Stochastic and deterministic algorithms for continuous black-box optimization

Wang, H.

CHAPTER 4

Infill Criteria

When using surrogate modeling in combination with optimization techniques, it is crucial to determine how the model should be explored/exploited properly due to the fact that surrogate models give rise to errors in the prediction. Firstly, it is possible to define a "gain" function $G : \mathrm{S} \to \mathbb{R}$ to assess the unknown locations (S is again the search space), e.g., the potential *improvement* over the current best fitness value. Secondly, considering the randomness from the surrogate model $\mathscr{M}$ (usually a statistical model), the "gain" function becomes stochastic and it is necessary to use some statistical features from it, e.g., the expectation:

$$\mathscr{A}(\mathbf{x}) = \mathbb{E}\left\{G(\mathbf{x}) \mid \mathscr{M}\right\}.$$

Such a function $\mathscr{A} : \mathrm{S} \to \mathbb{R}$ is the so-called **infill criterion**. Note that in some literature, it is also called *acquisition function* (Snoek et al., 2012). The next location to evaluate is simply the maximum of the infill criteria:

$$\arg\max_{\mathbf{x}\in\mathrm{S}} \mathscr{A}(\mathbf{x}). \tag{4.1}$$

This formalism depends on two design choices: the statistical model $\mathscr{M}$ and the "gain" function G. As for the statistical model, Kriging/GPR is very commonly applied as it provides a theoretical quantification for the modeling error, through the Kriging MSE (cf. Eq. (3.11)). Some other popular models include *random forests* (Hutter et al., 2011; Bartz-Beielstein et al., 2005) and *support vector regression* (SVR) (Forrester et al., 2008). For those models, however, the theoretical prediction error is typically not available and the *empirical error* is used instead. Here, Kriging/GPR shall be assumed as the statistical model throughout all the discussions.

As for the "gain" function, one common choice is the potential improvement over the current best fitness value, achieved by evaluating an unobserved location. Maximizing the expected improvement leads to a greedy, stepwise optimization strategy. As an alternative, the so-called *cumulative regret* is considered in the multi-armed bandit: the regret in iteration t is $R_t = f(\mathbf{x}^*) - f(\mathbf{x}_t)$, where $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathrm{S}} f(\mathbf{x})$ and $\mathbf{x}_t$ is the location chosen in iteration t. In practice, as $f(\mathbf{x}^*)$ is unknown, the regret R_t has to be estimated. It is possible to construct a non-stepwise gain by summing up all the regrets since the beginning of the optimization:

$$G = -\sum_{t=1}^{T} R_t.$$

Note that T stands for the current iteration and the minus sign is intended to convert the regret values to a gain function. For this gain function, the *Upper Confidence Bound* (UCB) criterion is proposed (Auer, 2002): given a surrogate model $\hat{f}$ (e.g., Kriging) of f and the MSE of the prediction $s^2(\mathbf{x})$, UCB is defined as:

$$\mathrm{LCB}(\mathbf{x};\beta) = \hat{f}(\mathbf{x}) - \sqrt{\beta s^2(\mathbf{x})}, \tag{4.2}$$

where β is a carefully chosen learning rate that explicitly controls the trade-off between exploitation and exploration. Note that this infill criterion is also known as *Lower Confidence Bound* (LCB) in terms of minimization. Obviously, a high value of β emphasizes more on the model uncertainty and thus tends to be more explorative. When Kriging/GPR is chosen as the statistical model, this infill criterion is called Gaussian Process Upper Confidence Bound (GP-UCB) (Srinivas et al., 2010). In addition, criteria with free parameters such as β shall be called *parameterized infill criteria* in this thesis. Other infill criteria have been proposed, based on different types of the gain function, e.g., *BayesGap* (Hoffman et al., 2014) and *UGap* (Gabillon et al., 2012), which are gap-based exploration approaches. For a survey and conceptual comparison among those infill criteria, please see Jones (2001); Forrester et al. (2008).

The infill criterion plays a vital role in many optimization paradigms, including the Efficient Global Optimization, Multi-armed Bandits, Monte-Carlo Tree Search (MCTS) and Multi-objective optimization (Emmerich, Yang, Deutz, Wang, and Fonseca, 2016). In this thesis, we shall focus on the Efficient Global Optimization algorithm, which is a sequential design strategy designed to solve expensive global optimization problems (Močkus, 1975, 2012; Jones et al., 1998).

It is important to point out that different infill criteria can give rise to conflicts on the promising location to evaluate, e.g., the probability of improvement favors low-risk locations while the expected improvement tends to find locations with a high gain (see Section 4.2). Multiple conflicting infill criteria can be considered as a multi-objective optimization problem (Bischl et al., 2014; Wang et al., 2016). Such a multi-objective treatment gives the decision makers the flexibility to choose among low-risk and/or high-gain solutions and possibly leads to parallelization of the Bayesian optimization. An alternative approach is proposed by Hutter et al. (2012), where multiple LCB criteria are created by sampling several β values from an exponential distribution with the unit mean. Furthermore, it is proposed to use portfolio strategies to select an infill criterion in each step of the Efficient Global Optimization (Hoffman et al., 2011; Ursem, 2014).

As for the optimization of the infill criteria, derivative-free Evolutionary algorithms (Bäck and Schwefel, 1993) and gradient-based method (e.g., quasi-Newton method) are often used/combined to search for the global optimum. As an alternative, Wang et al. (2018) propose to diversify the search by adapting the niching techniques to find multiple (local) optima of the acquisition function.

4.1 Improvement-based Infill Criteria

Over the last decades, much research has been put into finding a function $\mathscr{A}$ that provides a good balance between exploration and exploitation for various applications. One category of such functions, called *Improvement-based infill criteria*, is of particular interest for the following reasons: this category of functions has clear statistical meanings and those are widely applied in the field of efficient global optimization. Formally, the improvement is a function[1] defined on the stochastic process model Y of the objective function f. In terms of minimization, it is (Schonlau et al., 1998):

$$I(\mathbf{x}) = \begin{cases} f_{\min} - Y(\mathbf{x}) & \text{if } Y(\mathbf{x}) < f_{\min}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

Assume that we have observed the data set $(\mathrm{X}, \mathbf{y})$ on f and let $f_{\min} = \min\{\mathbf{y}\}$ stand for the best function value found so far. When choosing the Gaussian prior on Y, the posterior process $Y(\mathbf{x}) \mid \mathbf{y} \sim \mathcal{N}(\hat{f}(\mathbf{x}), s^2(\mathbf{x}))$ (cf. Eq. (3.29)) is taken

[1] Naturally, this function I is also a stochastic process over S.

in Eq. (4.3). In the following discussion, the posterior mean and variance shall be abbreviated as $\hat{f}$ and s^2 if there is no ambiguity on the location $\mathbf{x}$ under the consideration. The distribution of $I(\mathbf{x}) \mid \mathbf{y}$ is known as *Rectified Gaussian*[1], whose density function is written as (cf. Eq. (A.5)):

$$p_I(u;\mathbf{x}) = \begin{cases} \Phi\left(\dfrac{\hat{f} - f_{\min}}{s}\right)\delta(u) & u < f_{\min}, \\ s^{-1}(2\pi)^{-1/2}\exp\left(-\dfrac{(u-(f_{\min}-\hat{f}))^2}{2s^2}\right) & \text{otherwise.} \end{cases} \tag{4.4}$$

Here $\delta(\cdot)$ is the Dirac delta. Most of the improvement-based infill criteria are constructed to summarize the statistical properties of the improvement. A short review of the improvement-based infill criteria is given as follows.

- *Expected Improvement* (EI) is originally proposed by Močkus (1975) and utilized as the infill criterion in the standard *Efficient Global Optimization* (EGO) algorithm (Jones et al., 1998). It is defined as the first moment of the improvement:

$$\mathrm{EI}(\mathbf{x}) = \mathbb{E}\left\{I(\mathbf{x}) \mid \mathbf{y}\right\} = \left(f_{\min} - \hat{f}\right)\Phi\left(\frac{f_{\min} - \hat{f}}{s}\right) + s\phi\left(\frac{f_{\min} - \hat{f}}{s}\right). \tag{4.5}$$

  Here $\Phi(\cdot)$ and $\phi(\cdot)$ are the cumulative distribution function (c.d.f.) and probability density function (p.d.f.) of the standard normal random variable. As will be shown in the next section, the EI criterion is highly multi-modal and tries to balance between exploration and exploitation.

- *Bootstrapped Expected Improvement* (BEI) (Kleijnen et al., 2012) tries to correct the bias in EI due to the fact that the Kriging MSE derived in Eq. (3.11) is an *underestimate* of the true Kriging MSE when the hyper-parameters are estimated (den Hertog et al., 2006). Bootstrapped EI uses parametric bootstrapping to approximate the real Kriging MSE. Although BEI is shown as a more reliable alternative to EI, it also brings a large amount of computational cost.

- *Probability of Improvement* (PI) gives the probability of realizing an improvement (Žilinskas, 1992; Jones, 2001). This criterion is more biased towards exploitation than exploration since it rewards the solutions that are more

[1] Do not confuse the rectified Gaussian with the so-called truncated Gaussian distribution. See Appendix A for the clarification.

certain to yield an improvement over the current best solution, without taking the amount of the actual improvement into account:

$$\mathrm{PI}(\mathbf{x}) = \Pr(Y(\mathbf{x}) < f_{\min} \mid \mathbf{y}) = \Phi\left(\frac{f_{\min} - \hat{f}}{s}\right). \tag{4.6}$$

Loosely speaking, PI rewards low risk solutions that typically come with a relatively small amount of improvement while EI rewards solutions that give high improvement on average but could be risky to realize such an improvement. Therefore, the maximization of PI is considered as a risk minimization strategy and in contrast, the maximization of EI is considered as a gain maximization strategy. In Section 4.2, the trade-off between these two criteria is investigated by treating them as a bi-objective optimization problem.

