Stochastic and deterministic algorithms for continuous black-box optimization

Wang, H.

CHAPTER 2

Stochastic Variation

In the continuous search space $S \subseteq \mathbb{R}^d$, the most common stochastic variation operator is the **multivariate Gaussian distribution**. It is denoted as $\boldsymbol{\mathcal{N}}(\mathbf{m}, \mathbf{C})$ where $\mathbf{m}$ is the mean vector and $\mathbf{C}$ is the covariance matrix. The definition and some properties of the Gaussian distribution can be found in Appendix A. Generating d-dimensional random vectors from a multivariate Gaussian distribution is the key source of stochastic variations in many stochastic optimization algorithms, e.g., evolution strategies (Bäck et al., 2013). The standard method to achieve this, *simple random sampling* (or random sampling for short), samples pseudo-random numbers directly from a certain distribution. However, it also results in a high *sampling error* or sampling variation, which would lead to "bad" samples (explained in the following). The sampling error occurs when we estimate the statistical properties of a distribution from its realizations. By sampling error, we mean the estimation errors of statistical properties (e.g., mean, covariance) of a distribution, which are caused by unrepresentative or biased samples.

An example of biased samples is illustrated in Fig. 2.1, in which four i.i.d. mutation vectors are sampled from a multivariate Gaussian distribution $\boldsymbol{\mathcal{N}}(\mathbf{m}, \mathbf{C})$. The black solid ellipsoid represents the expectation of the mutations and reflects the covariance matrix $\mathbf{C}$. The diversity of the four samples is not satisfactory because the minimal distance between samples is relatively small compared to the axis length of the black solid ellipsoid. A strong sampling error incurs in this case because if the mean and covariance of the distribution are estimated from these four vectors, the results would deviate largely from $\mathbf{m}$ and $\mathbf{C}$.

As a result of the biased samples, a large portion of space is not reached (at least half the space in this case). Moreover, if the objective function is twice differentiable, the contour lines should be locally convex near the optimum (the dashed ellipsoids). The

probability that a new search point represents an improvement can be very small, shown by the area with vertical lines intersecting the solid ellipsoid. Therefore, if the population size is small, an undesired sampling case can take place such that none of the mutations represents an improvement, which renders the current generation inefficient. The sampling error has an even bigger side effect in modern evolution strategies (e.g., CMA-ES (Hansen, 2006)) because those algorithms tend to exploit small populations to speed up their convergence. To overcome this problem, it is proposed here to develop special sampling approaches for the reduction of sampling error in a small population, such that the statistical properties estimated from mutation samples are more similar to their underlying true distribution.

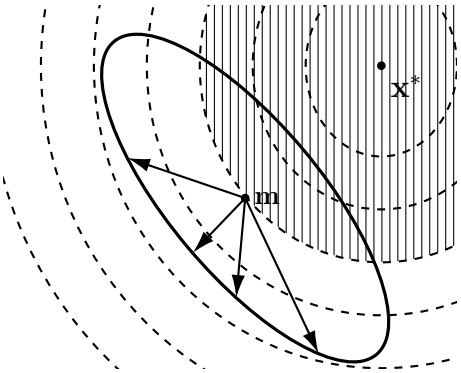


Figure 2.1: Illustration of a set of unsuccessful mutation samples. Four offspring are generated here while none of them is an improvement. This phenomenon reduces the convergence velocity of the algorithm.

The sampling method proposed in this chapter is plugged into evolution strategies (ES) for testing. To make this chapter self-contained, the algorithmic structure of $(\mu \overset{+}{,} \lambda)$-ES is given in Alg. 1. For the details on evolution strategies, please see Emmerich, Shir, and Wang (2018).

2.1 Quasi-Random Sampling

There are some techniques proposed to reduce the sampling error as much as possible and to enhance the diversity. The first method is called quasi-random sampling or low-discrepancy sequences (Dick and Pillichshammer, 2010). Low-discrepancy sequences are commonly used as a replacement of uniformly distributed

Algorithm 1 $(\mu \dot{+} \lambda)$ Evolution Strategy

procedure $(\mu \dot{+} \lambda)$-ES$(\mu, \lambda, f, \sigma_0)$
  $\mathbf{C} \leftarrow \mathbf{I}, \quad \sigma \leftarrow \sigma_0$ ▷ initialization
  while not terminated **do**
    $\mathbf{m} \leftarrow \frac{1}{\mu} \sum_{i=1}^{\mu} \mathbf{x}_i$ ▷ recombination
    for $i = 1 \rightarrow \lambda$ **do**
      $\mathbf{x}'_i \leftarrow \mathbf{m} + \sigma \mathbf{C}^{-1/2} \boldsymbol{\mathcal{N}}(\mathbf{0}, \mathbf{I})$ ▷ mutation/stochastic variation
      $f'_i \leftarrow f(\mathbf{x}'_i)$ ▷ function evaluation
    end for
    if comma selection is enabled **then**
      select the best μ solutions from $\{\mathbf{x}'_i\}_{i=1}^{\lambda}$.
    else
      select the best μ solutions from $\{\mathbf{x}'_i\}_{i=1}^{\lambda} \cup \{\mathbf{x}_i\}_{i=1}^{\lambda}$.
    end if
    Set the new population $\{\mathbf{x}_i\}_{i=1}^{\mu}$ to the selected points.
    Control step-size σ and covariance matrix $\mathbf{C}$.
  end while
  return the best solution found since the beginning.
end procedure

numbers. Intuitively, such sequences span the search space more "evenly" than the pseudo-random numbers. It is widely used in numerical problems like the quasi-Monte-Carlo method (Niederreiter, 1992) to achieve a faster rate of convergence. The discrepancy of a random sequence can be viewed as a quantitative measure for the deviation from the uniform distribution. Thus, the low-discrepancy sequence is able to solve the same problem as the one treated here, namely to create more evenly distributed samples.

Due to the advantages of quasi-random sampling, it is also applied in genetic algorithms (Kimura and Matsumura, 2005) and evolution strategies (Teytaud and Gelly, 2007). Specifically, it has already been applied to the well-known Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen and Ostermeier, 2001; Hansen et al., 2003). Teytaud and Gelly (2007) propose to replace the independent random Gaussian samples by a low-discrepancy sequence in the mutation operator. The method for generating quasi-random samples according to the Gaussian distribution is also proposed because the quasi-random samples are usually related to a uniform distribution. It is also argued that the efficiency of CMA-ES is

improved due to the bigger diversity of quasi-random samples. However, when applying the quasi-random sample and recombination operator, a systematic bias on the step-size adaptation is induced: the quasi-random samples are no longer independent from each other and thus for each highly anti-correlated samples, their recombination is much smaller on average compared to the Gaussian mutation. As the step-size adaptation mechanism typically depends on the expected size of the recombinations, quasi-random samples causes a downward bias in step-sizes.

2.2 Mirroring and Orthogonalization

The mirrored sampling technique (Brockhoff et al., 2010) is another method for obtaining "good" samples and it is successfully accelerating the convergence of ESs (Auger et al., 2010). It is a quite simple and elegant idea in which a single random mutation vector is used to create two search points. More specifically, instead of generating λ i.i.d. search points, only half of the mutation vectors are sampled during each ES generation, namely $\{\mathbf{z}_{2i-1}\}_{1\leq i\leq \lambda/2}$, $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{C})$, where σ is the current global step size and $\mathbf{C}$ is the current covariance matrix. Each mutation vector $\mathbf{z}_{2i-1}$ is used to generate two offspring, the usual one $\mathbf{x}_{2i-1} = \mathbf{m} + \mathbf{z}_{2i-1}$ and the mirrored offspring $\mathbf{x}_{2i} = \mathbf{m} - \mathbf{z}_{2i-1}$. Those two offspring are *symmetric* or *mirrored* to the parental point $\mathbf{m}$.

In order to make the argument here clearer, the mutations sampled from the distribution are denoted as *realized* mutations. The mirrored sampling method is described in Algorithm 2, acting as an alternative to the random mutation operator in evolution strategies. For odd λ, it begins by generating $\lceil \lambda/2 \rceil$ offspring in the first generation, which results in $\lceil \lambda/2 \rceil$ mirrored offspring. Then, all of the realized offspring and $\lceil \lambda/2 \rceil - 1$ mirrored ones are used immediately while the extra one mirrored mutation is kept to the next iteration (Lines $18-21$). In the next iteration, the extra mirrored offspring is used (Lines $3-9$) and only $\lfloor \lambda/2 \rfloor$ realized mutations need to be drawn. The following generations repeat this procedure. The static variable $\mathbf{z}_{\text{last}}$ in Algorithm 2 stores the extra realized mutation vector. Here, the notation proposed in Brockhoff et al. (2010) is used such that any ES algorithm with the mirrored sampling is denoted by $(1 \overset{+}{,} \lambda_{\text{m}})$-ES.

