applying resampling on the undecided individuals, the confidence bounds become tighter. The resampling loop is repeated until there are μ individuals marked as selected, meaning that there is sufficient belief (i.e., a confidence of $1-\delta$) that the μ best individuals are indeed the μ best. Or, in terms of the dominance relation on intervals proposed by Rudolph [Rud01], resampling is done on the non-dominated solutions until μ or less non-dominated solutions remain.

Within the *race* procedure, an upper evaluation limit m_{limit} for each individual is used, which is required for obtaining a finite value for n_b. If this limit is reached, the race is stopped and the best μ individuals are selected based on the sample mean. The limit is updated (i.e., increased or decreased with a factor α) each generation of the evolution cycle, based on whether the full budget m_{limit} is used. Besides that, an absolute evaluation limit $m_{\max}$ is included to prevent the sample size from getting too large.

As alternative for the *race* procedure Heidrich-Meisner [HM11] proposed the so-called ϵ-*race* procedure. In this adapted version, the limits m_{limit} and $m_{\max}$ are not required. Instead of resampling each of the undecided individuals once each racing step, each individual is reevaluated $\theta(t)-\theta(t-1)$ times in the tth racing step, with $\theta(t) = t^2$. Besides that, the required confidence for the nth computed bound is set to $\delta_n = c\delta/n^2$, with $c = 6/\pi^2$. Incorporating these changes removes the need for a fixed maximum race length. Finally, the notion of ϵ-similarity condition was introduced in order to avoid long races. That is, the racing-loop is stopped when the difference between the highest upper bound and the lowest lower bound of the individuals that are still undecided drops below a certain threshold ϵ.

A downside of using Bernstein and Hoeffding bounds is that these require known bounds for the objective function values. This means that either the noise should be bounded within known bounds or the bounds of the objective function should be known. If both are unknown, which is the scenario that we consider in this work, then these bounds should be estimated or the objective function should be transformed as described in [HM11, p. 112].

A more serious issue is that the Hoeffding and Bernstein bounds, as used in the racing procedures, are based on assumed bounds on the objective function and not (or hardly) on the measured noise. For instance, when using the Hoeffding bound in a racing procedure, it does not matter whether the noise is very small or very high, the Hoeffding bound is only based on the assumed bounds on the full domain of the objective function. The Bernstein bound does consider the sample variance, but still contains a large factor (the rightmost term in Eq. 5.46)

that is not based on it. This is conceptually very undesirable and in this work a reason not to consider this noise handling technique as suitable option.

Although studying modifications of these techniques falls beyond the scope of this work, two options could be tried to fix the aforementioned problems. First, the racing procedures could be adapted when the noise is bounded within known bounds (i.e., $z \in [a, b]$). In that case, the bounds for each individual $i = 1, \ldots, \lambda$ could be initialized separately, based on one evaluation of the objective function, yielding possibly much tighter bounds. Second, for Gaussian noise, an alternative to using Bernstein and Hoeffding bounds is to use the more classical type of confidence bounds within the same selection procedure. Given the selection procedure based on races (Technical Note 5.8), an alternative would therefore be to use Gaussian confidence intervals, as considered by Rudolph [Rud01] (see Eq. 5.39).

5.4.5 Rank-Change Based Uncertainty Measures

Hansen et al. [HNGK09] propose a scheme for handling noisy objective functions implemented within the CMA-ES; the *Uncertainty Handling* CMA-ES (UH-CMA-ES). Although originally proposed in the context of an application to feedback control of combustion, the main concepts can be applied in more general scenarios, as shown by Heidrich-Meisner and Igel [HMI09b]. Moreover, the uncertainty handling scheme can be applied within any rank-change based optimization algorithm.

The uncertainty handling scheme separates two components: the quantification of the uncertainty and the treatment of the uncertainty. That is, the evaluation intensity/accuracy is increased or decreased based on measurements of the impact of the noise on the selection.

The **uncertainty quantification** is based on counting the number of rank-changes that occur when reevaluating (a part of) the population. If the number of rank-changes after reevaluation is high, then it can be assumed that the uncertainty is high and the noise should be reduced. If there are only few rank-changes, then the uncertainty is low and a higher noise level may be allowed. The procedure is described in detail in Technical Note 5.9.

The **uncertainty treatment** scheme used in [HNGK09] consists of two methods: 1) increasing the evaluation time t_{eval}, and 2) increasing the population variance by increasing the stepsize. The former is specifically suitable for the problem considered in [HNGK09], as it is possible to increase the accuracy of the fitness function by increasing the measuring time. The latter is a secondary treatment method, used when the evaluation time has reached its maximum $t_{\max}$. For the uncertainty treatment, as suggested in [HNGK09], a cumulated version $\bar{s}$ of the uncertainty measure s is introduced, updated every generation using $\bar{s} = (1 - c_s)\bar{s} + c_s s, c_s \in [0, 1]$. Whenever $\bar{s}$ is greater than zero, the evaluation time is increased with a factor of α_t.

For selection, the solutions are re-ranked according to their rank sum: $\mathrm{rank}\,(L_i^{\mathrm{new}}) + \mathrm{rank}\left(L_i^{\mathrm{old}}\right)$. Ties are resolved firstly using the absolute rank change $|\Delta_i|$, using for the not reevaluated solutions: $\Delta_i = (1/\lambda_{\mathrm{reev}}) \sum_{j=1}^{\lambda_{\mathrm{reev}}} |\Delta_j|$, secondly using the sample mean.