- *Weighted Expected Improvement* (WEI) is an alternative approach to explicitly control the balance between exploration and exploitation. Consider Eq. (4.5): the first term calculates the difference between the current best $f_{\min}$ and the prediction $\hat{f}$, penalized by the probability of improvement. The second term is large when the RMSE s is large, meaning a large uncertainty about the prediction is preferred. Therefore, it is also straightforward to balance those two terms within EI (Sóbester et al., 2005):

$$\mathrm{WEI}(\mathbf{x}; w) = w\left(f_{\min} - \hat{f}\right)\Phi\left(\frac{f_{\min} - \hat{f}}{s}\right) + (1-w)s\phi\left(\frac{f_{\min} - \hat{f}}{s}\right), \tag{4.7}$$

  where $w \in [0, 1]$ is the balancing weight. Sóbester et al. (2005) argues that this additional parameter is problem-dependent.

- *Generalized Expected Improvement* (GEI) (Schonlau et al., 1998) is a generalization of the EI criterion where an additional parameter, g, is introduced to compute the gth-order moment of the improvement. The larger the value of g, the more explorative locations will be awarded by the criterion and vice versa. GEI is defined as follows:

$$\mathrm{GEI}(\mathbf{x}; g) = \mathbb{E}\left\{I(\mathbf{x})^g \mid \mathbf{y}\right\} = s^g \sum_{k=0}^{g} (-1)^k \binom{g}{k} \hat{f}^{g-k} T_k, \tag{4.8}$$

  where T_k is defined recursively for $k > 1$:

$$T_k = -u^{k-1}\phi(u) + (k-1)T_{k-2}, \tag{4.9}$$

with

$$u = \frac{f_{\min} - \hat{f}}{s}, \quad T_0 = \Phi(u), \quad T_1 = -\phi(u)$$

Note that as GEI is expressed recursively, additional computational costs are attached to it and moreover it becomes very difficult to differentiate GEI. The setting of the additional integer parameter g is entirely empirical. Sasena et al. (2002) propose a "Simulated Annealing"-like approach to decrease the value of g gradually, resulting in highly explorative behavior in the beginning of the optimization and more exploitative behavior after several iterations. The settings for g proposed are in the form of a look-up table where g starts at 20 and quickly goes down to 0 after iteration 35.

- *Multiple Generalized Expected Improvement* (MGEI) (Ponweiser et al., 2008) instantiates multiple normalized GEI criteria, using different g settings in parallel. They obtain k best local optima which are evaluated for the next iteration. The main disadvantage of this approach is the need of a large number of evaluations.

4.2 Balancing Risk and Gain

In either efficient global optimization or experimental design, a single infill criterion is maximized to generate the next promising location (solution) to evaluate. The choice of criteria has a significant impact on the performance because each criterion represents a different optimization strategy. When considering PI and EI, on the one hand, a low PI value can be viewed as risky evaluation and the maximization of PI can be seen as a *risk minimization strategy*. However, the risk minimization strategy can still lead to small improvement. On the other hand, the maximization of EI leads to high gains on average and thus it is a strategy of *expected gain maximization*. However, it does not minimize the rate of failure (no improvement is found). In Wang et al. (2016), we propose to consider the maximization of PI and EI simultaneously to find the trade-off between them, which naturally forms a *bi-objective* optimization problem. In this manner, it is possible to search for the best set of trade-off locations, namely the Pareto efficient set and candidate locations can be selected from the Pareto efficient set according to human preferences on risk/expected gain, or some pre-defined decision-making rules. Formally, the

following *vector-valued* infill criterion is considered:

$$\mathscr{A} : \mathrm{S} \to \mathbb{R}^2, \quad \mathbf{x} \mapsto (\mathrm{PI}(\mathbf{x}), \mathrm{EI}(\mathbf{x}))^\top. \tag{4.10}$$

The *set* of candidate locations is the Pareto efficient set of $\mathscr{A}$, namely:

$$\mathcal{X} = \arg\max_{\mathbf{x} \in \mathrm{S}} \mathscr{A}(\mathbf{x}). \tag{4.11}$$

Note that such a vector-valued infill criteria can be easily generalized to many objectives. Before solving Eq. (4.11) numerically, the following observations are seen in the objective space, i.e., the space formed by PI and EI values. As shown in the definition of PI and EI (Eq. (4.6) and (4.5)), both of them can be expressed completely by the Kriging predictor $\hat{f}$ and root mean squared error (RMSE) s. Therefore, both PI and EI can be considered in the space formed by $\hat{f}$ and s. Some properties of PI and EI can be inferred by investigating their behavior in such space with the benefit that such approach is independent of a specific Kriging model. The contour lines of two criteria are shown in Fig. 4.1. Without loss of generality, we put the current best function value $f_{\min}$ to zero and normalize $\hat{f}, s$ to $[-1, 1]$ and $[0, 1]$, respectively. In the figure, the maximum of PI and EI is illustrated by

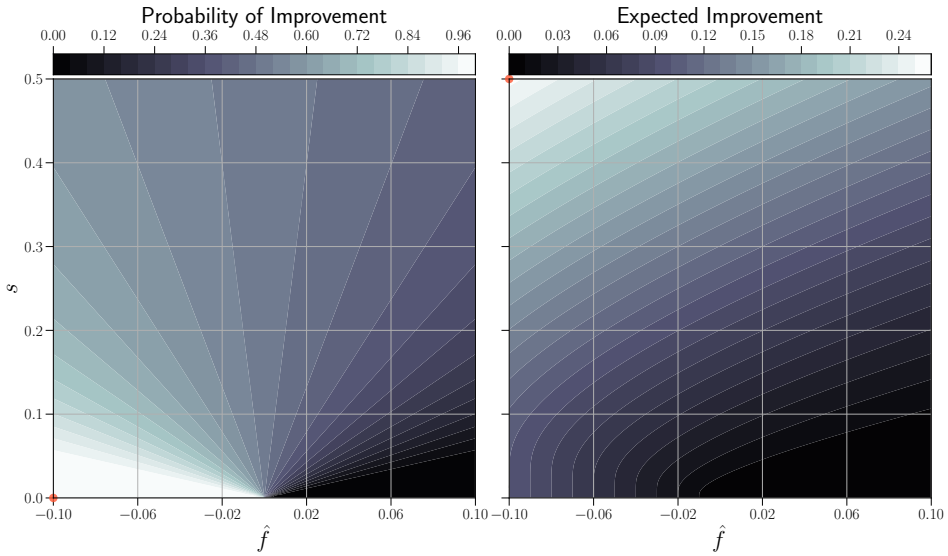


Figure 4.1: The contour lines of Probability of Improvement (Left) and Expected Improvement (Right) in the space of Kriging prediction and RMSE. The red points indicate the maximum of PI and EI, respectively.

the red point. Furthermore, $\hat{f} < 0$ indicates predictions that are smaller (better) than $f_{\min}$. In this case, PI tends to find the point with the minimal RMSE. According to the Kriging MSE expression (Eq. (3.11)), the RMSE s decreases to zero when approaching any observed location. Therefore, when better predictions exist in the Kriging model, PI prefers locations "next to" the observed ones. When Kriging predictions are worse than $f_{\min}$, PI prefers the point with the largest RMSE and thus is of high risk. In contrast, EI always increases with increasing

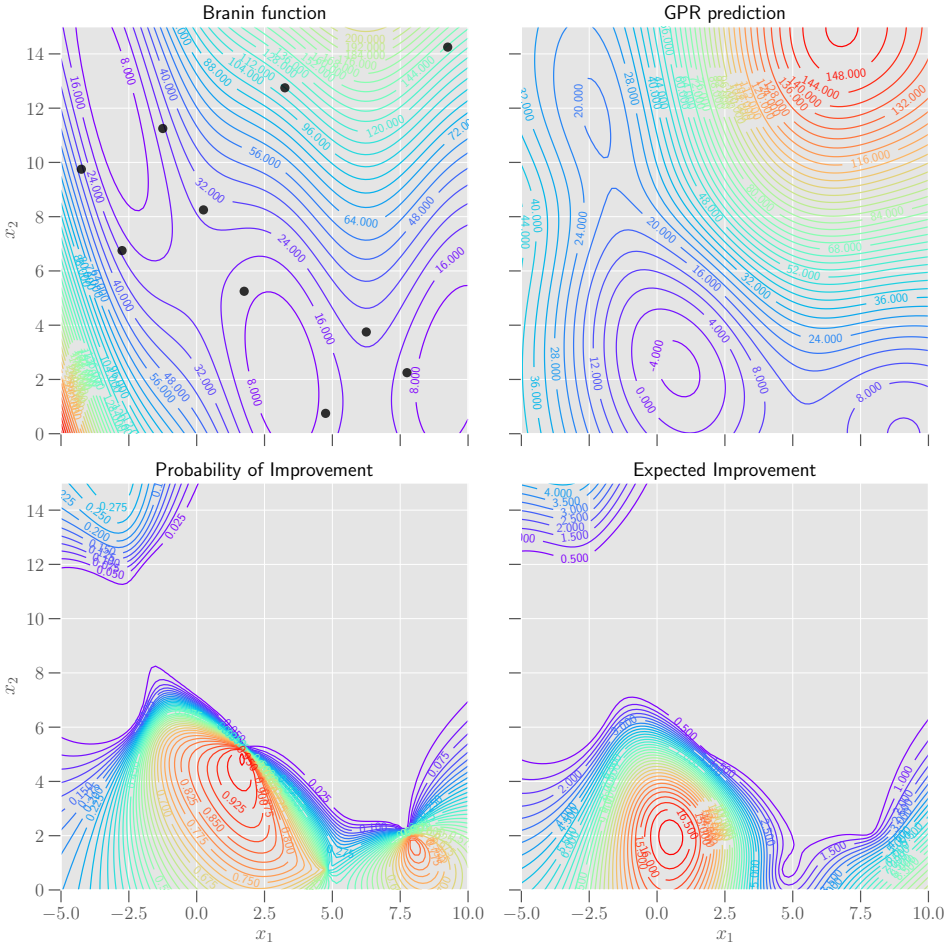


Figure 4.2: On the *Branin function* (**top-left**), 10 data points (black dots) are generated uniformly, on which an *Ordinary Kriging* model (with Matérn 3/2 kernel) is trained. Contour lines of the prediction (**top-right**), Expected Improvement (**bottom-left**) and Probability of Improvement (**bottom-right**) are depicted.