By using mirrored sampling, the mirrored mutations are entirely dependent on the realized mutation samples and explore the reverse (or mirrored) directions such

that the mirrored counterpart of an unsuccessful mutation has a certain chance to realize an improvement.

Algorithm 2 Mirrored Sampling

procedure MIRRORED($\mathbf{m}, \sigma, \mathbf{C}, \lambda$)
  $\mathbf{B}, \mathbf{D} \leftarrow$ EIGEN-DECOMPOSITION($\mathbf{C}$)
  if $\lambda \mod 2 \neq 0$ **and** $\mathbf{z}_{\text{last}}$ is set **then**
    $\mathbf{x}_\lambda \leftarrow \mathbf{m} - \sigma\mathbf{B}\mathbf{D}\mathbf{z}_{\text{last}}$ ▷ mutation $\mathbf{z}_{\text{last}}$ from the last iteration
    $\lambda' \leftarrow \lambda - 1$
    Unset the static variable $\mathbf{z}_{\text{last}}$ ▷ Unset $\mathbf{z}_{\text{last}}$
  else
    $\lambda' \leftarrow \lambda$
  end if
  for $i = 1 \rightarrow \lambda'$ **do**
    if $i \mod 2 = 0$ **then**
      $\mathbf{x}_i \leftarrow \mathbf{m} - \sigma\mathbf{B}\mathbf{D}\mathbf{z}_{i-1}$ ▷ Mirroring
    else
      $\mathbf{z}_i \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{I})$
      $\mathbf{x}_i \leftarrow \mathbf{m} + \sigma\mathbf{B}\mathbf{D}\mathbf{z}_i$
    end if
  end for
  if $\lambda' \mod 2 \neq 0$ **then** ▷ Odd number of mutations are created
    Set the static variable $\mathbf{z}_{\text{last}} \leftarrow \mathbf{z}_\lambda$ ▷ Save $\mathbf{z}_\lambda$ for the next iteration
  end if
  return $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\lambda\}$
end procedure

2.2.1 Deterministic Orthogonal Sampling

Orthogonal sampling, which denotes a the sampling approach utilizing orthogonal search directions, is another solution to enhance the mutation diversity. This sampling scheme can be found in Coordinate Descent (Schwefel, 1993), Adaptive Coordinate Descent (ACiD) (Loshchilov et al., 2011) and Rosenbrock's Local Search (Rosenbrock, 1960). Intuitively, by sampling on the mutually orthogonal directions, the samples spread quite diversely such that the search space would be

explored more evenly. This sampling method is not well suited for solving the problem discussed here, but gives a lot of inspiration for the proposed method.

Normally, in this approach, a set of orthogonal basis vectors $\Xi = \{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_n\}$ are maintained in each optimization iteration, determining the exploration directions. In each iteration, only a line search is conducted along a basis vector, which is achieved by sampling two trial points: one point is created by adding the basis to the current search point $\mathbf{m}$ while the other one is mirrored. In the next iteration, another basis vector in Ξ is picked for the exploration. The general framework of the optimization algorithm using this method is summarized below:

1. Initialize the search point $\mathbf{m}$, an orthonormal basis $\Xi = \{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_n\}$ and the step sizes $\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ for each vector in the basis.

2. If the termination condition is not satisfied, perform the following steps until (e) for each iteration. Let g be the iteration counter:

   (a) Choose vector $\boldsymbol{\xi}_i$ as the exploration direction where $i = g \mod n$ and generate one trial point: $\mathbf{x}_1 = \mathbf{m} + \sigma_i \boldsymbol{\xi}_i$.

   (b) For Rosenbrock's local search, goto (c). For the other methods, use vector $\boldsymbol{\xi}_i$ to generate the other trial point: $\mathbf{x}_2 = \mathbf{m} - \sigma_i \boldsymbol{\xi}_i$.

   (c) Evaluate the trial points $\mathbf{x}_1, \mathbf{x}_2$ (if $\mathbf{x}_2$ exists). Set the search point $\mathbf{m}$ to the one with the best fitness value.

   (d) Update the step size σ_i according to a deterministic or stochastic rule and increase the iteration counter g by one.

   (e) If $g \mod n = 0$, then update the basis Ξ according to the search points of the most recent n iterations.

When all vectors in Ξ are tried, the orthogonal basis Ξ is either unchanged or updated based on the successful vectors in the history. Note that the rules of the update may be different in every optimization algorithm. In Coordinate Descent, the basis is fixed to the standard basis in $\mathbb{R}^d$ during the process. In ACiD, the basis is updated by Adaptive Encoding (Hansen, 2008), which is the generalization of the covariance matrix update in CMA-ES. We deliberately term this sampling method as *deterministic* orthogonal sampling due to the fact that the update of the orthonormal basis is completely deterministic and it is easier to distinguish this sampling method from the *random* orthogonal sampling proposed here.

2.2.2 Mirrored Orthogonal Sampling

In this section, we propose a new sampling method based on the mirrored sampling technique. The motivation, the algorithm and the implementation are provided. This new method is motivated by the following observation: In mirrored sampling, half of the mutation vectors (the mirrored or dependent ones) completely depend on the other half (the realized or independent ones). Between these two sets of mutations, mirrored sampling is able to guarantee a significant difference between a realized mutation and its mirrored counterpart. In addition, the mirrored mutation is anti-parallel to the realized one and thus a mirrored pair would span two half-spaces almost surely, no matter how the search space is partitioned (such a pair can stay on the partition boundary with zero probability). However, within the realized half of mutations, everything is still purely random and not arranged evenly in high-dimensional space. Thus, the mirrored sampling technique still suffers from undesirable clustering of samples.

In order to improve the realized half of the mutations, we resort to the deterministic orthogonal sampling method (Section 2.2.1), where the mutations (new search points) are generated along a precomputed orthogonal basis and thus the minimal distance between samples is greatly enlarged. The disadvantage is that it only works in the single-parental evolutionary algorithm and only one of the orthogonal samples can be used in one evolution cycle, which limits its usability for the general (μ, λ)-ES. Instead of generating mutations along some orthogonal basis, it is proposed here to create half the mutations as "uniform random orthogonal vectors", in the sense that each vector is stochastic instead of the deterministic orthonormal basis and "uniform random" requires each vector to sample each direction evenly (Wang et al., 2014).

Definition 2.1. *The **uniform random orthogonal vectors** are defined as a set of random vectors $\{\boldsymbol{\mathcal{O}}_1, \boldsymbol{\mathcal{O}}_2, \ldots, \boldsymbol{\mathcal{O}}_k\} \subset \mathbb{R}^d$ $(k \leq d)$, satisfying the following three properties:*

1. *Orthogonality:* $\forall i \neq j \in \{1, 2, \ldots, k\}, \ \langle \boldsymbol{\mathcal{O}}_i, \boldsymbol{\mathcal{O}}_j \rangle = 0.$
2. $\chi(d)$*-distributed norm:* $\forall i \in \{1, 2, \ldots, k\}, \ \|\boldsymbol{\mathcal{O}}_i\| = \sqrt{\langle \boldsymbol{\mathcal{O}}_i, \boldsymbol{\mathcal{O}}_i \rangle} \sim \chi(d).$
3. *Uniformality: for each vector $\boldsymbol{\mathcal{O}}_i$, its normalization $\boldsymbol{\mathcal{O}}_i / \|\boldsymbol{\mathcal{O}}_i\|$ distributes* uniformly *on the unit sphere.*

Remark. 1) The norm of those vectors is restricted to $\chi(d)$-distribution for mimicking the behavior of the vector samples from the standard multivariate

Gaussian distribution. 2) The uniform distribution on the unit sphere is equivalent to the *rotation-invariant* property with respect to an arbitrary rotation matrix[1] $\mathbf{R} \in \mathbb{R}^{d\times d}$: the random vector $\mathbf{x}$ and the rotated one $\mathbf{x}' = \mathbf{R}\mathbf{x}$ are identically distributed. 3) Throughout this thesis, the dot product is taken for the inner product, namely $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y}$.