Technical Note 5.9: Rank-Change Based Uncertainty Quantification

For each solution $\mathbf{x}_i$, $i = 1, \ldots, \lambda$ an approximation of its fitness is obtained, i.e.,

$$L_i^{\text{old}} = f(\mathbf{x}_i). \tag{5.47}$$

Then, a parameter λ_{reev} is computed using the parameter r_λ (recommended $r_\lambda = 0.3$), with $\lambda_{\text{reev}} = f_{\text{pr}}(r_\lambda \cdot \lambda)$, where the function $f_{\text{pr}} : \mathbb{R} \to \mathbb{Z}$ is defined as

$$f_{\text{pr}}(x) = \begin{cases} \lfloor x \rfloor + 1 & \text{with probability } x - \lfloor x \rfloor \\ \lfloor x \rfloor & \text{otherwise} \end{cases}. \tag{5.48}$$

Furthermore, λ_{reev} is set to 1 whenever it has been set to 0 for more than $2/(r_\lambda \cdot \lambda)$ consecutive generations. The first λ_{reev} solutions are selected for reevaluation, i.e.,

$$L_i^{\text{new}} = \begin{cases} f(\mathbf{x}_i) & \text{if } i \leq \lambda_{\text{reev}} \\ L_i^{\text{old}} & \text{otherwise} \end{cases}. \tag{5.49}$$

For each solution $\mathbf{x}_i$, the rank change Δ_i is computed as

$$\Delta_i = \text{rank}(L_i^{\text{new}}) - \text{rank}(L_i^{\text{old}}) - \text{signum}\left(\text{rank}(L_i^{\text{new}}) - \text{rank}(L_i^{\text{old}})\right), \tag{5.50}$$

where $\text{rank}(L_i)$ is the rank of function value L_i in the set $\mathcal{L} = \{L_k^{\text{old}}, L_k^{\text{new}} | k = 1, \ldots, \lambda\}$. Hence, Δ_i counts the number of values from the set $\mathcal{L} \backslash \{L_i^{\text{old}}, L_i^{\text{new}}\}$ that lie between L_i^{old} and L_i^{new}. Based on the individual rank-changes, the uncertainty level s is determined as

$$\begin{aligned} s \quad = \quad & \frac{1}{\lambda_{\text{reev}}} \sum_{i=1}^{\lambda_{\text{reev}}} \left(2|\Delta_i| \right. \\ & -\Delta_\theta^{\text{lim}} \left(\text{rank}\left(L_i^{\text{new}}\right) - \mathbb{I}\{L_i^{\text{new}} > L_i^{\text{old}}\}\right) \\ & \left. - \Delta_\theta^{\text{lim}} \left(\text{rank}\left(L_i^{\text{old}}\right) - \mathbb{I}\{L_i^{\text{old}} > L_i^{\text{new}}\}\right)\right), \end{aligned} \tag{5.51}$$

where $\Delta_\theta^{\text{lim}}(R)$ is the $\theta \times 50\%$ percentile of the set $\{|1 - R|, |2 - R|, \ldots, |2\lambda - 1 - R|\}$. It represents the rank change for a given rank R that would occur when given a completely random function and is a reference for Δ_i. The indicator function $\mathbb{I}$ returns 1 if its argument is true, otherwise 0.

Algorithm 5.2: Rank-Change Based Adaptive Averaging

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 $m_1 = \lceil m_{\text{eval}}/2 \rceil$ noisy objective function evaluations

$$\tilde{f}_{i,j} = \tilde{f}(\mathbf{x}_i)\ ,\ i = 1, \ldots, \lambda\ ,\ j = 1, \ldots, m_1, \tag{5.52}$$

   compute the mean objective function value $\bar{f}_{i,\text{old}}$ for each individual $i = 1, \ldots, \lambda$ based on this sample set, and store them in the set $L^{\text{old}} = \{\bar{f}_{1,\text{old}}, \ldots, \bar{f}_{\lambda,\text{old}}\}$.

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

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

$$\Delta_i = \text{rank}(L_i^{\text{new}}) - \text{rank}(L_i^{\text{old}}) - \text{signum}\left(\text{rank}(L_i^{\text{new}}) - \text{rank}(L_i^{\text{old}})\right). \tag{5.53}$$

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

$$\begin{aligned} s \;=\; & \frac{1}{\lambda_{\text{reev}}} \sum_{i=1}^{\lambda_{\text{reev}}} \big(2|\Delta_i| \\ & -\Delta_\theta^{\lim}\left(\text{rank}\left(L_i^{\text{new}}\right) - \mathbb{I}\{L_i^{\text{new}} > L_i^{\text{old}}\}\right) \\ & -\ \Delta_\theta^{\lim}\left(\text{rank}\left(L_i^{\text{old}}\right) - \mathbb{I}\{L_i^{\text{old}} > L_i^{\text{new}}\}\right)\big). \end{aligned} \tag{5.54}$$

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

$$m_{\text{eval}} = \begin{cases} \alpha \cdot m_{\text{eval}} & ,\text{if } s > 0 \\ m_{\text{eval}} & ,\text{otherwise} \end{cases}. \tag{5.55}$$

6. Generate a ranking $\mathbf{x}_{1:\lambda}, \ldots, \mathbf{x}_{\lambda:\lambda}$ based on the sample means $(\bar{f}_{1,\text{old}} + \bar{f}_{1,\text{new}})/2, \ldots, (\bar{f}_{\lambda,\text{old}} + \bar{f}_{\lambda,\text{new}})/2$.

In this work we consider the algorithm as described in Algorithm 5.2 as an implementation of the rank-change based uncertainty handling scheme. It is an adapted version of the approach proposed in [HNGK09], which is done to allow for resampling, but also to make it such that this scheme differs from Algorithm 5.1 only in the uncertainty indicator. The latter is done for the sake of comparison of the two approaches.