RMSE and decreases with increasing Kriging predictions. Note that, although it seems possible to compute the Pareto front directly from Fig. 4.1 (by looking for the points that satisfy the Karush-Kuhn-Tucker (KKT) conditions based on the contour line of PI and EI), the resulting Pareto front is not feasible due to the fact that not every point in the objective space is *attainable*: depending on the underlying Kriging/GPR model, the pre-image of a point $(\hat{f}, s) \in [-1, 1] \times [0, 1]$ does not necessarily exist in the search/decision space S.

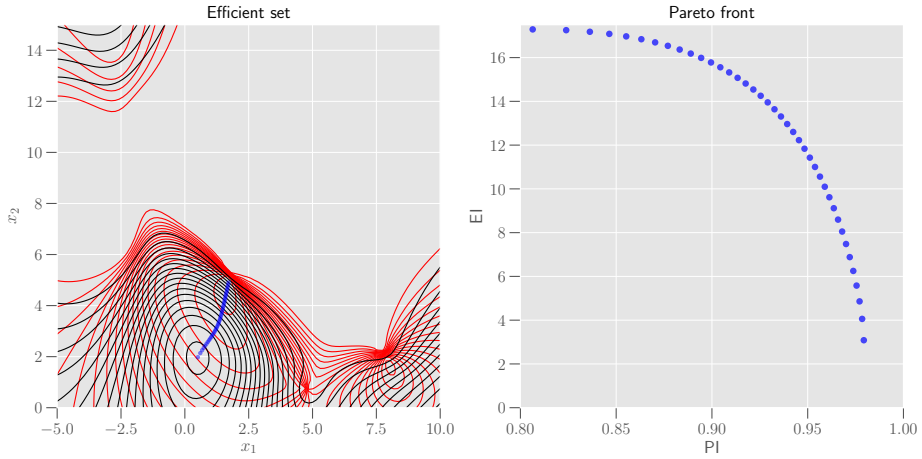


Figure 4.3: Pareto approximation sets of the bi-objective infill criterion formed by the PI and EI landscape (Fig. 4.2). The results are obtained from the HIGA-MO with population size 20. **Left**: Pareto efficient sets (blue dots) between PI (red contours) and EI (black contours). **Right**: The corresponding Pareto fronts.

In Wang et al. (2016), it is proposed to solve this problem numerically by a gradient-based multi-objective optimization algorithm, called *Hypervolume Indicator Gradient Ascent Multi-objective Optimization* (HIGA-MO) (Wang et al., 2017). The gradients of PI and EI are required for this algorithm, which are given as follows. By introducing the auxiliary variable $u = (f_{\min} - \hat{f})/s$, the gradients are:

$$\frac{\partial \,\mathrm{PI}}{\partial \mathbf{x}} = -\frac{\phi(u)}{s}\left(\frac{\partial \hat{f}}{\partial \mathbf{x}} + u\frac{\partial s}{\partial \mathbf{x}}\right), \quad \frac{\partial \,\mathrm{EI}}{\partial \mathbf{x}} = \phi(u)\frac{\partial s}{\partial \mathbf{x}} - \Phi(u)\frac{\partial \hat{f}}{\partial \mathbf{x}}.$$

By plugging those gradients into HIGA-MO (Alg. 10 in Section 5.2), the Pareto front/efficient set of $\mathscr{A}$ can be obtained. We illustrate the result of the bi-objective optimization on the well-known 2-D *Branin* function. Here 10 design locations are generated and evaluated on the Branin function using Latin Hypercube Sampling

and an Ordinary Kriging is built on those locations. The landscapes of PI and EI are shown in Fig. 4.2: the global structures of the landscapes are quite similar while the local landscape differs subtly. The following setting of HIGA-MO is used for this problem: the population size is set to 40 and the step-size is set to 0.004 multiplied by the maximal range of the search space. HIGA-MO is terminated after finishing 2000 generations. The results from the gradient-based algorithm are shown in Fig. 4.3, where the efficient set and the Pareto front are indicated by the blue points.

4.3 Moment-Generating Function of Improvement

Following the intuition on using the higher moments of the improvement, an novel infill criterion based on the *Moment-Generating Function* (MGF) of the improvement is introduced in (Wang et al., 2017), where all the moments are combined. Loosely speaking, given the existence of the moment-generating function, it can be expanded as a Taylor series, whose terms are proportional to all the moments (to the infinite order) of the improvement. Therefore, such a function is considered as combination of all the moments. Formally, the MGF of the improvement $I(\mathbf{x})$ is an alternative way to give its probability distribution and it is defined as:

$$\forall t \in \mathbb{R}, \quad M(\mathbf{x}, t) := \mathbb{E} \exp\left(tI(\mathbf{x})\right) = \int_{-\infty}^{\infty} e^{tu} p_I(u; \mathbf{x}) \, \mathrm{d}u.$$

Moreover, the moment-generating function can be calculated using the density function of $I(\mathbf{x})$:

$$\begin{aligned} M(\mathbf{x}, t) = 1 + \Phi\left(\frac{f_{\min} - \hat{f}'}{s}\right) \exp\left(\left(f_{\min} - \hat{f}\right) t + \frac{s^2 t^2}{2}\right) - \Phi\left(\frac{f_{\min} - \hat{f}}{s}\right), \\ \hat{f}' = \hat{f} - s^2 t. \end{aligned} \tag{4.12}$$

This function has a closed form and is well-defined for all $t \in \mathbb{R}$. From a different perspective, the Taylor expansion of the MGF is:

$$M(\mathbf{x}, t) = 1 + t\mathbb{E}I(\mathbf{x}) + \frac{t^2}{2!}\mathbb{E}I^2(\mathbf{x}) + \frac{t^3}{3!}\mathbb{E}I^3(\mathbf{x}) + \cdots = \sum_{n=0}^{\infty} \frac{t^n}{n!}\mathbb{E}I^n(\mathbf{x}). \tag{4.13}$$

Note that, for an arbitrary distribution, the above series might not converge for all the $t \in \mathbb{R}$, even if all the moments exist. When treating $t^n/n!$ as the weight for

each moment $\mathbb{E}I^n$, this function can also be considered as a linear combination of the moments, where the weights are controlled by variable t. In addition, it is possible to normalize the weights by observing the fact that: $\sum_{n=0}^{\infty} \frac{t^n}{n!} = e^t$, which converges for all $t \in \mathbb{R}$. Thus, the normalized MFG function[1] is obtained by dividing it by e^t. The additional parameter t controls the trade-off between exploration and exploitation of the search. To visualize this, multiple sets of weights are plotted in Fig. 4.4 by varying t. According to the figure, a low value of t (e.g., $t < 1$) assigns more weights to the lower moments (e.g., the expected improvement), rendering the search process mainly exploitative. As for the higher values of t, the "center" (mean) of the weight distribution is t and dispersion (variance) is also increasing with respect to t. This indicates that more higher moments of the improvement are taken into account when t increases and therefore the search tends to be more explorative.

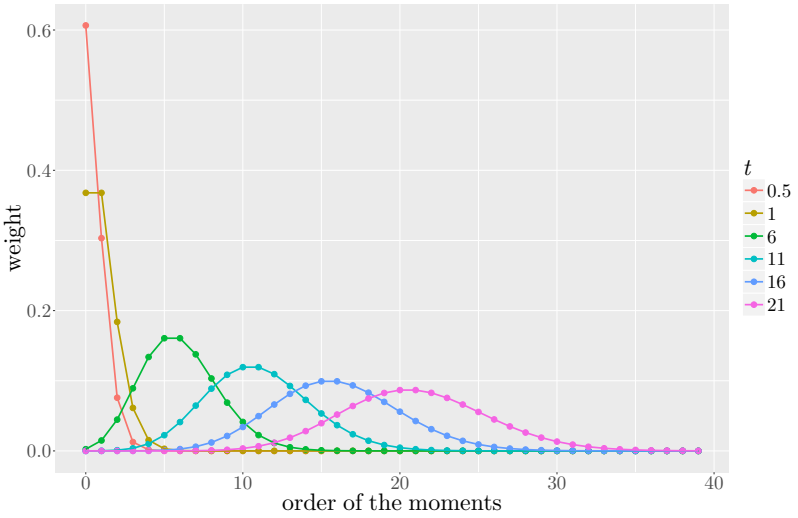


Figure 4.4: Distribution of the combination weights in the normalized moment-generating function by varying the t value from 0.5 to 21.

Finally, it is proposed to incorporate the Probability of Improvement (PI) in the proposed acquisition function. This is achieved by treating PI as the "zero-order" moment of $I(\mathbf{x})$ and replacing the constant 1 in Eq. (4.13). Putting all

[1] In fact, the normalized weight, $t^n/e^t n!$ is exactly the probability mass function of the Poisson distribution.

the considerations together, the proposed **Moment-Generating Function of Improvement** $\mathcal{M}$ (MGFI) is defined as:

$$\begin{aligned}\mathcal{M}(\mathbf{x};t) &= \frac{M(\mathbf{x},t) - 1 + \mathrm{PI}(\mathbf{x})}{e^t} \\ &= \mathrm{PI}(\mathbf{x}) + \frac{t}{e^t}\mathbb{E}I(\mathbf{x}) + \frac{t^2}{2!e^t}\mathbb{E}I^2(\mathbf{x}) + \frac{t^3}{3!e^t}\mathbb{E}I^3(\mathbf{x}) + \cdots \\ &= \Phi\left(\frac{f_{\min} - \hat{f}'}{s}\right)\exp\left(\left(f_{\min} - \hat{f} - 1\right)t + \frac{s^2t^2}{2}\right) \end{aligned} \tag{4.14}$$

where $\hat{f}'$ is defined in Eq. (4.12). In order to align with existing work (Wang et al., 2016) on using gradient-based optimization techniques for infill criteria, the gradient of MGFI is given as well:

$$\begin{aligned}\frac{\partial\mathcal{M}(\mathbf{x};t)}{\partial\mathbf{x}} = C\Bigg[&\Phi\left(\frac{f_{\min} - \hat{f}'}{s}\right)\left(t^2 s\frac{\partial s}{\partial\mathbf{x}} - t\frac{\partial\hat{f}}{\partial\mathbf{x}}\right) \\ &- \phi\left(\frac{f_{\min} - \hat{f}'}{s}\right)\left(\frac{1}{s}\frac{\partial\hat{f}'}{\partial\mathbf{x}} + \frac{f_{\min} - \hat{f}'}{s^2}\frac{\partial s}{\partial\mathbf{x}}\right)\Bigg],\end{aligned}$$

where

$$\frac{\partial\hat{f}'}{\partial\mathbf{x}} = \frac{\partial\hat{f}}{\partial\mathbf{x}} - 2ts\frac{\partial s}{\partial\mathbf{x}}, \quad C = \exp\left(\left(f_{\min} - \hat{f} - 1\right)t + \frac{s^2t^2}{2}\right).$$