The new mutation method is named *random orthogonal sampling*. For clarity, the mutation operator (that takes i.i.d. normal samples) in the canonical CMA-ES is called *standard random sampling*. In addition, the random orthogonal samples are rescaled and rotated according to the covariance matrix $\mathbf{C}$ before they are added to the parental point $\mathbf{m}$, which follows the same as the procedures as for the Gaussian mutations :

$$\mathbf{x}_{2i-1} \leftarrow \mathbf{m} + \sigma \mathbf{C}^{\frac{1}{2}} \boldsymbol{\mathcal{O}}_i, \ 1 \leq i \leq \lambda/2. \tag{2.1}$$

The $\mathbf{x}$'s are the new search points and σ denotes the step size. The implementation of the random orthogonal sampling algorithm and the validity of the implementation are discussed in the following section. Consider two i.i.d. vectors $\mathbf{x}$ and $\mathbf{y}$ drawn from a standard normal distribution. The expected value of the inner product of these two vectors is given as:

$$\mathbb{E}\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} \mathbb{E} x_i y_i = 0.$$

This indicates two independent standard normal vectors are orthogonal to each other in expectation. Intuitively, by generating random orthogonal samples, the mutations are derandomized such that the variance of the angle formed by a pair of mutations vanishes. Therefore, the search directions are guaranteed to be uncorrelated so that the mutation samples are spread over the space more evenly. In the next step, we combine mirrored sampling with random orthogonal sampling such that the remaining half of the search points are created by mirroring, which reads as follows:

$$\mathbf{x}_{2i} \leftarrow \mathbf{m} - \sigma \mathbf{C}^{\frac{1}{2}} \boldsymbol{\mathcal{O}}_i, \ 1 \leq i \leq \lambda/2. \tag{2.2}$$

Note that only using random orthogonal sampling is not sufficient for exploration due to the fact that random orthogonal vectors are only capable of spanning one orthant of the space, no matter how they are realized (just consider the

[1] A d-dimensional rotation matrix $\mathbf{R}$ satisfies conditions $\mathbf{R}^{-1} = \mathbf{R}^\top$ and $\det(\mathbf{R}) = 1$. All such matrices form so-called special orthogonal group $\mathrm{SO}(d)$.

Algorithm 3 Mirrored Orthogonal Sampling

1: **procedure** MIRRORED-ORTHOGONAL($\mathbf{m}, \sigma, \mathbf{C}, \lambda$)
2: $\mathbf{B}, \mathbf{D} \leftarrow$ EIGEN-DECOMPOSITION($\mathbf{C}$)
3: **if** $\lambda \mod 2 \neq 0$ **and** $\mathbf{z}_{\text{last}}$ is not set **then**
4: $\mathbf{x}_\lambda \leftarrow \mathbf{m} - \sigma\mathbf{B}\mathbf{D}\mathbf{z}_{\text{last}}$ ▷ $\mathbf{z}_{\text{last}}$: the unused mutation from the last iteration
5: $\lambda' \leftarrow \lambda - 1$ ▷ One offspring is already created
6: Unset the static variable $\mathbf{z}_{\text{last}}$. ▷ Unset $\mathbf{z}_{\text{last}}$ once it is used
7: **else**
8: $\lambda' \leftarrow \lambda$
9: **end if**
10: $p \leftarrow \lceil \lambda'/2 \rceil$
11: $\{\mathbf{z}_i\}_{i=1}^{p} \leftarrow$ ORTHOGONAL(p) ▷ sub-procedure, see Alg. 5
12: **for** $i = 1 \rightarrow p$ **do**
13: $\mathbf{x}_{2i-1} \leftarrow \mathbf{m} + \sigma\mathbf{B}\mathbf{D}\mathbf{z}_i$
14: $\mathbf{x}_{2i} \leftarrow \mathbf{m} - \sigma\mathbf{B}\mathbf{D}\mathbf{z}_i$ ▷ Mirroring
15: **end for**
16: **if** $\lambda' \mod 2 \neq 0$ **then** ▷ Save the unused mutation to the next iteration
17: Set the static variable $\mathbf{z}_{\text{last}} \leftarrow \mathbf{z}_p$
18: **end if**
19: **return** $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\lambda\}$
20: **end procedure**

canonical basis in 3-D). Combining Eq. (2.1) and (2.2), the new sampling approach is completed and is called *mirrored orthogonal sampling*. In addition, any ES algorithm exploiting it is denoted as $(\mu \overset{+}{,} \lambda_{\text{m}}^{\text{o}})$-ES here. The detailed algorithm of the mirrored orthogonal sampling method is given as Algorithm 3. Note that an algorithm for generating random orthogonal Gaussian vectors (which is explained in the following) is invoked in line 10 and replaces the direct sampling of the Gaussian distribution. The remainder of this algorithm is basically the same as mirrored sampling (Alg. 2).

The mirrored orthogonal sampling method is a variant of mirrored sampling. In addition to mirroring, which ensures the difference within any mirrored pair, the orthogonalization method is exploited to guarantee the significant differences among realized mutations. Therefore, it is quite straightforward to compare the performance of mirrored orthogonal sampling to that of mirrored sampling and to

that of standard sampling. Such a comparison is presented in the experimental results (Section 2.4).

2.2.3 Implementation of Random Orthogonal Sampling

In order to implement random orthogonal sampling as introduced previously, the well-known Gram-Schmidt process (Björck, 1994) is exploited to generate the orthogonal samples. The Gram-Schmidt process is a method for orthogonormalizing a set of vectors in an inner product space, most commonly the Euclidean space $\mathbb{R}^d$. It takes a finite, linearly independent set $\mathcal{S} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ for $k \leq d$ and generates an orthogonal set $\mathcal{S}' = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ that spans the same k-dimensional subspace of $\mathbb{R}^n$ as $\mathcal{S}$. The Gram-Schmidt process is shown in Alg. 4.

Algorithm 4 Gram-Schmidt orthonormalization

procedure GRAM-SCHMIDT($\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$)
  for $i = 2 \rightarrow k$ **do**
    for $j = 1 \rightarrow i - 1$ **do**
      $\mathbf{v}_i \leftarrow \mathbf{v}_i - \left(\mathbf{v}_i^\top \mathbf{v}_j / ||\mathbf{v}_j||^2\right) \mathbf{v}_j$ ▷ Othogonalizing $\mathbf{v}_i$ to $\mathbf{v}_j$
    end for
  end for
  for $i = 1 \rightarrow k$ **do**
    $\mathbf{v}_i \leftarrow \mathbf{v}_i / ||\mathbf{v}_i||$ ▷ Normalization
  end for
  return $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$
end procedure

Let p equal $\lambda/2$ again. In the first step, we sample p i.i.d. vectors from the standard normal distribution and record their norms (lengths), i.e.:

$$\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_p\}, \quad \mathbf{s}_i \sim \boldsymbol{\mathcal{N}}(\mathbf{0}, \mathbf{I}), \quad L_i = \|\mathbf{s}_i\|, \quad i = 1, \ldots, p. \tag{2.3}$$

Note that the Gram-Schmidt process is an orthonormalization method, normalizing the lengths of the i.i.d. samples. Therefore, the lengths have to be manually recorded so that we can restore mutation lengths for the samples. Then, processing $\mathcal{S}$ by the Gram-Schmidt process would give us a collection $\mathcal{S}'$ of random orthonormal vectors,

$$\mathcal{S}' = \left\{\mathbf{s}'_1, \ldots, \mathbf{s}'_p\right\} = \text{GRAM-SCHMIDT}(\mathcal{S}). \tag{2.4}$$

Note that each vector of $\mathbf{s}'_1, \ldots, \mathbf{s}'_p$ is of unit length and those vectors are orthogonal to each other. It is not very hard to see from Algorithm 4 that among all the resulting vectors, the direction of $\mathbf{s}'_1$ remains unchanged and the direction of $\mathbf{s}'_i$ depends on the set $\{\mathbf{s}_k\}_{k=1}^{i-1}$. Therefore, intuitively, the output vectors of the Gram-Schmidt process, $\{\mathbf{s}'_i\}_{i=1}^{p}$ are uniformly distributed on the unit sphere because the input vectors $\{\mathbf{s}_k\}_{k=1}^{p}$ are independent and identically distributed. Finally, we rescale all the $\mathbf{s}'_i$ by their corresponding original length:

$$\mathbf{z}_i = L_i \mathbf{s}'_i, \quad i = 1, \ldots, p. \tag{2.5}$$

The resulting random vectors are orthogonal Gaussian samples, which completes this process. A special situation takes place if p is greater than the dimensionality d: it is simply not possible to generate more than d distinct orthogonal vectors in $\mathbb{R}^d$. In this case, only d mutation samples are created using Equations (2.3), (2.4) and (2.5), and the remaining $p - d$ samples are created using the standard random sampling. The detailed procedure of orthogonal sampling is described in Algorithm 5. Lines $3 - 6$ correspond to Eq. (2.3). Through lines $7 - 17$, the Gram-Schmidt process is invoked and the number of samples p is handled properly. The advantage of this implementation is that there is no additional parameter to be considered. As for the time complexity, extra costs are spent in calling the Gram-Schmidt process, which is $O(k^2 d), k = \min\{p, d\}$.