Two issues that are deliberately left out are the decrement of m_{eval} when the uncertainty level is small (i.e., $s < 0$), and the upper limit on the sample size. The former is done under the assumption that the sample size should only increase during an optimization run. The latter is done in order test the uncertainty handling mechanism itself, eliminating side-effects that are due to sample size limits. To obtain an insight in the behavior of this rank-change based uncertainty handling scheme we perform the following experiment:

Experiment 5.4.2 (Performance of rank based adaptive averaging on the noisy sphere problem)**:** We perform 10 runs of a $(5/2_{DI}, 35)$-σSA-ES (see Section 4.2.2) incorporating the evaluation procedure of Algorithm 5.2 (named UH-$(5/2_{DI}, 35)$-σSA-ES) on the 10-dimensional noisy sphere problem (see Appendix A.1). We take parameter setting $\alpha = 1.5$ and use varying $\theta = 0.5, 0.7, 0.9$. As benchmark, we include a $(5/2_{DI}, 35)$-σSA-ES using a fixed sample size resampling scheme with $m = 50$. Each run uses a budget of $100,000$ objective function evaluations.

The results of Experiment 5.4.2 are presented in Figure 5.10 and Figure 5.11. Figure 5.10 shows the convergence dynamics of the three instances of this adaptive averaging approach, and as a benchmark the dynamics of a $(5/2_{DI}, 35)$-σSA-ES using a fixed sample size resampling scheme with $m = 50$. Figure 5.11 shows the single run and average dynamics of the three instances of this adaptive averaging scheme. The left column shows the distance to the optimizer versus the number of generations, the middle column the development of sample size parameter m_{eval}, and the right column the development of the uncertainty indicator.

The results of Experiment 5.4.2 are similar to the results of Experiment 5.4.1. As can be seen in Figure 5.11, increasing the strictness of the uncertainty indicator yields a quicker growth of the uncertainty level and the sample size, allowing for fewer generations. Also in this case there is a trade-off between allowing uncertainty and depending on the averaging effects of multiple generations or requiring a strict confidence in order to obtain a high progress rate per generation.

5.4.6 Rank-Inversions Based Adaptive Averaging

An alternative to the rank-change based uncertainty measure (see Eq. 5.51 in Technical Note 5.9) is to count rank inversions [Mar01, KEB09a]. For this measure, the distribution is known, and normal for $\lambda \to \infty$. This allows for a better founded measure of uncertainty which could be argued to be simpler to implement as compared to the rank-change based measure. Technical

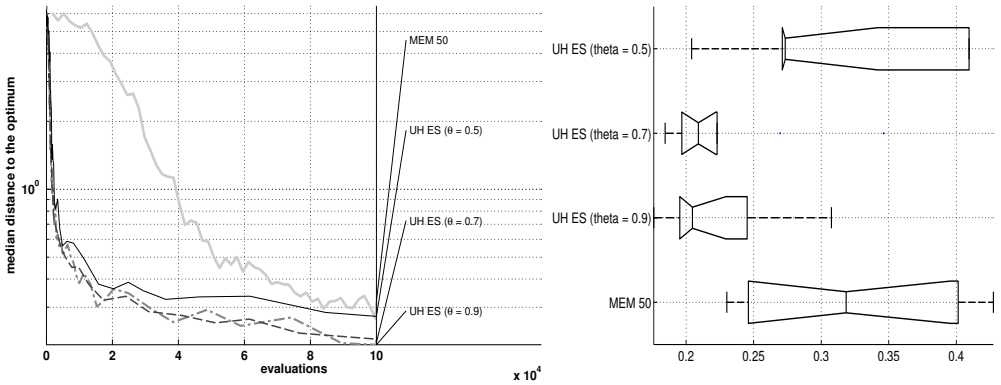

Figure 5.10: Left: The convergence dynamics (median over 10 runs) of the UH-$(5/2_{DI}, 35)$-σSA-ES on the noisy sphere problem using $\alpha = 1.5$ and using varying $\theta = 0.5, 0.7, 0.9$, compared against a fixed sample size resampling scheme with $m = 50$. Right: boxplots of the final solution quality.

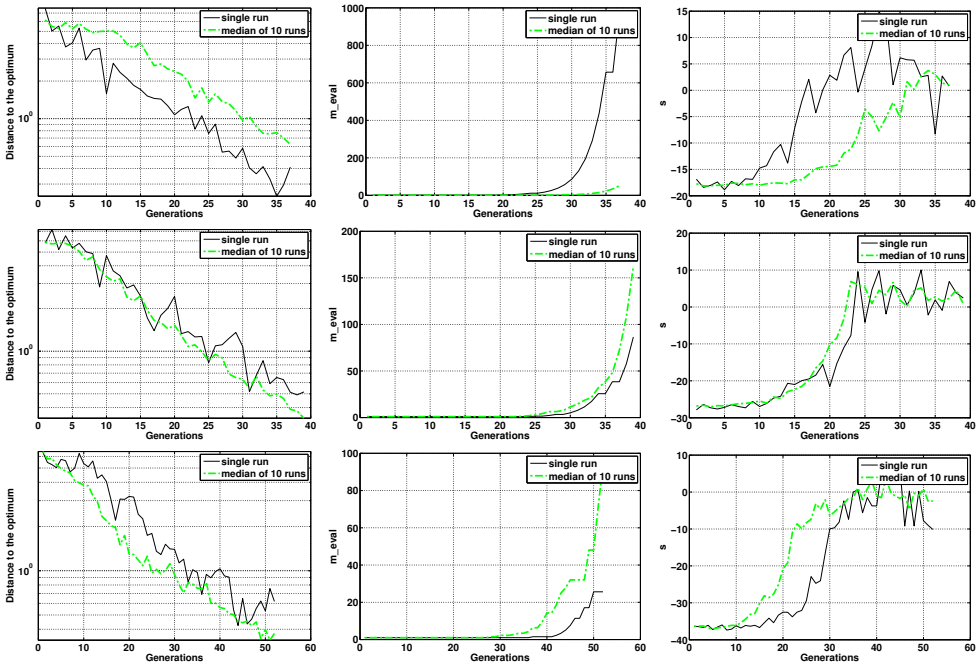

Figure 5.11: The dynamics (median over 10 runs) of the UH-$(5/2_{DI}, 35)$-σSA-ES on the noisy sphere problem, using different confidence levels $\theta = 0.5$ (top row), $\theta = 0.7$ (middle row), $\theta = 0.9$ (bottom row). Left: the distance to the optimizer versus number of generations. Center: the development of m_{eval}. Right: the development of the uncertainty level (i.e., the rank-change based indicator).