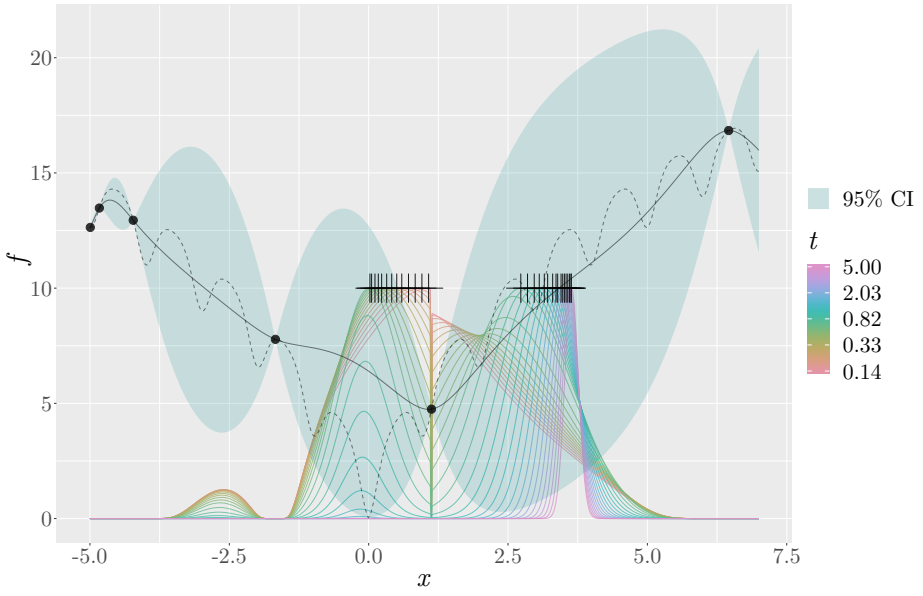
The gradients $\partial\hat{f}/\partial\mathbf{x}$ and $\partial s/\partial\mathbf{x}$ are expressed in Eq. (3.34) and (3.35).

Comparison to GEI As with the Generalized Expected Improvement, MGFI is designed to exploit the higher moments of the improvement. Compared to GEI, the main advantages of $\mathcal{M}$ are:

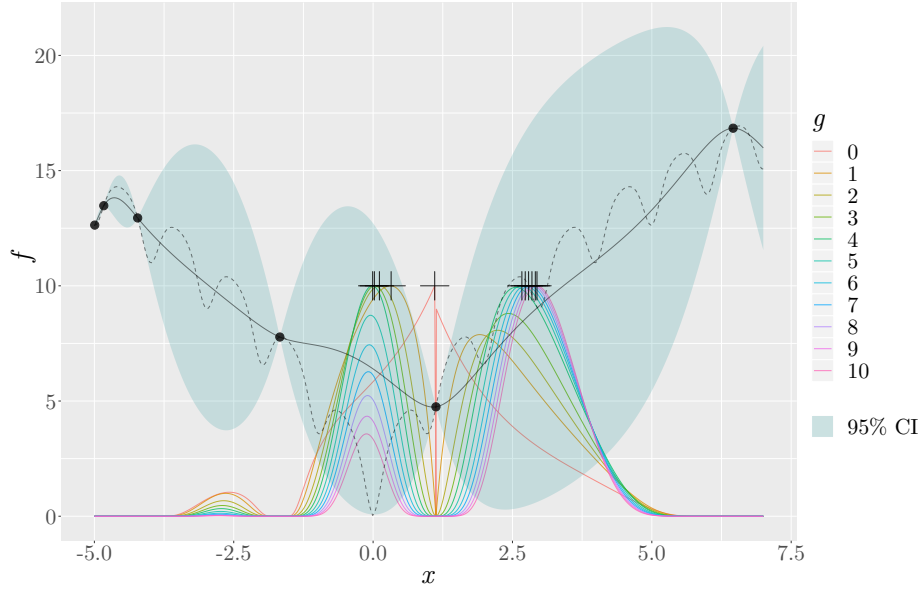
1. $\mathcal{M}$ combines all the moments using a weight distribution instead of using one moment each time in GEI. This leads to a much smaller "change" in the acquisition function when tuning the addition parameter t.

2. It has a simple closed-form expression, in contrast to a recursive formula (Eq. (4.9)) for GEI. As a result, it is obvious to see that $\mathcal{M}$ is computationally less expensive than GEI. The gradient of $\mathcal{M}$ can also be easily calculated.

3. The extra parameter t that balances the exploration and exploitation, takes continuous values while the parameter g in GEI is an integer variable. Consequently, when tuning this additional parameter, the effect can be more smooth: cooling schedules such as, e.g., the exponential decay, can be applied.



(a) MGF of Improvement (MGFI)



(b) Generalized Expected Improvement (GEI)

Figure 4.5: On the 1-D *Ackley function* (dashed curve), a GPR model (solid curve) is built on the black dots. MGFI and GEI (both normalized and rescaled to $[0, 10]$) are plotted by varying parameters t and g, whose maxima are indicated by black crosses. The shaded area shows the 95% confidence interval of the prediction.

The difference between MGFI and GEI is illustrated in Fig. 4.5. On the 1-D Ackley function, a GPR model is built on 6 uniformly distributed samples (black dots) in $[-5, 7]$. In addition, 11 MFGI functions are created using a log-scaled t value from roughly 0.1 to 3 while 11 GEI functions are depicted with g from 0 to 10. The maximum of the infill criteria (the black crosses) is evaluated in the next iteration. Comparing the spread of those maximum points, it is obvious that when g increases in GEI, those maxima of infill criteria are getting close to each other and thus become indistinguishable. This means the GEI functions are, in fact, indifferent when the parameter g increases. However, for MGFI, those maxima still show significant differences when t increases. Therefore, the *effective range* of the parameter g is much narrower than that of t.

4.4 Cooling Strategies for MGFI

As discussed in the last section, the functionality of parameter t is analogous to that of temperature in simulated annealing (Nourani and Andresen, 1998). Consequently it is straightforward to use the temperature cooling strategies to improve the optimization procedure. In Wang et al. (2018), we propose to adopt two most commonly applied cooling strategies in simulated annealing, namely (Nourani and Andresen, 1998):

- Exponential strategy: $t_{i+1} = \alpha t_i, \quad 0 < \alpha < 1.$
- Linear strategy: $t_{i+1} = t_i - \eta, \quad \eta > 0.$

In each cooling strategy, naturally, there are two parameters to set: the initial temperature t_0 and the cooling speed (α for exponential strategy and η for the linear one). As shown in the next subsection, the initial temperature t_0 is a free parameter, whose setting should be highly problem-dependent. As for the cooling speed, it should be determined with respect to the prescribed function evaluation budget. For example, a fast cooling speed for a short run length can be hazardous because the search will quickly become very exploitative and even ends up with stagnation. However, instead of setting the cooling speed directly, the temperature t_f at the final iteration of the algorithm[1] is of interest here due to the fact that the moment whose order is the closest to the current temperature

[1]Note that the algorithm might terminate before consuming all the budget if other termination criteria are implemented and satisfied.

has the biggest weight in MGFI (see Fig. 4.4). Therefore, t_f determines the major functioning order of moment in the final stage of the search. Let $N_{\max}$ be the maximal number of iterations. The cooling speed parameters are determined as follows: $\alpha = (t_f/t_0)^{1/N_{\max}}, \eta = (t_0 - t_f)/N_{\max}$.

4.4.1 Impact of Temperature Configurations

Intuitively, the optimal settings for the initial and final temperature t_0, t_f should depend on the specific problem. An experiment is performed to investigate how large the impact of temperature configurations can be on test problems. Here 10 different temperature configurations (shown in Tab. 4.1) are generated from the Latin hypercube sampling with design ranges: $t_0 \in [1, 5]$ and $t_f \in [10^{-3}, 0.5]$.

Table 4.1: Latin hypercube design of the temperature configuration.

No.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
t_0	1.084	1.273	1.495	1.756	2.063	2.423	2.847	3.344	3.928	4.613
t_f	0.366	0.197	0.005	0.009	0.016	0.001	0.031	0.106	0.057	0.003

The following experiment is carried out on noiseless functions of the BBOB benchmark (Hansen et al., 2010, 2009) for the *exponential cooling strategy*. We only select the 10 multi-modal functions $f15$-$f24$ from BBOB. The unimodal functions are skipped because efficient global optimization is designed for multi-modal functions and using a high temperature (high explorative effect) usually leads to inefficient convergence on a unimodal one. In addition, 30 independent runs are conducted on each test function. The function evaluation budget is set to $50 \times D$ and the size of the initial LHS design is fixed to $10 \times D$. For $D = 2$, the performance of each temperature configuration is reported in Fig. 4.6, where *empirical cumulative distribution functions of the running length (ECDFs)* are shown. ECDF measures the success probability (if the algorithm reaches a given target fitness $f^* + \Delta f$ in one run) as a function of maximal number of function evaluations allowed. On some functions, ECDFs from all the temperature configurations differ largely, e.g., $f19$ and $f23$. However, on $f17$, ECDFs are quite similar to each other, implying the temperature configuration has relatively small impact for $f17$. Moreover, by checking the winning configuration on each function, it is obvious that there is no global winner for this multi-modal function class. This