To justify this implementation, it is possible to check the generated samples according to Definition 2.1: the orthogonality and restriction on the vectors length are immediately satisfied. The rotation-invariance of the vectors can be shown as follows. Firstly, the standard normal vectors are rotation-invariant, meaning that for every $\mathbf{s}_i \sim \boldsymbol{\mathcal{N}}(\mathbf{0}, \mathbf{I})$, it has the same distribution as $\mathbf{R}\mathbf{s}_i$, where $\mathbf{R}$ is the rotation matrix taken from $\mathrm{SO}(d)$. Second, the orthogonalization formula of the Gram-Schmidt process, which is encoded in Algorithm 4, reads as follows:

$$\mathbf{s}'_i = \mathbf{s}_i - \sum_{j=1}^{i-1} \frac{\langle \mathbf{s}_i, \mathbf{s}_j \rangle}{||\mathbf{s}_j||^2} \mathbf{s}_j, \quad i = 1, \ldots, p,$$

Now if an arbitrary rotation operator $\mathbf{R} \in \mathrm{SO}(d)$ is applied on $\mathbf{s}'_i$, the resulting vector is,

$$\mathbf{s}''_i = \mathbf{R}\mathbf{s}'_i = \mathbf{R}\mathbf{s}_i - \sum_{j=1}^{i-1} \frac{\langle \mathbf{R}\mathbf{s}_i, \mathbf{R}\mathbf{s}_j \rangle}{||\mathbf{R}\mathbf{s}_j||^2} \mathbf{R}\mathbf{s}_j, \quad i = 1, \ldots, p, \tag{2.6}$$

Note that it is valid to put $\mathbf{R}$ in the norm and the inner product (e.g., $||\mathbf{R}\mathbf{s}_j||$) because such matrices preserve the inner product. Finally, $\mathbf{R}\mathbf{s}_i$ is identically

Algorithm 5 Orthogonal sampling

procedure ORTHOGONAL(p)
  for $i = 1 \to p$ **do**
    $\mathbf{s}_i \leftarrow \boldsymbol{\mathcal{N}}(\mathbf{0}, \mathbf{I})$ ▷ generate standard normal vectors
    $L_i \leftarrow \|\mathbf{s}_i\|$ ▷ store the length
  end for
  $k \leftarrow \min\{p, n\}$ ▷ number of inputs for Gram-Schmidt
  $\{\mathbf{s}'_1, \ldots, \mathbf{s}'_k\} \leftarrow$ GRAM-SCHMIDT$(\{\mathbf{s}_1, \ldots, \mathbf{s}_k\})$ ▷ sub-procedure, see Alg. 4
  for $i = 1 \to k$ **do**
    $\mathbf{z}_i \leftarrow L_i \mathbf{s}'_i$ ▷ rescale the length
  end for
  if $k < p$ **then** ▷ more than n samples are needed
    for $i = 1 \to p - k$ **do**
      $\mathbf{z}_{k+i} \leftarrow \mathbf{s}_{k+i}$ ▷ copy the standard normal vectors
    end for
  end if
  return $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_p\}$
end procedure

distributed as $\mathbf{s}_i$ and it also holds for the remaining terms in the right-hand-side of Eq. (2.6). Therefore, $\mathbf{s}''_i$ is identically distributed as $\mathbf{s}'_i$ and therefore it is rotation-invariant. A more rigorous proof can be found in Eaton (1983).

2.3 Convergence Analysis of Mirroring and Orthogonalization

The theoretical analysis is twofold. First, the progress rate analysis for $(1, \lambda)$-ES, introduced in Beyer (1993), is applied to analyze mirrored sampling. In addition, such analysis gives a straightforward explanation why mirrored orthogonal sampling improves performance. There are no analytical results for mirrored orthogonal sampling yet while its empirical results are compared to random and mirrored sampling. Second, the progress rate analysis is applied again to provide an analytical result about the worst case performance of mirrored orthogonal sampling. This will (partially) explain the advantages of the new sampling method. For the analysis in the following, we will only consider the $(1, \lambda)$-ES with isotropic mutations on

the so-called sphere function[1], which is defined as:

$$f(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^*)^\top (\mathbf{x} - \mathbf{x}^*),\ \mathbf{x} \in \mathbb{R}^d,$$

which has the global minimum $\mathbf{x}^*$. In addition, for the simplicity of our deviation, it is also assumed that the population size λ is **even** in the following analysis. In practice, when λ is odd, the corresponding progress rate can be bounded from below by using $\lambda - 1$ in the analysis and also be bounded from above by using $\lambda + 1$. Note that although some results (e.g., Fig. 2.3b) can be equivalently obtained, using the theoretical framework of *convergence rate analysis* (Brockhoff et al., 2010), we did not adopt such an analysis approach because the progress rate analysis gives more insight into why the proposed sampling method outperforms its counterparts. The link between progress rate and convergence rate is elaborated in Auger and Hansen (2011). For the convergence rate analysis on the mirrored sampling method, please see Auger et al. (2011a,b).

2.3.1 Mirrored Sampling

We will begin with the analysis of the $(1, \lambda_\mathrm{m})$-ES in order to show the reason why it outperforms random sampling and this analysis serves as a baseline for the comparison to mirrored orthogonal sampling, which is investigated here by the Monte Carlo simulation. The basics of the analysis are shown in Fig. 2.2a, following the same treatment as in Bäck (1995). Let $\mathbf{P}$ be the current parent which is at a distance R from the optimum $\mathbf{O}$. Due to the spherical symmetry, only the distance R is crucial, not the actual position of $\mathbf{P}$. The hypersphere centered at $\mathbf{P}$ has a radius of $\sigma\sqrt{d}$ and represents the mean length of isotropic Gaussian mutations: $\mathbf{z} = \boldsymbol{\mathcal{N}}(\mathbf{0}, \sigma^2\mathbf{I})$. The mirrored mutation is indicated as $-\mathbf{z}$. The progress of each mutation can be measured by the projection of $\mathbf{z}$ onto line $\mathbf{PO}$, which is the random variable z. Due to the invariance properties of isotropic Gaussian vectors, z is found to be normally distributed as $\mathcal{N}(0, \sigma^2)$, regardless of the actual direction of $\mathbf{PO}$. The progress made by mutation $\mathbf{z}$ is $R - r$. Furthermore, for a set of mutations $\{\mathbf{z}_i\}_{1 \le i \le \lambda}$, the actual progress made by all the mutations is $R - r_{1:\lambda}$, where $r_{1:\lambda}$ is the smallest order statistic among $\{r_i\}_{1 \le i \le \lambda}$. The progress rate is

[1] The sphere function is a standard function for theoretical analysis, reflecting local convergence properties of ESs.

defined in Beyer (2013):

$$\varphi_{1,\lambda} = \mathbb{E}\left\{R - r_{1:\lambda}\right\} \simeq \mathbb{E}\left\{R - \sqrt{(R - z_{\lambda:\lambda})^2 + \sigma^2 d}\right\}. \tag{2.7}$$

The approximation in Eq. (2.7) takes place when we replace $||\mathbf{z}||$ by $\sigma\sqrt{d}$. Note that $z_{\lambda:\lambda}$, the largest order statistic among the projections of all the mutations onto **PO**, determines the expectation above.

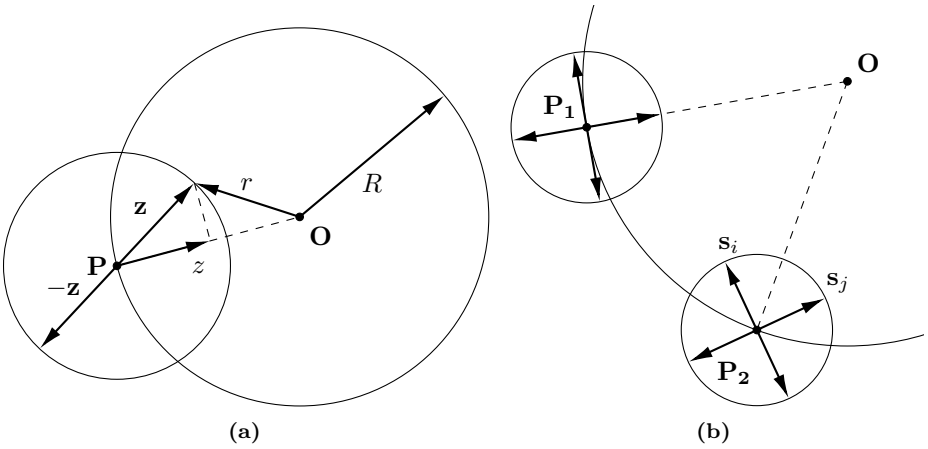


Figure 2.2: (a). Schematic diagram for the progress rate analysis on the sphere function. The mutations are centered at **P**, which is at distance R from the optimum **O**. (b) In 2-D, the diagram shows the best case ($\mathbf{P}_1$) of progress and the worst case ($\mathbf{P_2}$) for mirrored orthogonal sampling on the sphere function.