Technical Note 5.10: Rank-Inversion Based Uncertainty Measure

Given a population of λ candidate solutions, let $\mathrm{rank}_i^{\mathrm{old}}$ denote the rank of solution i based on the fitness values of the first evaluation step and let $\mathrm{rank}_i^{\mathrm{new}}$ denote the rank of solution i based on the fitness values of the reevaluation step. The number of inversions is computed as

$$\mathrm{Inv}_\lambda = \sum_{(i,j)\in\{1,\ldots,\lambda\}^2} I(i,j), \tag{5.56}$$

$$I(i,j) = \begin{cases} 0 & \text{, if } (\mathrm{rank}_i^{\mathrm{old}} < \mathrm{rank}_i^{\mathrm{old}}) \wedge (\mathrm{rank}_i^{\mathrm{new}} > \mathrm{rank}_i^{\mathrm{new}}) \\ 1 & \text{, otherwise} \end{cases}. \tag{5.57}$$

Let ξ_λ denote a random variable measuring the number of inversions for a pure random ordering. The number of inversions for randomly generated perturbations follows a normal distribution for $\lambda \to \infty$ with mean $\mathbf{E}[\xi_\lambda] = \lambda(\lambda-1)/4$ and its variance is $\mathrm{Var}[\xi_\lambda] = (2\lambda^3 + 3\lambda^2 - 5\lambda)/72$ (see, [Mar01]).

Using this, an uncertainty measure s can be constructed as

$$s_{\mathrm{Inv}} = \mathrm{Inv}_\lambda - \left(\mu_{\mathrm{Inv}} + \sigma_{\mathrm{Inv}} \cdot \Phi^{-1}(\theta)\right), \tag{5.58}$$

with $\mu_{\mathrm{Inv}} = \lambda(\lambda-1)/4$, $\sigma_{\mathrm{Inv}} = \sqrt{(2\lambda^3 + 3\lambda^2 - 5\lambda)/72}$, and $\Phi^{-1}(.)$ being the inverse cumulative distribution function of the standard normal distribution.

Note 5.10 describes how this measure can be used to obtain a similar, but alternative uncertainty measure for the rank-change based uncertainty handling method of Algorithm 5.2.

In order to test this alternative uncertainty measure, we consider it to be incorporated in the procedure of Algorithm 5.2 and use it instead of the rank-change based uncertainty measure. For this adapted scheme we run the following experiment:

Experiment 5.4.3 (Performance of inversions based adaptive averaging on the noisy sphere problem)**:** We perform 10 runs of a $(5/2_{DI}, 35)$-σSA-ES (see Section 4.2.2) incorporating the evaluation procedure of Algorithm 5.2 that uses the alternative uncertainty measure of Technical Note 5.10. This scheme is named IUH-$(5/2_{DI}, 35)$-σSA-ES, and the experiments are performed on the 10-dimensional noisy sphere problem (see Appendix A.1). We take parameter setting $\alpha = 1.5$ and use varying $\theta = 0.3, 0.5, 0.7$. As benchmark, we include a $(5/2_{DI}, 35)$-σSA-ES using a fixed sample size resampling scheme with $m = 50$. Each run uses a budget of $100,000$ objective function evaluations.

The results of Experiment 5.4.3 are presented in Figure 5.12 and Figure 5.13. Figure 5.12 shows the convergence dynamics of the three instances of this adaptive averaging scheme compared to the convergence dynamics of a $(5/2_{DI}, 35)$-σSA-ES using a fixed sample size resampling scheme with $m = 50$. Figure 5.13 shows the single run and average dynamics of the three instances of this adaptive averaging scheme. The left column shows the distance to

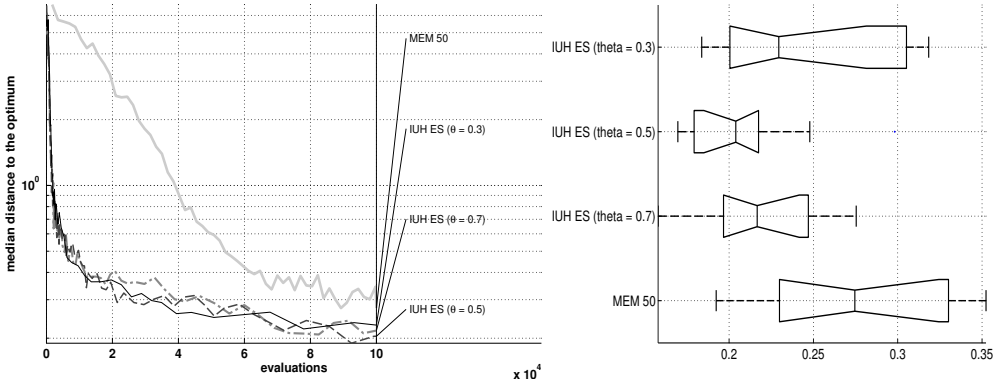

Figure 5.12: Left: The convergence dynamics (median over 10 runs) of the IUH-$(5/2_{DI}, 35)$-σSA-ES on the noisy sphere problem using $\alpha = 1.5$ and using varying $\theta = 0.3, 0.5, 0.7$, compared against a fixed sample size resampling scheme with $m = 50$. Right: boxplots of the final solution quality.