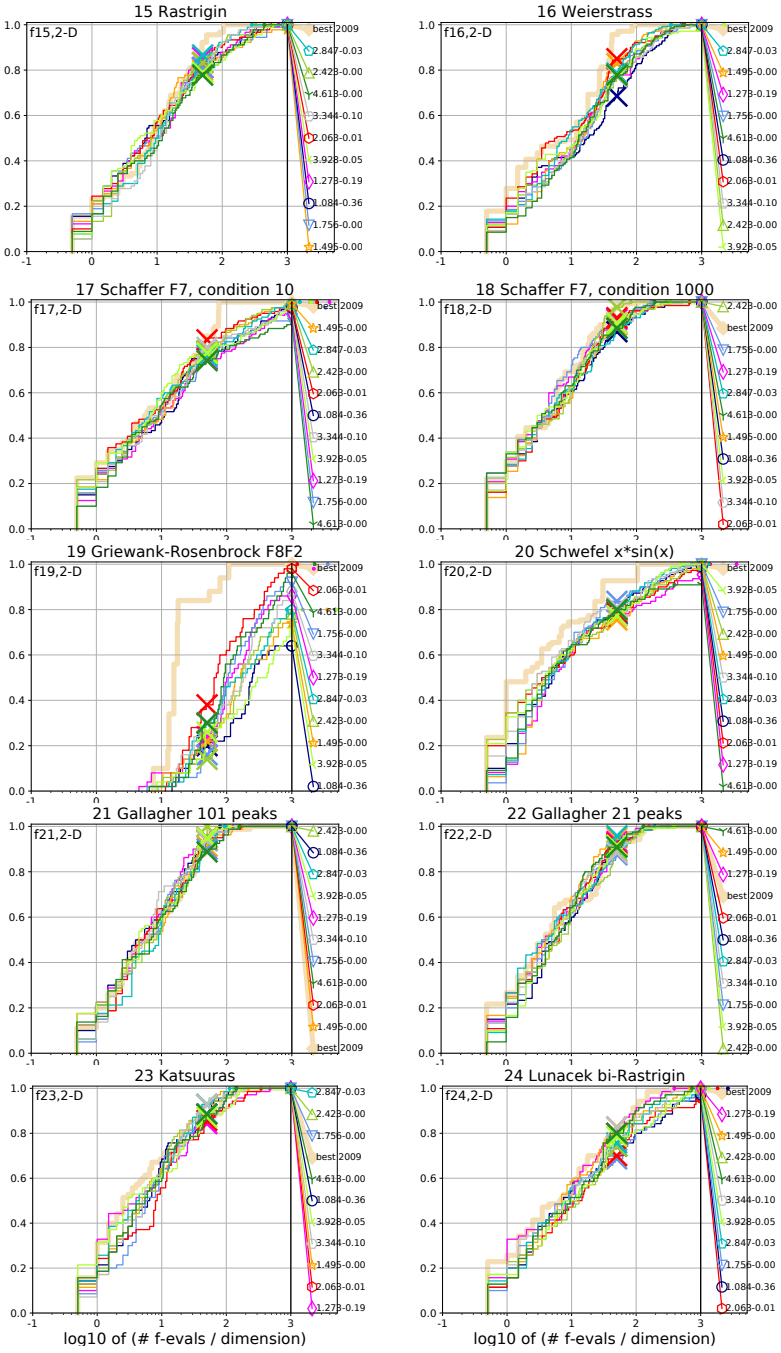


Figure 4.6: Bootstrapped empirical cumulative distribution of the number of function evaluations divided by dimension for all functions in 2-D. The targets are chosen from $10^{[-8..2]}$ such that the bestGECCO2009 artificial algorithm just did not reach them within a given budget of $k \times$ DIM, with $k \in \{0.5, 1.2, 3, 10, 50\}$.

observation suggests that the temperature settings are highly problem-dependent and should indeed be configured in practice.

4.4.2 Benchmarking the Cooling Strategies

For the experiments, only the multi-modal test functions $f15 - f24$ are picked for the same reason. Both exponential and linear cooling strategies are tested. Note that the temperature configuration is not investigated here due to the high computational time it takes. Instead, the following setting is used: $t_0 = 2, t_f = 0.1$. A relatively small function budget is chosen because it is the scenario on which Bayesian optimization should perform well: the size of the initial design is set to $10 \times D$ and the totally function evaluation is limited to $50 \times D$. The maximal function evaluation is set as the only termination criterion in the experiment. Therefore, $40 \times D$ iterations are executed in each run. For each test function, 30 instances are created for independent runs. For the settings of the Gaussian process, the Matérn 3/2 kernel is used throughout the experiment. The model is fitted via the maximum likelihood method, which is solved by the Limited-memory BFGS (L-BFGS) algorithm with the restarting heuristic. In addition, a small number of function evaluations $8 + \lfloor 40 \log D \rfloor$ is set for L-BFGS because the computational overhead on the likelihood function explodes quickly as the number of evaluations goes up.

On $D = 10$, the benchmark results are shown in Fig. 4.7. In the plot, "MGFI" indicates the application of MGFI criterion with a constant temperature setting of 1 (no cooling strategy in this case). "MGFI-exp" stands for MGFI with exponential cooling strategy while "MGFI-linear" is for the linear cooling. Note that all the infill-criteria fail completely on $f19$, indicating many more function evaluations are needed to solve this function. Using a linear or exponential cooling strategy, MGFI outperforms the commonly applied expected improvement criterion on seven functions out of $f15 - f24$. This comparison suggests that MGFI with a cooling strategy is preferable for BBOB multi-modal functions. In addition, there is no clear winner between the linear and exponential cooling strategies. However, on function $f21$, EI wins the competition. This situation might be caused by a very poor setting of the cooling temperature and requires further investigation.

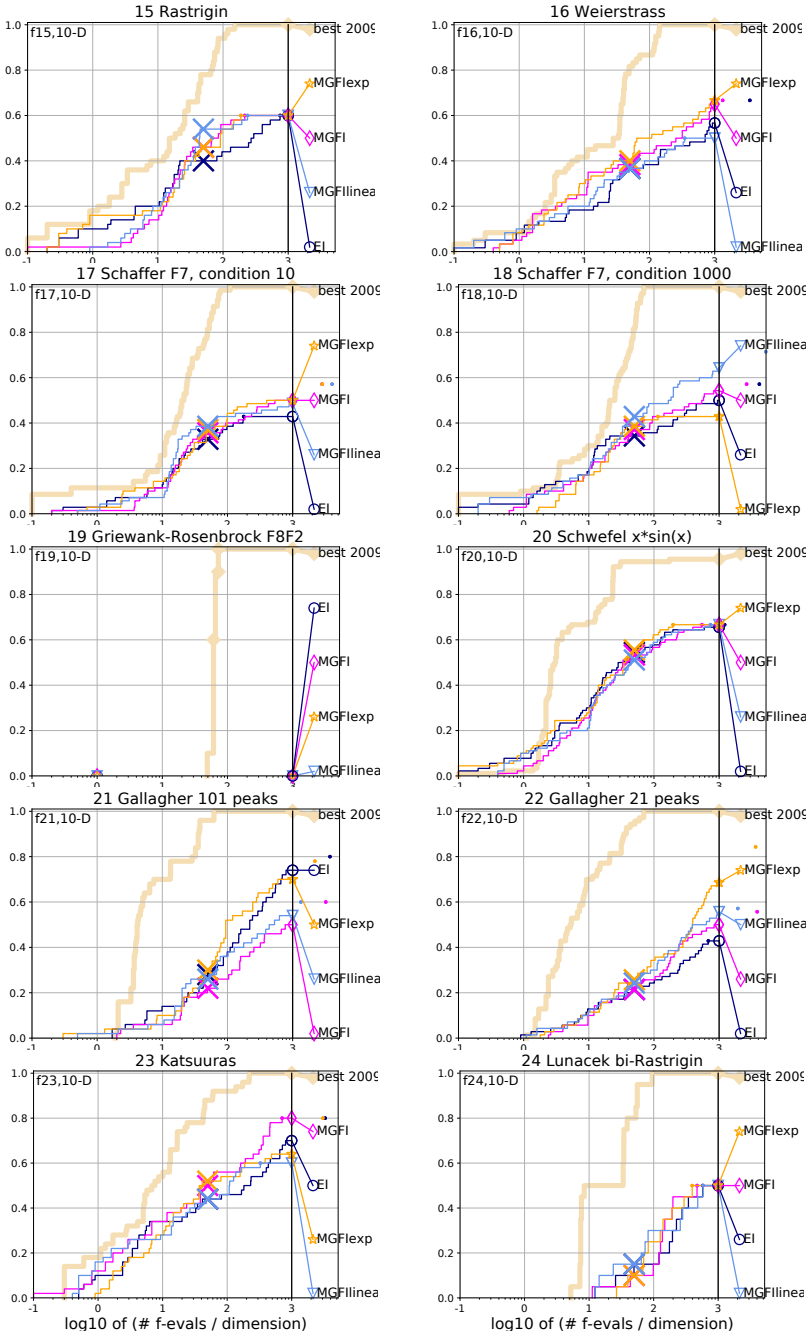


Figure 4.7: Bootstrapped empirical cumulative distribution of the number of function evaluations divided by dimension for all functions in 10-D. The targets are chosen from $10^{[-8..2]}$ such that the bestGECCO2009 artificial algorithm just did not reach them within a given budget of $k \times$ DIM, with $k \in \{0.5, 1.2, 3, 10, 50\}$.