For the mirrored sampling, if z_i is the projection of mutation $\mathbf{z}_i$ onto **PO**, then the projection of its mirrored mutation $-\mathbf{z}_i$ is $-z_i$ by symmetry. Thus, the set of the projections of all the mutations of mirrored sampling can be written as $\{z_i, -z_i\}_{1\leq i\leq \lambda/2}$. Let $P^{\mathrm{m}}_{\lambda:\lambda}(Z \leq z)$ denote the cumulative probability distribution (c.d.f.) of the largest order statistic among $\{z_i, -z_i\}_{1\leq i\leq \lambda/2}$. Suppose for every $z \geq 0$, in order to facilitate the condition in $P^{\mathrm{m}}_{\lambda:\lambda}(Z \leq z)$, namely the largest order statistic is less than or equal to z, we must have $z_i \leq z, -z_i \leq z$ for all the z_i, which implies $-z \leq z_i \leq z$ for all the z_i. The intuition is that all random mutation points are required to be sampled less than or equal to z. In addition, because mirrored mutations are generated by reversing the signs of random mutations, every random mutation also needs to be bigger than $-z$, otherwise the mirrored

counterpart of an outlier would be larger than z and fails the condition. The argument reads as follows:

$$\begin{aligned} P^{\mathrm{m}}_{\lambda:\lambda}(Z \leq z) &= \left[\Pr\left(-z < Z \leq z\right)\right]^{\lambda/2} \\ &= \left[\Phi\left(\frac{z}{\sigma}\right) - \Phi\left(-\frac{z}{\sigma}\right)\right]^{\lambda/2} \\ &= \left[2\Phi\left(\frac{z}{\sigma}\right) - 1\right]^{\lambda/2}, \quad \forall z \geq 0. \end{aligned}$$

Note that $\Phi(\cdot)$ stands for the c.d.f. of a standard normal random variable. Then, in case of $z < 0$, the cumulative probability should be always 0. The reason is that if a realized mutation is sampled negative, then its mirrored counterpart would be positive. Therefore the largest order statistics could not be negative ever. In total, the c.d.f. of the largest order statistic is summarized as:

$$P^{\mathrm{m}}_{\lambda:\lambda}(Z \leq z) = \begin{cases} \left[2\Phi\left(\frac{z}{\sigma}\right) - 1\right]^{\lambda/2} & \forall z \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

And its probability density function is:

$$p^{\mathrm{m}}_{\lambda:\lambda}(z) = \begin{cases} \lambda p(\frac{z}{\sigma}) \left[2\Phi\left(\frac{z}{\sigma}\right) - 1\right]^{\lambda/2-1} & \forall z \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.8}$$

where $p(\cdot)$ denotes the probability density function (p.d.f.) of a standard normal distribution. This density can be compared to the largest order statistic among the same projections of random samples (Beyer, 1993):

$$p_{\lambda:\lambda}(z) = \lambda p\left(\frac{z}{\sigma}\right) \Phi\left(\frac{z}{\sigma}\right)^{\lambda-1}.$$

In 5-D with $\lambda = 10$, we plot the c.d.f. and p.d.f. of mirrored sampling and random sampling in Fig. 2.3a. It is clear from the figure that the distribution of the largest projection for mirrored sampling is shifted to the right, compared to that for the Gaussian sampling and therefore the corresponding distribution of projections is shifted towards larger values. This advantage would affect the progress rate (as shown in the following) and is the main reason why mirrored sampling has a better performance than random sampling. By using the normalized quantities,

$$\varphi^* = \varphi\frac{d}{R}, \quad \sigma^* = \sigma\frac{d}{R},$$

and applying the same derivation as in Beyer (1993), the progress rate of $(1, \lambda_{\mathrm{m}})$-ES can be obtained by expanding the expectation in Eq. (2.7) (the details of the

simplification are not shown here):

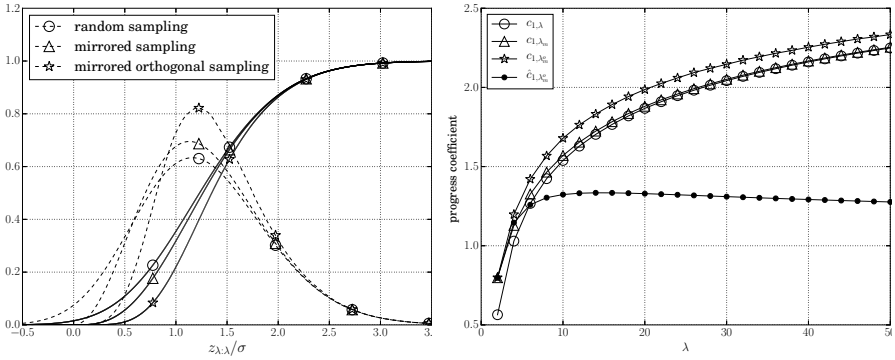
$$\begin{aligned}
\varphi^*_{1,\lambda_\mathrm{m}} &= \int_0^\infty z p^{\mathrm{m}}_{\lambda:\lambda}(z)\,\mathrm{d}z - \frac{(\sigma^*)^2}{2} \\
&= \lambda \int_0^\infty z p\left(\frac{z}{\sigma}\right)\left[2\Phi\left(\frac{z}{\sigma}\right)-1\right]^{\lambda/2-1}\,\mathrm{d}z - \frac{(\sigma^*)^2}{2} \\
&= \sigma^*\left(\lambda \int_0^\infty z' p(z')\left[2\Phi\left(z'\right)-1\right]^{\lambda/2-1}\,\mathrm{d}z'\right) - \frac{(\sigma^*)^2}{2} \\
&= c_{1,\lambda_\mathrm{m}}\sigma^* - \frac{(\sigma^*)^2}{2}. \qquad (2.9)
\end{aligned}$$

In the equation above, the integral about the normalized largest projection $z' = z/\sigma$ computes its expectation and it is known as the *progress coefficient* from (Beyer, 1993). We denote it by c_{1,λ_m} here. It can be compared to the progress coefficient of random sampling, which reads:

$$c_{1,\lambda} = \lambda \int_{-\infty}^{\infty} z p(z) \Phi(z)^{\lambda-1}\,\mathrm{d}z.$$

Note that the progress rate of random sampling can be easily obtained by replacing c_{1,λ_m} in Eq. (2.9) with $c_{1,\lambda}$. Numerically, we plot the progress coefficients of random sampling and mirrored sampling against population size in Fig. 2.3b. The mirrored sampling (the curve marked by triangles) shows a small yet obvious advantage compared to the random sampling for small population sizes. In larger populations, these two converging curves imply that mirrored sampling provides no speed-up compared to the standard ES algorithm. Thus, the application of mirrored sampling should be limited to the small population setting.

For mirrored orthogonal sampling, we would like to use the same approach as for the mirrored sampling analysis above. However, it is hard to analytically obtain the c.d.f. and the density function of the largest projection onto **PO** of the mirrored orthogonal sampling. Therefore, we compute its c.d.f. and density function empirically by Monte-Carlo simulation. For the simulation, the population size λ is set to $2d$. The mirrored orthogonal samples are projected onto **PO** and the largest projections are stored, from which the c.d.f. is estimated. The results are also summarized in Fig 2.3. In Fig. 2.3a, the c.d.f. of mirrored orthogonal sampling (the solid curve marked by stars) is more likely to distribute samples towards bigger values compared to the c.d.f. of mirrored sampling. As a consequence, in Fig. 2.3b, the progress coefficients of mirrored orthogonal sampling are significantly bigger than those of mirrored sampling, even in a large population.



(a) **(b)**

Figure 2.3: (a): The c.d.f. (solid) and p.d.f. (dashed) of the largest projection (normalized) onto **PO** for random, mirrored and mirrored orthogonal sampling. The dimension d is set to 5 and $\lambda = 10$ for all curves. There are 10^6 trials used in the estimation for mirrored orthogonal sampling. For the other sampling methods, the curves show the corresponding analytical results. (b): Progress coefficients against population size λ for random sampling, mirrored sampling and mirrored orthogonal sampling. The dimensionality d is set to $\lambda/2$ for all curves. The black dotted curve is the lower bound on the progress coefficient of mirrored orthogonal sampling.

2.3.2 Mirrored Orthogonal Sampling

The worst case analysis of mirrored orthogonal sampling is conducted when the population size is set to $2d$. We will call such population setting as "full mutations". Under this condition, the progress rate is maximized (as will be explained later) and it is possible to provide analytical results. The progress under the condition $\lambda < 2d$ will be also discussed later. In 2-D with $\lambda = 4$, the worst case (together with best case) of progress for $(1, \lambda^{\circ}_{\mathrm{m}})$ is shown in Fig. 2.2b. Suppose the step size $\sigma = 1$ here for simplification. In the mutations centered at $\mathbf{P_1}$, there is one mutation pointing to the optimum $\mathbf{O}$ and therefore this mutation performs optimally. We call this mutation scenario the best case of progresses. The progress coefficient in this case is the expectation of the standard norm mutation length. It serves as the upper bound of the progress coefficient and is the same for random, mirrored and mirrored orthogonal sampling.