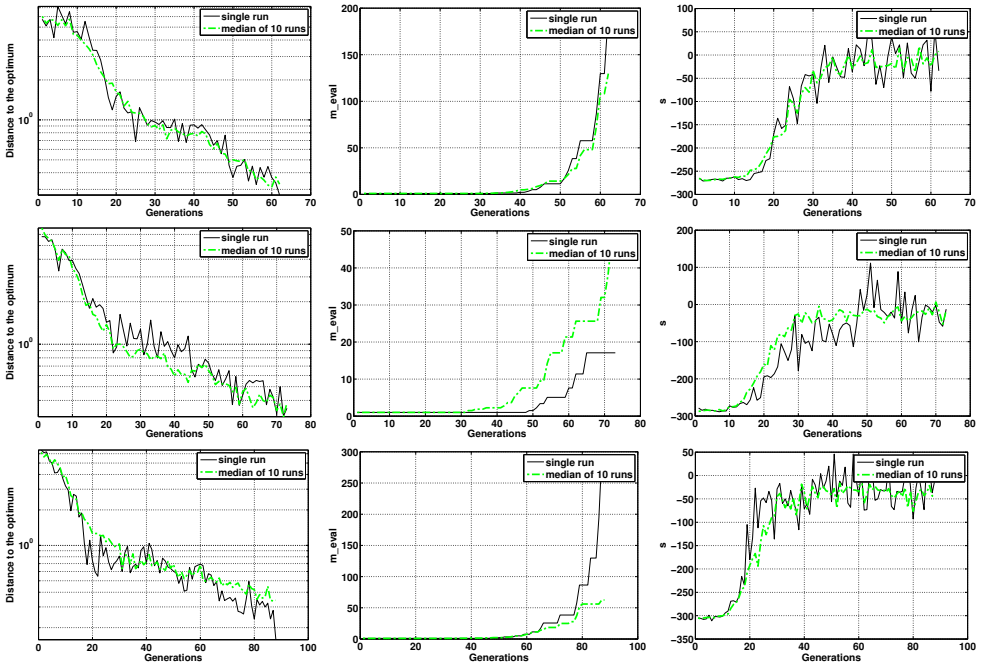

Figure 5.13: The dynamics (median over 10 runs) of the IUH-$(5/2_{DI}, 35)$-σSA-ES on the noisy sphere problem, using different confidence levels $\theta = 0.3$ (top row), $\theta = 0.5$ (middle row), $\theta = 0.7$ (bottom row). Left: the distance to the optimizer versus number of generations. Center: the development of m_{eval}. Right: the development of the uncertainty level (i.e., the inversions based indicator).

the optimizer versus the number of generations, the middle column the development of sample size parameter m_{eval}, and the right column the development of the uncertainty indicator.

The results of Experiment 5.4.3 are similar to the results of Experiment 5.4.2 of the rank-change based uncertainty handling approach. As can be seen in Figure 5.13, increasing the strictness of the uncertainty indicator yields a quicker growth of the uncertainty level and the sample size, allowing for fewer generations. The uncertainty indicator is, however, not so strict as the rank-change based uncertainty measure when using the same value for θ, though from these results this seems to be the only difference.

5.4.7 A Discussion on Adaptive Noise Handling Techniques

Comparing the adaptive averaging techniques discussed in this chapter, we observe the following:

Uncertainty quantification and uncertainty treatment: All schemes either implicitly or explicitly distinguish between uncertainty quantification and uncertainty handling. The term *uncertainty quantification* regards the decision mechanism that determines whether or not to increase the evaluation accuracy of the underlying noise treatment mechanism. The term *noise treatment* refers to the underlying noise handling mechanism. All except one of the adaptive averaging techniques that have been discussed perform uncertainty treatment using resampling (or explicit averaging). Interestingly, the alternative of increasing the population size is not considered in any of the adaptive averaging techniques.

In-generation and inter-generation mechanisms: There are two different types adaptation mechanisms; *in-generation* and *inter-generation* mechanisms. In-generation mechanisms are used in, e.g., the sample allocation scheme, t-test based adaptive resampling, and the races based approaches. Here, the uncertainty is targeted directly by continuing the resampling procedure until the uncertainty level is sufficiently reduced. In inter-generation mechanisms, present in, e.g., the duration scheduling and rank-change based uncertainty handling mechanism, the uncertainty treatment is adapted after each generation based on the previous uncertainty quantification. Inter-generation methods are based on trusting the evolution process to be partially robust against disturbances in the selection that are higher than the desired level indicated by the uncertainty quantification as long as the evaluation intensity of the following generations is increased. Although in-generation methods are more direct, an advantage of using inter-generation methods is that these are less sensitive to scenarios as observed in the t-test based approach where many samples are spend on trying to distinguish between two solutions while their difference might be of no importance in the perspective of the current stage of the optimization. Note that both mechanisms can be used within the same method. For instance, the races based approach uses both mechanisms.

Evaluation intensity limitations: Most adaptive averaging methods incorporate an absolute upper limit for the evaluation intensity. For instance, the races based approach, or the rank-

change based adaptive averaging method use a maximum sampling limit and a maximum evaluation time respectively. Moreover, a scheme in which the sample size is allowed to grow without limit, being the scheme discussed with the t-test based approach, suffers from a rapid sample size explosion. Apparently there are practical reasons for bounding the evaluation intensity. However, using such bounds is undesirable from a theoretical perspective, because it introduces limitations on the convergence accuracy.

Underlying assumptions: When looking at the assumptions on which the uncertainty handling methods are based (e.g., the type of noise), we see that there are only two uncertainty indicators that are not based on specific assumptions regarding the noise or the objective function; the rank-change and rank-inversions based uncertainty measure. Besides that, the assumption that the noise is Gaussian and the assumption of having the ability to establish confidence bounds take in a prominent place.