4.5 Parallelization

In some applications, the target function f to optimize is actually represented by a time-consuming simulator run. Multiple simulators can be distributed over many CPUs or a grid of machines, allowing for the parallelization of function evaluations. In order to take advantage of this parallelism, several promising candidate locations are needed from infill criteria $\mathscr{A}$, in addition to the maximum of $\mathscr{A}$. For instance, naively, this can be done by randomly sampling q points ($q > 1$) in addition to $\mathscr{A}$'s maximum (Hutter et al., 2010). However, this method might be *inefficient* because many random samples would never be chosen for the evaluation if a sequential q-step maximization of infill criteria were performed. Although parallelization methods for infill criteria have been discussed extensively in the literature, there is a lack of a clear definition of the problem itself. Here the following formulation is introduced. Rigorously, considering a GP prior Y on f and the initial data set $(\mathrm{X}, \mathbf{y})$, each step in the sequential maximization of infill criteria is:

$$\mathbf{x}'_i = \arg\max_{\mathbf{x}\in\mathrm{S}} \mathscr{A}\left(\mathbf{x} \mid \mathbf{y}, \left\{ f(\mathbf{x}'_k) \mid \mathbf{x}'_k = \arg\max_{\mathbf{x}\in\mathrm{S}} \mathscr{A}\left(\mathbf{x} \mid \mathbf{y}, \{f(\mathbf{x}'_n) \mid \ldots\}_{n=1}^{k-1}\right)\right\}_{k=1}^{i-1}\right).$$

Note that the infill criterion is denoted as $\mathscr{A}(\mathbf{x} \mid \mathbf{y}, \{f(\mathbf{x}'_k)\}_k)$, to emphasize its dependence (from the underlying GP posterior) on evaluations $\{f(\mathbf{x}'_k)\}_k$ from all previous steps as well as the initial observations $\mathbf{y}$. In addition, the expression above is written in a recursive manner on purpose to show that it is not possible to decompose the maximization of infill criteria at a certain step from all previous function evaluations. Here, the goal of *infill criteria parallelization* is formulated as to obtain q points $\{\mathbf{x}_i\}_{i=1}^q \subset \mathrm{S}$ from an infill criterion $\mathscr{A}$ without any function evaluation, such that when comparing to points $\{\mathbf{x}'_i\}_{i=1}^q$ from the sequential q-step maximization of $\mathscr{A}$, the following condition holds:

$$\begin{aligned}\mathbb{E}\left\{\min\left(Y(\mathbf{x}_1), \ldots, Y(\mathbf{x}_q)\right) \mid \mathbf{y}\right\} \approx \min \Big\{ & \mathbb{E}\left\{Y(\mathbf{x}'_1) \mid \mathbf{y}\right\}, \\ & \mathbb{E}\left\{Y(\mathbf{x}'_2) \mid \mathbf{y}, f(\mathbf{x}'_1)\right\}, \\ & \ldots, \\ & \mathbb{E}\left\{Y(\mathbf{x}'_q) \mid \mathbf{y}, \{f(\mathbf{x}'_i)\}_{i=1}^{q-1}\right\} \Big\}.\end{aligned}$$

This condition regulates that the expectation of the best fitness from q locations is roughly the same as the minimal expected fitness value obtained from sequential q-step maximization of $\mathscr{A}$, or shortly the parallelization exhibits the same effect as

the sequential approach on average. Note that, although this condition gives a clear target when designing the parallelization method for infill criteria, it is difficult to validate parallelization methods on it. Thus, in the following, the approaches shall be discussed regardless of this condition.

4.5.1 Multi-point Infill Criteria

Multi-point Expected Improvement (q-EI) is proposed by Schonlau (1998) and computes the expectation of the smallest improvement among a set of correlated locations:

$$\mathrm{EI}^q = \mathbb{E}\left\{ f_{\min} - \min(Y(\mathbf{x}_1), Y(\mathbf{x}_2), \ldots, Y(\mathbf{x}_q)) \mid \mathbf{y} \right\}. \tag{4.15}$$

To give the exact formula of q-EI, it requires to integrate the smallest order statistic from q-correlated Gaussian random variables. The exact formula of 2-EI is derived in Ginsbourger et al. (2010). For an arbitrary number of points, the formula is given in Chevalier and Ginsbourger (2013). There are two heuristics, *Kriging Believer* and *Constant Liar*, proposed to approximate q-EI with fewer computations (Ginsbourger et al., 2010). In Kriging Believer, q points are obtained via the sequential q-step maximization of $\mathscr{A}$, where the real evaluation $f(\mathbf{x})$ is replaced by the Kriging prediction. In Constant Liar, a pre-defined fixed value is used for $f(\mathbf{x})$ and thus is called a "lie".

4.5.2 Multi-instance of Infill Criteria

Using the additional parameter β in LCB (Eq. (4.2)), Hutter et al. (2012) propose an alternative parallelization method, where q different β-values are sampled from the log-normal distribution $\mathit{Lognormal}(0, 1)$ and subsequently q different LCB criteria are instantiated using β samples:

$$\beta_i \sim \mathit{Lognormal}(0,1), \quad \mathbf{x}_i = \arg\min_{\mathbf{x}\in\mathrm{S}} \mathrm{LCB}(\mathbf{x}; \beta_i), \quad i = 1, 2, \ldots, q. \tag{4.16}$$

Compared to q-EI, this method brings no additional computational cost and it serves as reasonable between exploration and exploitation. On one hand, as the probability mass of the standard log-normal distribution concentrates around small values, most of the β samples will be relatively small and the corresponding LCB criteria are of low risk. On the other hand, the standard log-normal also possesses a long tail, meaning that it is possible to obtain a few large β samples with a small

probability. Note that such a trade-off is controlled by the mean and standard deviation of the log-normal distribution. At the time of writing, to the best of our knowledge, there is no theoretical work on investigating the impact of those parameters in the log-normal distribution, or a proof to show that the log-normal distribution is the optimal probability law for this purpose, in the first place. Regardless of this theoretical concern, this method can be easily applied to other *parameterized* infill criteria, e.g., MGFI and GEI, although its performance needs to be tested systematically.

4.5.3 Multi-objective Infill Criteria

The multi-objective treatment of the infill criteria (Section 4.2) also allows for the parallelization. As a natural extension to the bi-objective formulation in Eq. (4.10), the general vector-valued infill criteria is considered:

$$\boldsymbol{\mathscr{A}} : \mathrm{S} \to \mathbb{R}^m, \quad \mathbf{x} \mapsto \left(\mathscr{A}_1(\mathbf{x}), \mathscr{A}_2(\mathbf{x}), \ldots, \mathscr{A}_m(\mathbf{x})\right)^\top,$$

where $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_m : \mathrm{S} \to \mathbb{R}$ are real-valued infill criteria (subject to maximization) that are either selected from some well-defined infill criteria or generated from a parameterized one, e.g., using Eq. (4.16). To select q points based on this problem, for instance, we could adopt the so-called *decomposition-based multi-objective optimization* (Zhang and Li, 2007), in which q different linear scalarizations on infill criteria are generated uniformly and q points are obtained as the maximum of the linear scalarization, namely

$$\mathbf{x}_k = \underset{\mathbf{x} \in \mathrm{S}}{\arg\max} \sum_{i=1}^{m} w_i \mathscr{A}_i(\mathbf{x}), \quad w_1, w_2, \ldots, w_m \sim \mathcal{U}(0,1), \quad k = 1, \ldots, q.$$

Note that $\mathcal{U}(0,1)$ stands for the uniform distribution over $[0,1]$. Although this multi-objective proposal seems plausible, it remains untested as of the time of writing.

4.5.4 Niching-based Infill Criteria Maximization

Considering the landscape of EI, this approach is motivated by the observation that its landscape is usually highly multi-modal (Jones et al., 1998; Wang et al., 2018) (also see Fig. 4.2). The original EGO algorithm aims at finding the global

optimum of the EI landscape, which is another global optimization task and therefore is difficult to solve (the exhaustive branch-and-bound method has been proposed initially). Suppose $\mathbf{x}^*$ is the global maximum of EI and $\mathbf{x}'$ is a *local optimum*. After incorporating the fitness $f(\mathbf{x}^*)$, the Kriging formula are updated such that the Kriging MSE are largely reduced at $\mathbf{x}^*$ and a neighborhood around it. Consequently, the EI value drops locally in this neighborhood due the fact that EI decreases with deceasing Kriging MSE. As this is only a local change of the EI landscape, EI values around $\mathbf{x}'$ are normally not affected. Then, $\mathbf{x}'$ would possibly become new global maximum if EI is maximized again. Instead of using the sequential maximization of EI to "expose" local maxima, it is proposed to combine EGO with a so-called *niching evolution strategy* (Shir and Bäck, 2005b), where the niching method is intended to find multiple distinct local maxima simultaneously. In the evolutionary computation, niching refers to a collection of methods that aims at locating multiple distinct local optima, in order to improve the exploration on highly multi-modal function. In this manner, within one iteration, it is possible to locate the global maximum of EI as well as some local maxima, which would be explored using a few iterations, if the sequential maximization were performed. The resulting parallelization method is called **Niching-q-EI** (Wang et al., 2018). Interestingly, from the niching perspective, niching-q-EI can also be considered as a meta-model assisted niching algorithm where the niche formation is performed on the EI landscape instead of the real objective function. The niching-q-EI algorithm is summarized in Alg. 8.

Niching Evolution Strategies Herein the *niching evolution strategy* is briefly introduced. In general, Evolutionary Algorithms (EAs) have the tendency to converge quickly into a single solution in the search space. However, in many problem solving scenarios (e.g., global optimization), locating and maintaining multiple solutions/optima is required. Niching is developed to achieve this goal by forming sub-populations in order to maintain the population diversity of EAs. The most successful techniques are *fitness sharing* (Goldberg and Richardson, 1987) and *Crowding* (De Jong, 1975).

Although the niching technique is initially proposed mainly for Genetic Algorithms (GAs), it has also been introduced to classic $(1 \dot{+} \lambda)$ self-adaptive Evolution Strategies by Shir and Bäck (2005b). Later, it is further developed for the derandomized ES (Shir and Bäck, 2005a) including the well-known Covariance Matrix Adaptation

Algorithm 8 Niching-based Efficient Global Optimization

procedure NICHING-q-EI(q, S, f)
  Given: the number of points q for parallelization
  Sample the initial design $\mathrm{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathrm{S}$
  Evaluate $\mathbf{y} \leftarrow (f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n))^\top$
  Construct the Kriging/GPR model $\hat{f}$ on $\mathrm{X}, \mathbf{y}$.
  while the stop criteria are not fulfilled **do**
    $\{\mathbf{x}'_1, \ldots, \mathbf{x}'_q\} \leftarrow (1 \dot{+} \lambda)$-NICHING-ES($q, \mathrm{S}, \mathrm{EI}(\mathbf{x}; \hat{f})$) ▷ Alg. 9
    Parallel evaluation $y_1, \ldots, y_q \leftarrow f(\mathbf{x}'_1), \ldots, f(\mathbf{x}'_q)$
    $\mathrm{X} \leftarrow \mathrm{X} \cup \{\mathbf{x}'_1, \ldots, \mathbf{x}'_q\}$
    $\mathbf{y} \leftarrow (\mathbf{y}^\top, y_1, \ldots, y_q)^\top$
    Re-construct the Gaussian process model $\hat{f}$ on $\mathrm{X}, \mathbf{y}$
  end while
end procedure

Evolution Strategy (CMA-ES) (Hansen and Ostermeier, 2001) and finally evolved into a self-adaptive approach which allows for the niche radius and the niche shape adaptation (Shir et al., 2007). The $(1 \dot{+} \lambda)$-niching evolution strategy with fixed niche radius and spherical niche shape (Alg. 9) is chosen for our purpose and it works as follows: given q optima expected to investigate, the niching procedure initializes $q + p$ so-called "D-sets" (Shir et al., 2008) which are evolution strategy kernels containing all the adapted strategy parameters (step-size, covariance matrix) as well as decision parameters (current search point/solution, mutation vectors). Each D-set defines the current search point and all the internal information regarding an evolution strategy in a given time during the evolution.