The worst case of progress is indicated by the mutations centered at $\mathbf{P_2}$ in which

the angle formed by the line segment $\mathbf{P_2O}$ and mutation $\mathbf{s}_i$ is the same as the one ($\pi/4$ as shown in the figure) formed by $\mathbf{P_2O}$ and $\mathbf{s}_j$. In this scenario, the expected projections of $\mathbf{s}_i$ and $\mathbf{s}_j$ are the same. It is not possible to make the expected projection of one mutation smaller without rendering the expected projection of the other one larger. For example, if we rotate $\mathbf{s}_j$ a little bit clockwise, then its projection becomes smaller. However, in the meanwhile $\mathbf{s}_i$ is also rotated and its projection gets larger. Consequently, the largest projection of all the mutations becomes larger. Therefore, among all the possible mutation scenarios, $\mathbf{P}_2$ gives the lower bound of the largest projection of mutations onto $\mathbf{P_2O}$. Recall from Eq. (2.7) that the progress made by $(1, \lambda)$-ES is determined by the largest projection. Thus, the scenario $\mathbf{P}_2$ is the worst case of progress. Under the "full" mutation condition, we generalize the worst case for arbitrary dimensions. Let the mirrored orthogonal samples be denoted as $\{\boldsymbol{\mathcal{O}}_i, -\boldsymbol{\mathcal{O}}_i\}_{1 \le i \le \lambda/2}$. The unit vectors along the orthogonal mutations are defined as:

$$\mathbf{u}_i = \frac{\boldsymbol{\mathcal{O}}_\mathbf{i}}{\|\boldsymbol{\mathcal{O}}_i\|}. \tag{2.10}$$

Combining the unit vectors for mirrored mutations, all the unit vectors are $\{\mathbf{u}_i, -\mathbf{u}_i\}_{1 \le i \le \lambda/2}$. The worst case of progress is defined by the following conditions: for all the unit vectors, the linear combination with equal weights (denoted as $\mathbf{d}$ in the following) of $\lambda/2 = n$ unit vectors points to the optimum $\mathbf{O}$ and also to the reverse direction of the gradient of the sphere function, which reads:

$$\mathbf{d} = \sum_{k=1}^{\lambda/2} a_k \mathbf{u}_k = -\alpha \nabla f(\mathbf{x}), \quad \alpha > 0, a_k = \pm 1,$$

where a_k is a sign operator to select among $\mathbf{u}_k, -\mathbf{u}_k$. Then the scalar projection of mutation $\boldsymbol{\mathcal{O}}_i$ onto $\mathbf{d}$ is expressed as:

$$\mathrm{proj}_\mathbf{d}(\boldsymbol{\mathcal{O}}_i) = \frac{\langle \boldsymbol{\mathcal{O}}_i, \mathbf{d} \rangle}{\|\mathbf{d}\|} = \frac{\sum_{k=1}^{\lambda/2} a_k \langle \boldsymbol{\mathcal{O}}_i, \mathbf{u}_k \rangle}{\left\| \sum_{k=1}^{\lambda/2} a_k \mathbf{u}_k \right\|} = \frac{\sum_{k=1}^{\lambda/2} a_k \langle \boldsymbol{\mathcal{O}}_i, \boldsymbol{\mathcal{O}}_k \rangle / \|\boldsymbol{\mathcal{O}}_k\|}{\left\| \sum_{k=1}^{\lambda/2} a_k \mathbf{u}_k \right\|} = \frac{a_k \|\boldsymbol{\mathcal{O}}_i\|}{\sqrt{d}}.$$

Note that we substitute the expression of $\mathbf{u}_i$ (Eq. (2.10)) in the derivation above. The projections of all the mutations onto $\mathbf{d}$ can be written:

$$\mathrm{proj}_\mathbf{d} = \left\{ \frac{\|\boldsymbol{\mathcal{O}}_i\|}{\sqrt{d}}, -\frac{\|\boldsymbol{\mathcal{O}}_i\|}{\sqrt{d}} \right\}_{i=1}^{\lambda/2}.$$

The largest order statistic of all the projections is the maximum of $\mathrm{proj}_\mathbf{d}$:

$$\max\{\mathrm{proj}_\mathbf{d}\} = \max_{1 \le i \le \lambda/2} \left\{ \frac{\|\boldsymbol{\mathcal{O}}_i\|}{\sqrt{d}}, -\frac{\|\boldsymbol{\mathcal{O}}_i\|}{\sqrt{d}} \right\} = \frac{1}{\sqrt{d}} \max_{1 \le i \le \lambda/2} \{\|\boldsymbol{\mathcal{O}}_i\|\} = \frac{z}{\sqrt{d}}.$$

Here we denote the maximal mutation length by z. Note that the $\|\boldsymbol{\mathcal{O}}_i\|$ are independently distributed according to $\chi(n)$ (see Algorithm 5). Therefore, the density function of the maximal mutation length among $\lambda/2$ mutations reads:

$$p_{\frac{\lambda}{2}:\frac{\lambda}{2}}(z) = \frac{\lambda}{2} p_\chi(z) \left(F_\chi(z)\right)^{\frac{\lambda}{2}-1},$$

where $p_\chi(\cdot), F_\chi(\cdot)$ denote the density and c.d.f. of the $\chi(n)$ distribution, respectively. The worst case progress coefficient of mirrored orthogonal sampling, which is the expectation of $z/\sqrt{n}$, is denoted as $\hat{c}_{1,\lambda^{\text{o}}_{\text{m}}}$ and derived as follows:

$$\begin{aligned}
\hat{c}_{1,\lambda^{\text{o}}_{\text{m}}} &= \int_0^\infty \frac{z}{\sqrt{d}} p_{\frac{\lambda}{2}:\frac{\lambda}{2}}(z)\,\mathrm{d}z \\
&= \frac{\lambda}{2\sqrt{d}} \int_0^\infty z p_\chi(z) \left(F_\chi(z)\right)^{\frac{\lambda}{2}-1}\,\mathrm{d}z \\
&= \sqrt{d} \int_0^\infty z p_\chi(z) \left(F_\chi(z)\right)^{n-1}\,\mathrm{d}z. \qquad (2.11)
\end{aligned}$$

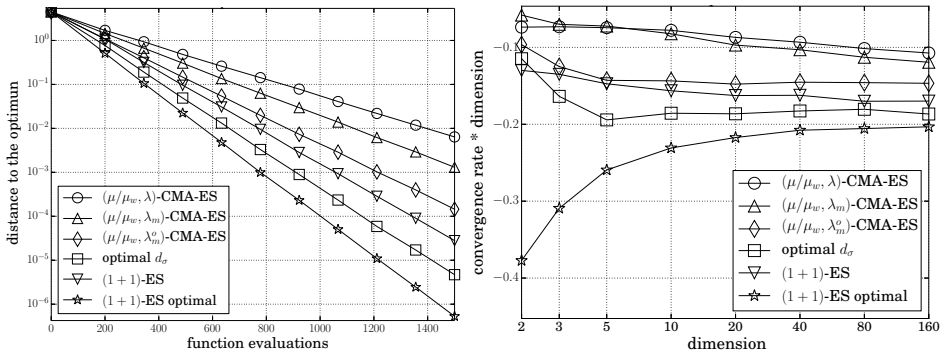
The last equation results from the fact that we picked the special population size $\lambda = 2d$ from the previous analysis setting. Eq. (2.11) is numerically evaluated and plotted in Fig. 2.3b. The curve for the worst case is above 1 and roughly stays constant when λ increases. It provides a non-zero lower bound of the progress coefficient of mirrored orthogonal sampling with "full mutations", which indicates no matter in what scenario, the mirrored orthogonal sampling with "full mutations" is going to guarantee positive progress on the sphere function. To compare, for random sampling, the lower bound of the progress coefficient is zero because it is possible to have all the mutations generated as in Fig. 2.1, where no mutation makes progress. For mirrored sampling, the lower bound of the progress coefficient is also zero because it is possible that all the mutations are generated in a tangent space of the local gradient, in which all the vectors are orthogonal to the gradient. Thus, the non-zero lower bound of mirrored orthogonal sampling with "full mutations" is its main advantage over the random and mirrored sampling.

In the case that mirrored orthogonal sampling does not use "full mutations", namely $\lambda < 2d$, the progress rate would be reduced in contrast to the "full mutations" case. This is because it can now happen that some subspace could not be covered when $\lambda < 2d$. Therefore, it is possible that the subspace in which the progress can be made is simply unexplored.

2.4 Empirical Results on Mirroring and Orthogonalization

For the multi-parental variants of ES, we only consider their empirical convergence rates here. Similar to the convergence rate estimation in Loshchilov et al. (2011), the effect of the mirrored orthogonal sampling technique on the sphere function is investigated empirically by incorporating it into the well-known CMA-ES algorithm.