Parameters: The in-generation adaptive averaging schemes that were discussed require at least one parameter, namely an uncertainty quantification threshold (e.g., a confidence level). Inter-generation adaptive averaging mechanisms require at least two parameters, namely an uncertainty quantification threshold and a scaling factor for the evaluation intensity (e.g., a growth factor for the sample size in explicit averaging). Moreover, when including upper limits on the evaluation intensity or cumulation of the uncertainty quantification, this number increases rapidly. In the methods summarized so far, the parameters were mostly set based on empirical testing.

In conclusion, Table 5.2 shows the different adaptive averaging techniques, summarized with respect to the assumptions on which they are based, the used uncertainty quantification measure, and the used uncertainty handling method. Considering the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES, the partial order based adaptive averaging (*PUH*, Section 5.4.3), rank-change based adaptive averaging (*UH*, Section 5.4.5), and rank-inversions based adaptive averaging or (*IUH*, Section 5.4.6) provide the most promising alternatives to explicit and implicit averaging. In the remainder of this work, these schemes will be studied in more depth for their practical applicability. Two essential, but yet unanswered questions are:

- For each uncertainty quantification measure, at what uncertainty level do Evolution Strategies still have a positive expected progress and which uncertainty level is optimal?
- For inter-generation adaptive averaging methods, at what rate should the evaluation intensity increase in order to allow Evolution Strategies to progress?

5.5 Metamodel Assisted Noise Handling

Another class of techniques for dealing with noisy objective functions in the context of Evolutionary Algorithms is formed by techniques that construct a surrogate model (or metamodel) of

Noise handling scheme	Assumptions	Uncertainty quantification	Uncertainty treatment
Duration scheduling [AW93, AW94]	The "real" underlying fitness values of the population are normally distributed and the noise is Gaussian	Ratio between population variance and approximation error	Resampling
Sample allocation [AW93, AW94]	The "real" underlying fitness values of the population are normally distributed and the noise is Gaussian	The probability of each individual to be the best	Resampling
t-Test based resampling [Sta98, CP04, KEB09a]	Gaussian noise	Ranking confidence among all/some pairs of individuals based on the t-test	Resampling
Partial order based selection [Rud01]	The noise is bounded or Gaussian noise is assumed, utilizing confidence bounds	Non-dominance ranking on interval orders / acceptance threshold	Resampling
Races based uncertainty handling [HMI09a]	The objective function values are bounded within known bounds	Ranking confidence among all individuals based on the Hoeffding or Bernstein bound	Resampling
Rank-change based uncertainty handling [HNGK09]	No assumptions	Rank changes after reevaluation of (part of) the individuals in the population	Increasing evaluation time
Rank-inversions based uncertainty handling	No assumptions	Rank inversions after reevaluation of the individuals in the population	Resampling

Table 5.2: The different adaptive averaging techniques, summarized with respect to the assumptions which they are based on, the uncertainty quantification measure that is used, and the uncertainty handling method that is used.

the underlying objective function based on the full history of noisy measurements, and perform optimization on this model. Such approaches aim to use all information that is available about the objective function. Methods that are based on this idea were proposed by Sano and Kita [SK00, SK02], and Branke et al. [BSS01]. Though these schemes are not the main focus of this work, this section will provide a brief technical summary.

5.5.1 Memory-Based Fitness Estimation

In [SK00, SK02], Sano and Kita propose an approach named Memory-Based Fitness Estimation (MFE) (see Technical Note 5.11). The core idea of the approach is that the objective function value of candidate solution $\mathbf{x}^*$, given fitness observation $\tilde{f}^*$, can be estimated by means of a maximum likelihood approach based on Gaussian model assumptions and the assumption that there is a spatial relation between $\mathbf{x}^*$ and an archive of previous objective function measurements $A = \{(\mathbf{x}_i, \tilde{f}_i) \,|\, i = 1, \ldots, L\}$. In the approach, the fitness of every individual in the population is computed following this method. The archive is in this approach filled with objective function measurements of the previous populations.

In [SK02], an adaptation is proposed to account for situations where the objective function estimate of a candidate solutions differed too much from the measured objective function value. In order to prevent these cases, it is proposed to reject the candidate solutions in a population of which the measured objective function values differed more than a threshold Z from the best measured objective function value. In [SK02], the threshold Z is recommended to be set such that the probability of such errors to occur is less than 0.3.

5.5.2 Local Regression Based Fitness Estimation

Branke et al. [BSS01] propose a similar approach, only they consider a model that is based on different assumptions as compared to the approach of Sano and Kita [SK00, SK02]. In their approach, they consider stationary Gaussian noise and optimization of the real underlying objective function.

Technical Note 5.12 describes the general approach proposed in [BSS01]. The modeling assumption is that the objective function can be locally approximated by a low order polynomial function. They propose to estimate the objective function value of each candidate solution by building a local model based on an archive of previously evaluated points, weighting each archive solution based on the distance to the to-be-evaluated solution. The choices that remain open are to determine the degree of the polynomial used for regression, the choice for the neighborhood parameter h that assigns a weighting contribution for each archive solution based on the distance to the to-be-evaluated solution, and the way in which the regression model is fitted. In [BSS01], a quadratic model is used. As an extension, it is also suggested to use the information of the local models more extensively by performing local hill-climbing on the model to locally improve candidate solutions.

Technical Note 5.11: Memory-Based Fitness Estimation

Assume that the objective function value of each candidate solution is normally distributed and consider the following modeling assumption:

$$f_j \sim \mathcal{N}(f_i, kd_{ij}), \tag{5.59}$$
$$\tilde{f}_j \sim \mathcal{N}(f_i, kd_{ij} + \sigma_\epsilon^2). \tag{5.60}$$

Here, f_i and f_j denote the true objective function values of candidate solutions $\mathbf{x}_i$ and $\mathbf{x}_j$, k is some constant, and d_{ij} denotes the distance d_{ij} between $\mathbf{x}_i$ and $\mathbf{x}_j$ (i.e., $d_{ij} = ||\mathbf{x}_i - \mathbf{x}_j||$). Hence, the true objective function value at $\mathbf{x}_j$ is assumed to be distributed normally random around the true objective function value of $\mathbf{x}_i$, proportional to the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$.