The q D-sets are meant for identifying q possible local optima/peaks while p D-sets are for a "non-peak" domain, which are randomly regenerated every κ generations. The purpose of p "non-peak" D-sets is to explore the search space so that new niches would emerge and the probability of finding undiscovered optima is increased. The niching procedure then proceeds to generate λ offspring for each D-set. The population of $\lambda(q + p)$ offspring is evaluated according to the fitness function. Using their corresponding fitness values, the selection of p search points is conducted based on the *dynamic peak identification* (DPI) algorithm (Miller and Shaw, 1996), using a prescribed niche radius ρ.

The functionality of DPI is to select a subset from the population in which each

search point has a good fitness value and is not within the radius of the remaining points. The selected p search points are considered as parental points for the next generation and their D-sets are inherited from their parents. Finally, the D-sets are updated based on the selected points. The process above is repeated until the termination criteria are satisfied.

Algorithm 9 Niching Evolution Strategy

procedure $(1 \overset{+}{,} \lambda)$-NICHING-ES$(q, \mathrm{S}, f)$▷ q: number of niches, S: search space, f: fitness function
  Initialize D-set $\{D_i\}_{i=1}^{q+p}$ in search space S
  Set generation counter $c \leftarrow 0$
  while the stop criteria are not fulfilled **do**
    for $i = 1 \rightarrow (q + p)$ **do**
      Generate λ mutations according to D_i
    end for
    Evaluate the population using fitness function f.
    Obtain the dynamic peaks set DPS by performing Dynamic Peak Identification.
    for p in DPS **do**
      Set p as a new search point
      Inherit the D-set from the parent of p and update the D-set accordingly.
    end for
    if $N =$ size of $DPS < q$ **then**
      Generate $N - q$ new search points and reset corresponding D-sets.
    end if
    $c \leftarrow c + 1$
    if $c \mod \kappa = 0$ **then**
      Randomly re-generate $(q + 1)^{th} \ldots (q + p)^{th}$ D-sets.
    end if
  end while
end procedure

For implementation details, we choose the niching-DR2 (Shir and Bäck, 2005b) among many other niching evolution strategies which could be introduced into EGO. It is the niching version of a so-called "second derandomized evolution strategy" (DR2) (Ostermeier et al., 1994). We choose niching-DR2 because it is both simple to implement and converges fast. The parameters of niching-DR2,

which are listed in Alg. 9, are set as following: the function evaluation budget is set the be $10^3(q+p)$. The parameter κ controls the frequency of sub-population resampling and is set to 10. The niche radius ρ required by DPI procedure is computed as (Shir and Bäck, 2005b):

$$\rho = \frac{r}{\sqrt[n]{q}}, \quad r = \frac{1}{2}\sqrt{\sum_{k=1}^{n}(x_{k,max} - x_{k,min})^2},$$

where n is the dimensionality and $x_{k,max}, x_{k,min}$ are the upper and lower bound of each coordinate in the search space. The termination criterion for the niching ES is currently given by the function evaluation budget.

Example As with the Constant Liar (CL) strategy (Ginsbourger et al., 2010), the niching approach is also expected to have a repulsive behavior between the points, due to the niching formation. The CL repulsive behavior is controlled with increasing lie value L. The setting $L = \max\{\mathbf{y}\}$ and $L = \text{mean}\{\mathbf{y}\}$ leads to a space filling behavior (Ginsbourger et al., 2010). The behavior of the niching approach and the CL strategy are further compared and visualized on the Himmelblau's function:

$$y_H(x_1, x_2) = (x_1^2 + x_2 + 11)^2 + (x_1 + x_2^2 - 7)^2$$

Himmelblau's function has four global optima located at $(3, 2)$, $(-2.81, 3.13)$, $(-3.78, -3.23)$ and $(3.58, -1.85)$ with global minimal value 0. In order to show that the niching approach can maintain multiple distinct points, we choose four points to be generated in each iteration, which are expected to identify four global optima on Himmelblau's function.

In Fig. 4.8, we compare the CL strategy with $L = \min\{\mathbf{y}\}$ (CL min) to niching-q-EI in four consecutive iterations (from top to bottom). Both of the two approaches locate the four basins of attraction. They also explore the search space while keeping track of all the points found, showing a trade-off between exploitation and exploration. The difference is that the niching approach is not likely to sample two points in one high performance region while the CL min strategy would result in two (or even more) points explore the same region (see step 3 and 4 in the figure). In addition, we also estimate the q-EI values of points found from a 4-point EGO with niching on Himmelblau's function. The estimation is conducted by Monte Carlo simulations of the q-EI formula (Ginsbourger et al., 2010).

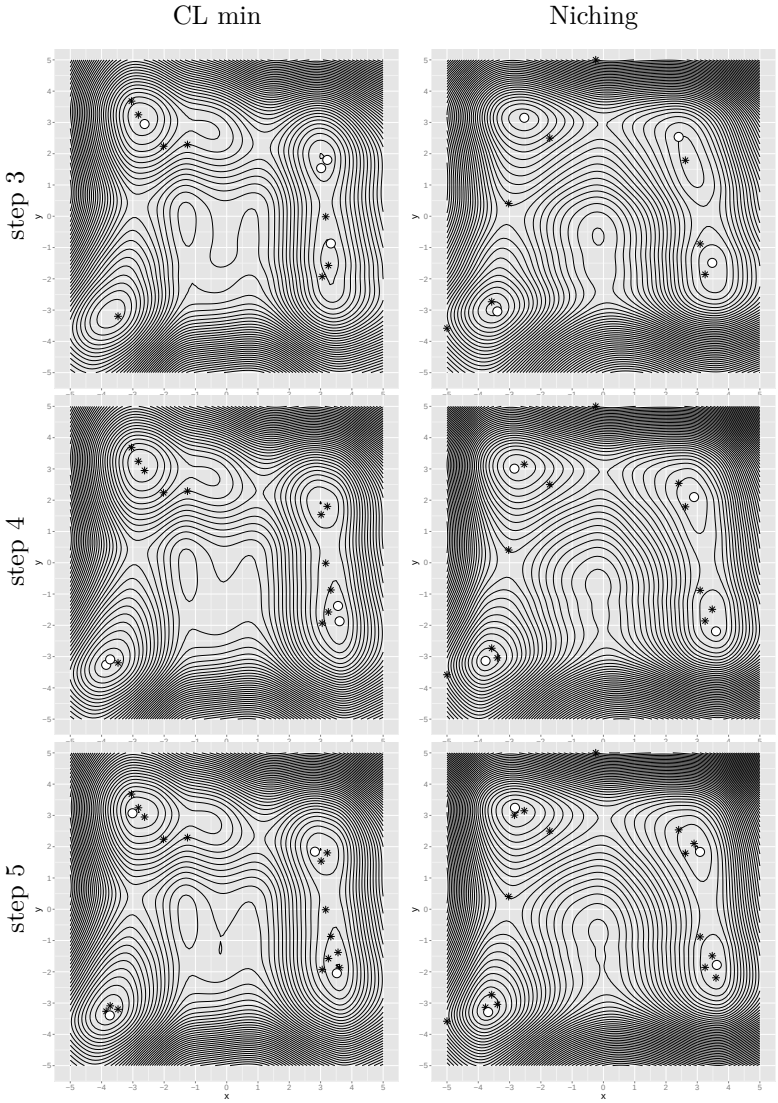


Figure 4.8: On *Himmelblau's function*: 4-point EGO in four iterations. The white circles indicates the current sites found in each iteration while black stars show the sites sampled in history. **Left**: Constant Liar Strategy using min{**y**}. **Right**: niching-q-EI.

4.6 Experimental Comparison

In this section, we test the niching-q-EI and three Constant Liar variants: CL min, CL max and CL mix on a collection of test functions. The CL mix strategy (Chevalier and Ginsbourger, 2013) is a mixture of the CL min and CL max in which two batches of points are generated from the CL min/max and the batch of better q-EI value is provided to the Kriging model. For all the tests we use the `DiceKriging` and `DiceOptim` packages (Roustant et al., 2012). The experiment is presented in three parts: First we list the test functions selected. Then, the global convergence is compared among all the tested algorithms and finally the q-EI values of the point found are compared. All the algorithms are tested in 10 iterations and all the results are averaged over 100 runs. The reason to choose a small number of iterations as test run length is simple: on one hand, the classic EGO algorithm is capable of locating all the optima on several test functions (Jones et al., 1998) using 10 to 20 iterations. Thus, longer runs are not necessary. On the other hand, due to our observations, most of the space explorations and the Kriging model updates happen in the first 10 to 15 iterations on the 2-D functions, in which the EGO algorithm makes large progress.

Test Functions We select 6 artificial multi-modal continuous functions:

- The transformed *Hartman6* function is defined on $[0,1]^6$ and is a unimodal function. The original function is transformed by $-\log(-\,\text{Hartman6}(\mathbf{x}))$. This is the test-case where niching-q-EI is expected to perform badly. We set $q = 3$ on this function.
- M is a hyper-grid multi-global function. Its global optima are uniformly distributed and have optimal value -1. The function expression is listed below. We test the algorithms in 2-D where 10 minima are located in $[0,1]^2$.

$$M(\mathbf{x}) = -\frac{1}{d}\sum_{i=1}^{d}\sin^{\alpha}(5\pi x_i).$$

  Note that d is the dimensionality and we choose $\alpha = 6$.