On the 20-D sphere function, the convergence rates of the $(\mu, \lambda_{\mathrm{m}}^{\circ})$-CMA-ES and other comparable ES variants are illustrated in Fig. 2.4a. The empirical convergence rate is estimated as the average slope of the convergence curve over 200 runs. For



(a) Sphere 20-D **(b)** Convergence rate

Figure 2.4: The comparison of empirical convergence rates on the sphere function. All the results are estimated over 200 runs. The suggested λ setting $4 + \lfloor 3 \ln d \rfloor$ (Hansen, 2006) is used for all the CMA-ES variants (a): Plot of the history of distance to the global optimum against the number of function evaluations for four ES algorithms: $(\mu, \lambda_{\mathrm{m}}^{\circ})$-CMA-ES with standard d_σ and optimal d_σ, $(\mu, \lambda_{\mathrm{m}})$-CMA-ES, standard (μ, λ)-CMA-ES and $(1 + 1)$-ES in dimension 20. (b): Plot of convergence rate $\times$ dimensionality against the dimensionality for different algorithms on the sphere function, using 1500 function evaluations.

all the CMA-ES variants tested here, the default settings of population size are applied (Hansen, 2006): $\lambda = 4 + \lfloor 3 \ln d \rfloor, \mu = \lfloor \lambda/2 \rfloor$. The legend "$(1 + 1)$-ES" represents the $(1+1)$-ES with $1/5$ success rule step size control while the "$(1+1)$-ES optimal" is for the $(1 + 1)$-ES with scale-invariant step size setting $\sigma = \frac{1.2}{d} \|\mathbf{x}^{(k)}\|$,

which proves to be the optimal step size setting on the sphere function (Loshchilov et al., 2011).

The pairwise selection is always used if the mirroring operation is present in the sampling procedure. The mirrored sampling CMA-ES is denoted as "$(\mu, \lambda_{\mathrm{m}})$-CMA-ES ". The curve labeled by "$(\mu, \lambda_{\mathrm{m}}^{\mathrm{o}})$-CMA-ES " stands for the mirrored orthogonal CMA-ES. In addition, "optimal d_σ" represents the mirrored orthogonal CMA-ES using the optimal d_σ[1] tuning on the sphere function. Due to the empirical results, the convergence of $(\mu, \lambda_{\mathrm{m}}^{\mathrm{o}})$-CMA-ES (marked by diamonds) is slower but close to that of the $(1+1)$-ES (marked by upside-down triangles) while the $(\mu, \lambda_{\mathrm{m}}^{\mathrm{o}})$-CMA-ES using the optimal parameter settings gradually catches the convergence rates of the optimal $(1+1)$-ES in high dimensions.

The relation between the empirical convergence rate and the dimensionality is shown in Fig. 2.4b. The algorithms tested here are the same as Fig. 2.4a. It is obvious that there is a leap of convergence rates between the CMA-ES and its mirrored orthogonal competitor. The advantages of the mirrored orthogonal CMA-ES over the mirrored CMA-ES are significant and preserved even for large dimensions. The upper limit of the $(\mu, \lambda_{\mathrm{m}}^{\mathrm{o}})$-CMA-ES on the sphere function is shown by the convergence rates achieved under the optimal d_σ tuning, which is even better than $(1+1)$-ES for almost all the dimensions. However, the optimal d_σ setting on the sphere function turned out to be not robust when considering other fitness functions and therefore is not used.

2.4.1 Experiments on BBOB

The mirrored orthogonal version of CMA-ES with pairwise selection has been tested on the noiseless Black-Box Optimization Benchmark (BBOB) (Hansen et al., 2010). By using the automatic comparison procedures provided in this benchmark, the BBOB results of $(\mu, \lambda_{\mathrm{m}}^{\mathrm{o}})$-CMA-ES are compared to those of $(\mu, \lambda_{\mathrm{m}})$-CMA-ES and (μ, λ)-CMA-ES.

Experimental Settings The three algorithms, $(\mu, \lambda_{\mathrm{m}}^{\mathrm{o}})$-CMA-ES, $(\mu, \lambda_{\mathrm{m}})$-CMA-ES and (μ, λ)-CMA-ES are benchmarked on BBOB-2012 and their results are compared and processed by the post-processing procedure of BBOB. The BBOB

[1]For the definition of the parameter d_σ, please see Hansen et al. (2003).

parameter settings of the experiment are the same for all the tested ES variants. The initial global step size σ is set to 1. The maximum number of function evaluations is set to $10^4 \times d$. The initial solution (initial parent) is uniformly sampled in the hyper-box $[-4, 4]^n$. The dimensions tested in the experiment are $d \in \{2, 3, 5, 10, 20, 40\}$. The experiment employs a relatively large population size, namely $2d$, the result of which is denoted as **large population**. In this experiment, the strategy parameters used are exactly the same for the three ES variants. The modified d_σ is not used because it is tuned under the default population setting instead of the large population setting.

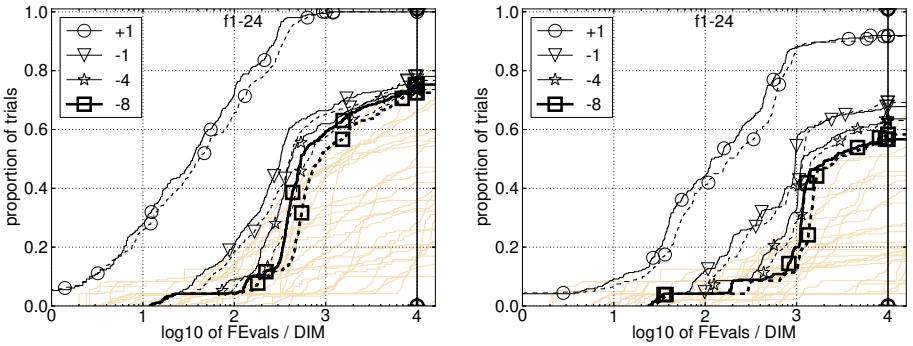


Figure 2.5: Left: $d = 5$. Right: $d = 20$. For the large population, the empirical cumulative distributions (ECDF) of run lengths (the number of function evaluations divided by dimension) for $(\mu, \lambda_{\mathrm{m}}^{\mathrm{o}})$-CMA-ES (solid lines) and $(\mu, \lambda_{\mathrm{m}})$-CMA-ES (dashed lines) needed to reach a target value.

Results and Discussion The BBOB noiseless testbed (Hansen et al., 2009) contains 24 test functions which are classified into several groups as separable, ill-conditioned or multi-modal functions. The performance of tested algorithms are compared using the aggregated empirical cumulative distribution functions (ECDFs) of run length over all the test functions are presented here. The ECDF of run length estimates the cumulative distribution of the function evaluations consumed in ESs, with respect to a given precision target. The comparisons between the mirrored orthogonal sampling and its mirrored sampling competitor are illustrated in Fig. 2.5. From the comparisons between the ECDFs of 5-D (left half) to that of 20-D (right half), it is obvious that the amount of the improvement is still significant when the dimensionality increases. The experimental results

for the large population suggest that the newly proposed mirrored orthogonal sampling technique would be most suitable in the case where the population size is about two times the dimensionality.

2.5 Efficient Global Optimization

Apart from the aforementioned mutation methods, that are directly defined by the realization of some probability distributions, in this section we shall extract and discuss the special method of creating new solutions from the *Efficient Global Optimization* (EGO) (Jones et al., 1998; Močkus, 1975, 2012) algorithm. Briefly, in this mutation method, the new candidate solution is obtained via the optimization on a well-specified utility function, which quantifies the potential "gain" in fitness value by evaluating this new solution. Therefore, we shall call this method **Mutation by Optimization**. In general, the utility function depends on a stochastic model (or statistical estimator) $\hat{f}$ of the fitness function f and statistical properties of this estimator $\hat{f}$, e.g., the *mean squared error* of the estimate: $s^2(\mathbf{x}) = \mathbb{E}\{\hat{f}(\mathbf{x}) - f(\mathbf{x})\}^2, \forall \mathbf{x} \in \mathrm{S}$. Note that $\hat{f}$ is usually called predictor in machine learning and we shall use these terminologies interchangeably here. Typically, the estimator $\hat{f}$ is a function of (random) sample $\mathrm{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathrm{S}$ and its corresponding fitness values $\mathbf{y} = (f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n))^\top$. The utility function is then denoted as $\mathscr{A}\colon \mathrm{S} \to \mathbb{R}$. The new candidate solution $\mathbf{x}'$ is proposed by solving the following problem[1]:

$$\mathbf{x}' = \arg\max_{\mathbf{x} \in \mathrm{S}} \mathscr{A}(\mathbf{x}; \Theta), \tag{2.12}$$

where Θ is a set of parameters that $\mathscr{A}$ might rely on (see Eq. (4.2) for example). In the literature, $\mathscr{A}$ is termed as *infill criterion* (Jones, 2001) or *acquisition function* (Martinez-Cantin, 2014) and we shall adopt the former throughout this thesis. Some commonly used infill criteria include: Expected Improvement, Probability of Improvement and Lower Confidence Bound. Typically, infill criteria are designed to make a balance between the model prediction $\hat{f}$ and the MSE of prediction (uncertainty) s^2. The detailed discussion on infill criteria can be found in Chapter 4. Built on the mutation by optimization mechanism, Efficient Global Optimization (also referred as Bayesian optimization (Močkus, 2012)) is able to perform a direct

[1] Some of the infill criteria are subject to minimization by the original definition. However, it can be equivalently transformed into the maximization task.

optimization efficiently on expensive objective functions. EGO is a *sequential* design strategy and it is presented in Alg. 6.