Given known values of k and σ_ϵ^2, a maximum likelihood estimation approach can be used to estimate the true objective function value f^* of candidate solution $\mathbf{x}^*$ based on its own objective function measurement $\tilde{f}^*$ and an archive of previously observed measurements $A = \{(\mathbf{x}_i, \tilde{f}_i), i = 1, \ldots, L\}$. When given f^*, the probability of obtaining $\tilde{f}_1, \ldots, \tilde{f}_L$ is expressed by

$$\prod_{i=1}^{L} p(\tilde{f}_i, d_i), \text{ where } p(\tilde{f}_i, d_i) = \frac{1}{\sqrt{2\pi(kd_i + \sigma_\epsilon^2)}} \exp\left(-\frac{1}{2} \frac{\left(\tilde{f}_i - f^*\right)^2}{kd_i + \sigma_\epsilon^2} \right). \tag{5.61}$$

Here, d_i denotes the distance between $\mathbf{x}^*$ and $\mathbf{x}_i$. One can maximize this expression for f^*, from which we obtain

$$\hat{f}^* = \frac{\tilde{f}^* + \sum_{i=1}^{L} \frac{\sigma_\epsilon^2}{kd_i + \sigma_\epsilon^2} \tilde{f}_i}{1 + \sum_{i=1}^{L} \frac{\sigma_\epsilon^2}{kd_i + \sigma_\epsilon^2}}. \tag{5.62}$$

The model parameters k and σ_ϵ^2 can be estimated using Eq. 5.61 by maximization of the log-likelihood

$$\underset{k, \sigma_\epsilon^2}{\operatorname{argmax}} \left\{ -\frac{1}{2} \left(L \log 2\pi + \sum_{i=1}^{L} \log(kd_i + \sigma_\epsilon^2) + \sum_{i=1}^{L} \frac{(\tilde{f}_i - f^*)^2}{kd_i + \sigma_\epsilon^2} \right) \right\}. \tag{5.63}$$

That is, these estimates are taken from the perspective of one candidate solution $\mathbf{x}^*$, with true objective function value f^*. In [SK00, SK02], the best individual of the current population is recommended to be used for this estimation procedure and its true fitness is recommended to be estimated by averaging the objective function values of the five closest individuals. A hill-climbing method acting on a logarithmic space of k and σ_ϵ^2 should be used for maximization of the log-likelihood[a].

[a] In [SK02], an exact derivation for the maximum log-likelihood was suggested with respect to σ_ϵ^2, but this derivation is incorrect.

Technical Note 5.12: Local Regression Based Fitness Estimation

Assume that the fitness of individual i is normally distributed, i.e.,

$$\tilde{f}_i \sim \mathcal{N}(f_i, \sigma_\epsilon^2). \tag{5.64}$$

Furthermore, assume that the "real" underlying objective function $f(\mathbf{x})$ can be locally approaximated by means of a low order polynomial function.

For a candidate solution $\mathbf{x}^*$, construct a locally weighted regression model $g_{\mathbf{x}^*,h}(\mathbf{x})$ based on an archive of previously observed measurements $A = \{(\mathbf{x}_i, \tilde{f}_i), i = 1, \ldots, L\}$ and use as objective function approximation:

$$\hat{f}_{\exp_i} = g_{\mathbf{x}_i,h}(\mathbf{x}^*). \tag{5.65}$$

The locally weighted regression model $g_{\mathbf{x}_i,h}(\mathbf{x})$ is based on a weight function $w_h(d)$, assigning a weight to the contribution of each archive point based on the Euclidean distance d between the archive point and $\mathbf{x}^*$. The weight function considered in [BSS01] is the tri-cube function

$$w_h(d) = \begin{cases} (1 - d^3/h)^3 & ,\text{if } d < h \\ 0 & ,\text{otherwise} \end{cases}. \tag{5.66}$$

The parameter h is called the neighborhood parameter and reflects the size of the neighborhood for which the assumptions can be considered to be valid.

For setting h, two methods are considered: 1) choosing it such that 5% of the archive points are considered to be neighboring points, 2) choosing h such that the following cross-validation criterion is minimized (computed using a numerical hill-climber):

$$CV(\hat{g}_{\mathbf{x}*,h}) = \frac{\sum_{i=1}^{L} w_h(d_i) \left(\tilde{f}_i - \hat{g}^{-i}_{\mathbf{x}*,h}\right)^2}{\sum_{i=1}^{L} w_h(d_i)}. \tag{5.67}$$

.

5.5.3 A Discussion on Metamodeling Noise Handling Techniques

The validity of these metamodeling approaches depends on three key issues: 1) the extent to which the modeling assumptions are valid, 2) the quality of the archive with respect to the to-be-estimated objective function value, and 3) the accuracy of the tuning of the model parameters. Moreover, it is well-known that for higher dimensional search spaces, metamodeling becomes increasingly more difficult due to the *curse of dimensionality* (see, e.g., [FSK08, p. xvii]).

Regarding the validity of the modeling assumptions, we note that this depends purely on the optimization problem at hand. Assuming a Gaussian spatial correlation, as done by Sano and Kita [SK00, SK02], is not uncommon and used, for instance, also in Kriging (see, e.g., [SWMW89, JSW98, FSK08]). The same holds for the assumption that the objective function landscape can be locally approximated using a polynomial model, as done by Branke et al. [BSS01].

The quality of the archive is in this context an issue of a different kind. For the estimation of the objective function for a given candidate solution, ideally, the archive should contain data points that lie well-spread around the given candidate solution. However, when the archive is straightforwardly built up as the history of candidate solutions obtained by the evolutionary process of an Evolutionary Algorithm itself, this is not necessarily achieved. See [KEB10] for an example where straightforward utilization of an archive within an Evolutionary Algorithm is outperformed by a more careful archive maintenance approach. The approaches discussed here do not actively try to maintain a "proper" archive with respect to the to-be-estimated solution qualities, yet the improvements proposed in [SK02] are based on negative effects that can be related to a poor archive quality.