- The *Branin function* is a multi-global function and a classical test-case in global optimization (Jones et al., 1998; Schonlau, 1998). It is defined in 2-D with three global optima. The global minimal value is roughly 0.4.

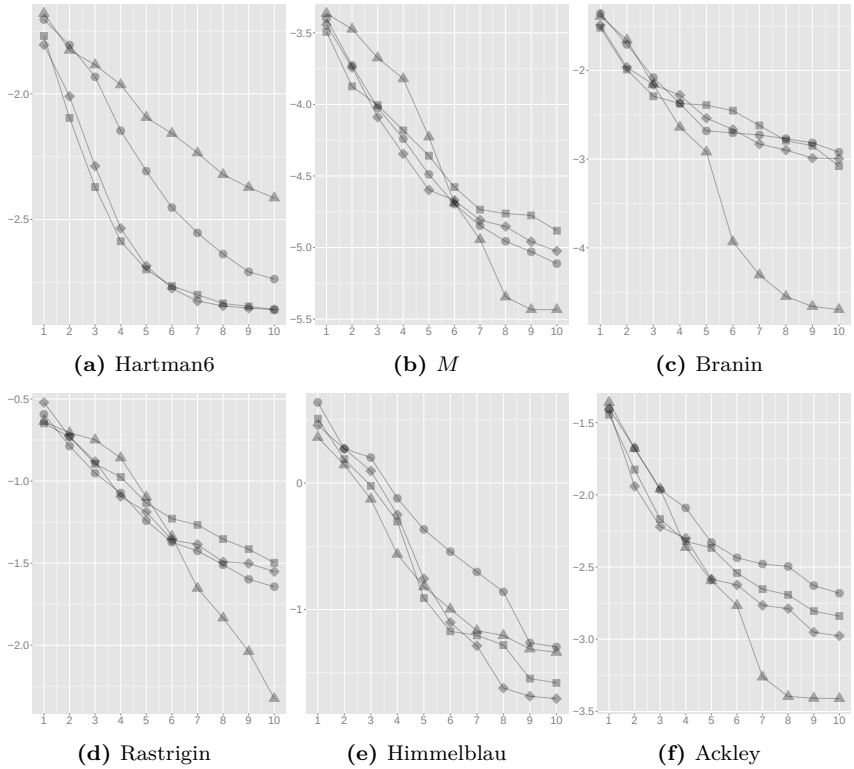
- The *Rastrigin function* (Torn and Zilinskas, 1989) is a multi-modal function, which has only one global optimum, surrounded by a number of local minima. The test is performed on 2-D. It has 6 optima in the space $[0, 1.5]^2$.
- *Himmelblau's function* is a multi-global function which is introduced previously in this thesis. The search space is $[-5, 5]^2$.
- The *Ackley function* is a multi-modal function and has only one global optimum. The global optimum has a much lower value than the local optima.

We choose the number of points generated in each iteration equal to the number of minima with two exceptions: on Hartman6, which is a unimodal function, we choose $q = 3$ and on the Ackley function where the local optima increases exponentially with the increasing distance to the global optimum, we choose $q = 9$. For Hartman6 function, it is intended to show that the niching-q-EI could perform quite badly with $q > 1$ setting. We thus choose a moderate value $q = 3$. For the Ackley function with range $[-5, 5]^2$, there are 8 sub-optimal locations whose function values are the same, inferior to the global optimal but superior to the remaining local optima. We would like to locate such sub-optima as well as the global one and thus choose $q = 9$.

Convergence results The *relative mean squared error* (Chevalier and Ginsbourger, 2013) is used to measure the convergence rate to the global optimum. It is defined as:

$$\mathrm{rMSE}_i = \frac{1}{p} \sum_{k=1}^{p} \left(\frac{f_i^k - f^*}{f^*} \right)^2 .$$

Here rMSE_i denotes the relative mean squared error at iteration i (rMSE should not be confuse the Kriging RMSE in Section 3.1.1). Furthermore, p is the number of runs performed while f_i^k is the minimum value observed at iteration i in run number k. Note that we translate optimal values of some test functions to prevent 0 when calculating the rMSE. The relative mean squared error on each test function is shown in Fig. 4.9. Note that the rMSE is scaled by $\log 10$. On the unimodal function Hartman6, the results of 3-points EGO show that the niching-q-EI performs much worse than any variants of CL strategy. This is the expected behavior because the niches formed on the EI landscape of Hartman6 do not map to any local optima and the niching method is performing space-filling using all three niches.



(a) Hartman6 **(b)** M **(c)** Branin

(d) Rastrigin **(e)** Himmelblau **(f)** Ackley

Figure 4.9: The relative mean squared error to the global optimal (y axis in $\log 10$) against iterations (x axis). Legend: ■: CL max, ●: CL min, ◆: CL mix, ▲: niching-q-EI.

On the multi-modal Rastrigin function, niching-q-EI actually outperforms all the CL variants. On the Branin function, niching-q-EI performs equally to the CL max in the first three iterations and makes a large acceleration from the fourth iteration. On the M function, the niching approach works much worse in the first 6 iterations and accelerates again in the later iterations. On the Rastrigin function, the same behavior is observed. We think that the reason is that initially the Kriging prediction response surface differs from the real landscapes drastically so that the niches formed on the EI landscape do not map to any high performance region. After updating the Kriging model, local optima on the objective function would possibly create local optima in the EI landscape. On Himmelblau's function, niching-q-EI is the worst method initially and finally catches up with the CL mix from iteration 8. On the Ackley function, niching-q-EI performs roughly the

same as the CL min strategy and outperforms both the CL min and CL mix after iteration 5. In general, convergence plots suggest that initially niching-q-EI performs worse than or equally to the CL variants and accelerates the convergence after updating the Kriging model for some iterations. Furthermore, such behavior may even suggest a possible mixture approach where the CL strategy is applied in the beginning and then the algorithm is switched to the niching method to gain from the acceleration.

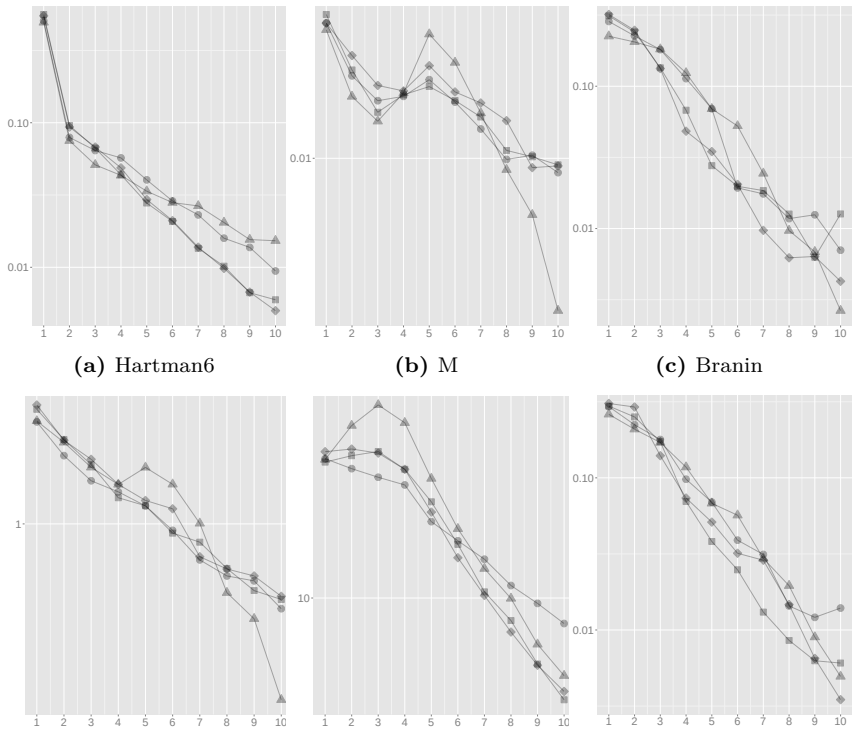


Figure 4.10: Average q-EI value of points from tested algorithms. Legend: ■: CL max, ●: CL min, ◆: CL mix, ▲: niching-q-EI. The x axis represents the iteration while the y axis is the averaged q-EI measured.

q-EI of search points The average q-EI values of the points generated in each iteration are computed by Monte Carlo simulations (Ginsbourger et al., 2010), which are shown in Fig. 4.10. The plotted values are scaled by $\log 10$. All the average q-EI values decrease with respect to increasing iterations. It is not clear

which method is the winner in general. Focusing on the first 4 iterations, q-EI values of the search points found by the niching approach are roughly smaller than q-values from the CL variants on the Hartman6, $\mathcal{M}$ and Ackley function. On the Rastrigin and Himmelblau's function, the proposed algorithm gives higher q-EI values in the middle (iteration $4, 5, 6$) of the test. On the Branin function, the differences are not significant. In general, niching-q-EI shows a much faster q-EI value reduction when the iteration goes upper than 8.

4.7 Summary

Infill criteria control the exploration-exploitation trade-off in the Efficient Global Optimization algorithm. In this chapter, we focus on the improvement-based infill criteria, that is a class of functions that calculates the potential improvement over the current best objective value. It is shown that the exploration-exploitation trade-off can be explicitly controlled by considering the risk and return of the infill criterion, resulting in a multi-objective infill criterion. Alternatively, such an exploration-exploitation control can also be realized by the novel infill-criterion, called Moment-Generating Function of Improvement (MGFI). MGFI introduces a real parameter, called "temperature" that tunes the exploration-exploitation trade-off smoothly. Furthermore, the cooling strategies, that are originally introduced in the Simulated Annealing, are applied to the temperature parameter of MGFI. The resulting infill criterion exhibits high exploration behaviors in the beginning (high temperature) and evolves towards exploitative behaviors as the temperature cools down. Finally, the parallelization problem of infill-criteria is investigated. The challenge here is how to obtain several well-performing candidate solutions based on the infill criterion. Several new methods are proposed for this purpose, including multi-objective infill criteria and niching-assisted infill criteria maximization.