Algorithm 6 Efficient Global Optimization

procedure EGO($f, \mathscr{A}, \mathrm{S}$) ▷ f: objective function, $\mathscr{A}$: infill criterion, S: search space
  Sample the initial design $\mathrm{X} \subset \mathrm{S}$
  Evaluate $\mathbf{y} \leftarrow (f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n))^\top$
  Construct the fitness estimator $\hat{f}$ on $\mathrm{X}, \mathbf{y}$.
  while the stop criteria are not fulfilled **do**
    $\mathbf{x}' \leftarrow \arg\max_{\mathbf{x} \in \mathrm{S}} \mathscr{A}(\mathbf{x}; \Theta)$
    Evaluate $y' \leftarrow f(\mathbf{x}')$
    $\mathrm{X} \leftarrow \mathrm{X} \cup \{\mathbf{x}'\}, \mathbf{y} \leftarrow (\mathbf{y}^\top, y')^\top$
    Re-construct the estimator $\hat{f}$ on $\mathrm{X}, \mathbf{y}$
  end while
end procedure

Randomness It seems that the new location provided by solving Eq. (2.12) is *deterministic* and thus mutation by optimization is, by definition, not a stochastic variation method. However, although most of the infill criteria exhibit smooth landscapes, they are also highly *multi-modal* (see Section 4.2). This causes a practical difficulty in solving Eq. (2.12) globally. Although the exact solver, e.g., branch-and-bound (Jones et al., 1998) has been adapted for this task, such a solver only works a few types of infill criteria/covariance functions (see the discussion on the stochastic model below) and thus becomes inapplicable in practice. Instead, a stochastic optimization algorithm, e.g., evolutionary algorithms, is frequently applied for the infill criteria maximization, yielding random solutions. Thus, mutation by optimization is indeed **practically stochastic** and its randomness is determined by the underlying stochastic optimizer of the infill criterion $\mathscr{A}$.

Stochastic model To approximate the unknown objective function, the **Gaussian process regression** (**GPR**)/**Kriging** (Rasmussen and Williams, 2006; Krige, 1951) is used in EGO. It is a stochastic interpolation approach, which stems from earth science (Krige, 1951) and originally targets mining problems. It has been widely used as a surrogate model in the design and analysis of computer

experiments (Sacks et al., 1989; Santner et al., 2003), where the time-consuming simulations (computer models) are replaced by predictions from a Kriging model. In this technique, the objective function f is modeled as a realization of a Gaussian process Y. Conditioning on the data set $(\mathrm{X}, \mathbf{y})$, the so-called posterior process is obtained via Bayesian inference. The Gaussian Process Y is completely defined by a prescribed mean (trend) function $t(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$ (Rasmussen and Williams, 2006, Chapter 2.2):

$$\begin{aligned} t(\mathbf{x}) &= \mathbb{E}Y(\mathbf{x}), \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}\{(Y(\mathbf{x}) - t(\mathbf{x}))(Y(\mathbf{x}') - t(\mathbf{x}'))\}. \end{aligned}$$

When the mean function is assumed to be constant and unknown, the method is called *Ordinary Kriging* (OK) and is typically used in EGO. Now we wish to predict $f(\mathbf{x})$ at an unknown location $\mathbf{x} \in \mathrm{S}$. Without giving the derivation, the conditional distribution of Y on the observations $\mathbf{y}$ is a Gaussian distribution (Rasmussen and Williams, 2006):

$$Y(\mathbf{x}) \mid \mathbf{y} \sim \mathcal{N}\left(\hat{f}(\mathbf{x}), s^2(\mathbf{x})\right). \tag{2.13}$$

The conditional mean function $\hat{f}(\cdot)$ is used as the predictor for f while $s^2(\cdot)$ gives the MSE of the predictor $\hat{f}$. For the detailed discussion on Kriging/GPR, please see Chapter 3.

Step-wise risk Conceptually, EGO is a greedy step-wise search strategy. For instance, the model prediction can be set as the infill criterion, namely $\mathscr{A} \coloneqq \hat{f}$, giving the complete "trust" on the stochastic model. However, this is a highly *risky* action as the model is typically not accurate in the early stage of the optimization. To quantify the risk of the step-wise maximization of infill criteria, it is straightforward to calculate the *rate of failure*:

$$r = \Pr(f(\mathbf{x}) > f_{\min}) = 1 - \Pr(f(\mathbf{x}) < f_{\min}),$$

where $f_{\min} \coloneqq \min\{\mathbf{y}\}$ is the current minimal function value. Note that it is not feasible to calculate this rate due to the fact that there is a lack of the distribution information (e.g., which parametric family should be taken) about f, when assuming f is stochastic[1]. Thus, the typical approach is to approximate the

[1]Theoretically, this can be done by considering all the probabilistic models $\mathscr{M}$ (e.g., Gaussian/Student's t-process) for f and assuming a distribution over the models (e.g., a Dirichlet process). Then $r = \mathbb{E}\{\Pr(f(\mathbf{x}) > f_{\min} \mid \mathscr{M})\}$.

rate of failure under a specific distribution on f. When choosing the Kriging/GPR for f (Eq. (2.13)), the risk approximate is:

$$\hat{r} = 1 - \Pr(Y(\mathbf{x}) < f_{\min} \mid \mathbf{y}).$$

Note that $\Pr(Y(\mathbf{x}) < f_{\min} \mid \mathbf{y})$ is also a commonly used infill criterion, called *probability of improvement* (Eq. (4.6)). From the perspective of step-wise risks, it is interesting to compare EGO with the well-known *Simulated Annealing* (SA) algorithm (Agrawal et al., 1995):

- In SA, each candidate location $\mathbf{x}'$ that is worse than its parent $\mathbf{x}$ is accepted with the probability:

  $$r_{\text{SA}} = \exp\left(-\frac{f(\mathbf{x}') - f(\mathbf{x})}{t}\right),$$

  where $t \in \mathbb{R}_{>0}$ is the current temperature of SA. In other words, the step-wise risk of SA is r_{SA}.

- In EGO, each mutation $\mathbf{x}'$ obtained from Eq. (2.12) is always accepted and the step-wise risk of this action is $\hat{r}$.

From this conceptual comparison, it is obvious that EGO has no control over the step-wise risk if the probability of improvement is not chosen as the infill criterion. However, when using the probability of improvement, the resulting algorithm behaves very exploitative (see Section 4.2). To make a trade-off between exploitation and exploration, it is possible to enforce maximal risk (minimal probability of improvement) on the infill criterion maximization:

$$\begin{aligned} \arg\max_{\mathbf{x} \in \mathrm{S}} \quad & \mathscr{A}(\mathbf{x}; \Theta) \\ \text{subject to} \quad & \hat{r} < \upsilon, \end{aligned} \tag{2.14}$$

where υ is the threshold of the step-wise risk. It can be either determined by the user or controlled online as with r_{SA} in the Simulated Annealing. Intuitively, Eq. (2.14) pre-screens out highly risky regions in the search space. As will be described in Section 4.2, an alternative approach is to consider the step-wise risk and the other infill criterion as a *bi-objective* optimization task.

2.6 Summary

In this chapter, we discuss the stochastic variation operator, which is one of the most important component of stochastic optimization algorithms. Specifically, the so-called Gaussian sampling is re-visited: the sampling error of Gaussian random sampling could be very large when the sample size is quite small. The large sampling error could potentially reduce the efficiency of the stochastic variation. As a remedy, the mirrored orthogonal sampling is proposed to reduce the sampling error and therefore accelerate the convergence velocity for the small sample size. Apart from improving the existing stochastic variation operator, we manage to extract a stochastic variation operator from the well-known Efficient Global Optimization algorithm. The resulting operator is called mutation by optimization. In this manner, the EGO algorithm becomes conceptually similar to the canonical stochastic optimization algorithm, e.g., the Simulated Annealing.