The accuracy of the tuning of the model parameters of the MFE approach of [SK00, SK02] is questionable. That is, these parameters are estimated from the perspective of the observed best candidate solution, using a crude estimate of its real objective function value. The approach of Branke et al. [BSS01] has a more solid mathematical basis, yet also requires to tune a correlation distance.

5.6 A General Discussion of Noise Handling Techniques

In the previous sections, the working mechanisms of a number of noise handling techniques have been summarized. In this review we have made the categorization of basic noise handling techniques, adaptive averaging techniques, and metamodel assisted noise handling techniques. The basic noise handling techniques provide the basic techniques on how to reduce the undesirable effects of noise. Adaptive averaging techniques aim to automatically adapt the parameters of static noise handling techniques. Metamodeling techniques, on the other hand, attempt to build a (local) surrogate model of the original objective function, therewith aiming to generate a near noise-free surrogate objective function.

Although implicit and explicit averaging are easier to implement than all other schemes presented, different noise handling techniques seem to be preferable for three reasons:

1. Maintaining arbitrary convergence accuracy.

2. Costly evaluations require efficient usage of the evaluation budget.

3. Eliminating the need to specify sensitive noise-handling parameters.

The first reason regards the drawback of implicit and explicit averaging to target the optimum with arbitrary precision. However, many adaptive averaging techniques have serious difficulties with a seemingly exploding number of required evaluations. The races and rank-change based uncertainty handling techniques therefore use upper bounds on the evaluation intensity, which in turn bounds the convergence accuracy.

The second, aiming for saving costly function evaluations, has a more practical basis. This can be seen as motivation for the races and rank-change based uncertainty handling techniques that by using upper limits on the evaluation effort, therewith aiming to achieve a better convergence accuracy within less time than a static noise handling technique. Metamodeling techniques are constructed for this practical purpose too. The computational demands of metamodeling techniques makes them suitable only in the cases where evaluations are costly, but the alleged gain is that a reduced number of objective function evaluations are acquired.

The third regards the ideal not to have to set parameters like the sample size for explicit resampling beforehand. When looking at the adaptive averaging and metamodeling techniques that have been discussed in the previous section, none of them manages to accomplish this. Yet, the parameters that are replaced, for instance, in the rank-change based uncertainty handling technique could be less sensitive than using a fixed sample size.

Based on the former observations, we conclude that for an advanced noise handling technique to be practically viable, it should be an improvement with respect to implicit or explicit averaging in either of the following scenarios:

1. **Arbitrary convergence accuracy:** the advanced noise handling technique can be proven to maintain the global convergence criterion.

2. **Sampling efficiency:** given an arbitrary, but fixed evaluation budget, the advanced noise handling technique should ideally outperform any static averaging technique.

3. **Parameter reduction:** the advanced noise handling technique should either be completely parameter-less (*strict parameter reduction*), or given canonical settings, it should outperform any fixed static averaging scheme on a majority of problems (*weak parameter reduction*).

Of the approaches presented in literature, we can say that *arbitrary convergence accuracy* does not hold for implicit and explicit averaging nor for adaptive averaging techniques that limit the evaluation accuracy. For the adaptive averaging techniques that do not limit the evaluation accuracy and for the metamodeling approaches, *arbitrary convergence accuracy* remains to be proven. For all adaptive averaging techniques considered in this chapter, *sampling efficiency* remains to be proven. For all adaptive averaging techniques and metamodeling noise handling techniques, strict *parameter reduction* is not achieved and *weak parameter reduction* remains to be proven.

5.7 Summary and Discussion

In this chapter we have reviewed the problem of optimization of noisy objective functions using Evolution Strategies from the perspective of the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES.

There are different goals for optimization of noisy objective functions, which are modeled differently in an effective objective function formulation. Choosing an effective objective function is a design issue that depends on the problem at hand. In systems with intrinsic additive noise, optimization of the expected objective function is most customary. In systems in which the noise is due to errors in measurement, optimization of the real underlying objective function is more appropriate. When the noise is stationary, these two goals are equivalent. When the noise is non-stationary, other options are also reasonable.

Evolution Strategies are fairly robust against noise when considering the expected objective function as an optimization goal. However, noise limits the convergence accuracy of Evolution Strategies and countermeasures are needed when higher accuracy is desired.

The way in which to adapt Evolution Strategies in order to deal with noisy objective functions depends on what is aimed for. In the second part of this chapter, a number of noise handling techniques have been described, categorized as: basic noise handling, adaptive averaging, and metamodel assisted noise handling. Explicit and implicit averaging techniques can be used to improve the convergence accuracy of Evolution Strategies, but they also suffer from convergence accuracy limitations. When arbitrary convergence accuracy is aimed for, adaptive averaging techniques or metamodeling techniques should be used.

The question which of the techniques considered in this chapter is most suitable depends on the particular type of noise. Moreover, even when considering the specific case of stationary Gaussian noise, the question remains open which of the advanced noise handling schemes is the best. Suitable techniques for the $(5/2_{DI}, 35)$-σSA-ES and the CMA-ES are the partial order based adaptive averaging technique (PUH), the rank-change based adaptive averaging technique (UH), and the inversion-based adaptive averaging technique (IUH). We will consider these three adaptive averaging methods next to implicit and explicit averaging as the most appropriate candidates for noise handling. For these techniques, the questions that remain are: 1) How should the algorithmic parameters of these adaptive averaging methods be set? 2) How

do these techniques compare against each other and how do these techniques compare against their static counterparts; implicit and explicit averaging? In the next chapter we will study these issues in more detail.