Multi-objective Bayesian global optimization for continuous problems and applications

Yang, K.

Chapter 2

Continuous Multi-objective Optimization

Evolutionary algorithms (EAs) and Bayesian global optimization (BGO) are two major branches in the field of continuous optimization algorithms. Both of them share a similar structure: (1) initialization, (2) evaluation of current solutions, (3) adjustment of the current solutions for the aim of seeking an improvement in the next loop, and (4) repetition of the evaluation and adjustment loop. The difference lies in the adjustment mechanism. For EAs, it is accomplished by evolutionary operators, such as recombination and mutation. For BGO, it is achieved by learning from the past evaluations and updating a surrogate model.

To start the journey, a good preparation is always required. This chapter serves to lay out the terminologies and the groundwork of the studies in this dissertation. The structure of this chapter is structured as follows: Section 2.1 provides the definition of multi-objective optimization; Section 2.2 defines some fundamental terminologies in the field of multi-objective optimization; Section 2.3 provides the definitions of some common infill criteria; Section 2.4 describes two state-of-the-art evolutionary multi-objective optimization algorithms, namely SMS-EMOA and NSGA-II, which are utilized to solve a power distribution network reconfiguration problem in this chapter; Section 2.5 introduces multi-objective Bayesian global optimization, together with Kriging and a simple example.

2.1 Multi-objective Optimization

Multi-objective optimization is a generalization of single-objective optimization. It can be generalized by means of selecting the best combination of parameters in order to optimize the multiple performances simultaneously. The basic idea of MOO is that it optimizes the performances depending on these parameters, possibly subject to some restrictions on the allowed parameter ranges. The performances of the problem which needs to be optimized are called *objective functions or fitness functions* and they depend on the combination of parameters; the parameters are called *decision variables or the decision vector*; the range of the decision vectors is known as *search space*; the restrictions on allowed parameters are called *constraints*; an allowed decision vector is called *a feasible decision vector*. A multi-objective optimization (MOO) problem is an optimization problem that involves multiple objective functions and it can be formulated as:

$$\begin{aligned} &\mathbf{max}\big(y_1(\mathbf{x}), y_2(\mathbf{x}), \cdots, y_d(\mathbf{x})\big) \\ &\text{subject to} \qquad \mathbf{x} \in \mathcal{X} \subseteq \mathbb{S} \end{aligned} \tag{1-1}$$

where the integer d is the number of objective functions, $\mathcal{X}$ is the feasible set of decision vectors, $y_i \;\; i = 1, \cdots, d$ are the objective functions, and $\mathbb{S}$ is the search space of decision vectors $\mathbf{x}$ in m dimensional space.

Multi-objective optimization consists of two main branches of algorithmic solution approaches. The first approach is called *weighted sum method*. It converts a multi-objective optimization problem into a single-objective optimization problem by multiplying each objective function with a corresponding weighting factor and summing them up. The *weighted sum method* is simple and easy to be implemented. However, its weakness is obvious. On the one hand, it is very difficult to depict the thoroughly complete Pareto front set [12]. On the other hand, the solution obtained by using the weighted sum method does not necessarily reflect the preferences, when we want to represent the preferences of a decision maker by weights [13].

The second approach treats each objective function separately and utilizes the concept of a Pareto front as the fundamental concept to optimize each objective function by using different mechanisms (non-dominated ranking, infill criteria based on Pareto front, etc.). This research focuses on the second approach.

2.2 Terminologies

This section mainly introduces the concepts and terminologies of a **Pareto front** based on the objective space[1]. A **Pareto front** set is based on the dominance concept. The dominance is defined as follows:

Definition 2.1 (Dominance [14]) *Given two decision vectors* $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in \mathcal{X}$ *and their corresponding objective values* $\mathbf{y}^{(1)} = \mathbf{y}(\mathbf{x}^{(1)})$, $\mathbf{y}^{(2)} = \mathbf{y}(\mathbf{x}^{(2)})$, *it is said that* $\mathbf{y}^{(1)}$ *dominates* $\mathbf{y}^{(2)}$, *being represented by* $\mathbf{y}^{(1)} \prec \mathbf{y}^{(2)}$, *iff* $\forall i \in \{1, 2, \cdots, d\} : y_i(\mathbf{x}^{(1)}) \leq y_i(\mathbf{x}^{(2)})$ *and* $\exists j \in \{1, 2, \cdots, d\} : y_j(\mathbf{x}^{(1)}) < y_j(\mathbf{x}^{(2)})$.

Dominance is a fundamental concept in multi-objective optimization, and it provides an explicit relation between two solutions. In some cases, it can be used to decide which solution is better than the other. However, more interests from the perspective of research are put on the non-dominated solutions in MOO, because a point in a non-dominated space means a potential improvement of the objective function values.

Definition 2.2 (Non-dominance [2]) *Given a decision vector set* $\mathbf{X} \subset \mathbb{S}$, *and the image of the vector set is* $\mathbf{Y} = \{\mathbf{y}(\mathbf{x}) | \mathbf{x} \in \mathbf{X}\}$, *the non-dominated subset of* $\mathbf{Y}$ *is defined as:*

$$nd(\mathbf{Y}) := \{\mathbf{y} \in \mathbf{Y} | \nexists \mathbf{z} \in \mathbf{Y} : \mathbf{z} \prec \mathbf{y}\} \tag{2-2}$$

A vector $\mathbf{y} \in nd(\mathbf{Y})$ in the objective space is called a non-dominated point. A non-dominated set means that there is no solution better or equally good in all components of the objective space. However, there could be solutions that are, at least, better in some component(s) with sacrificing the performance in the other component(s). The goal of MOO is trying to find all non-dominated solutions in a whole feasible search space, which is called **Pareto front**, and defined as:

Definition 2.3 (Pareto front [14]) *For a feasible decision set* $\mathcal{X} \subset \mathbb{S}$, *the image of it is* $\mathcal{Y} = \{\mathbf{y}(\mathbf{x}) | \mathbf{x} \in \mathcal{X}\}$, *the Pareto front set* $\mathcal{P}^*$ *is defined as:*

$$\begin{aligned} \mathcal{P}^* : &= \{\mathbf{y} \in \mathcal{Y} | \nexists \mathbf{z} \in \mathcal{Y} : \mathbf{z} \prec \mathbf{y}\} \\ &= nd(\mathcal{Y}) \end{aligned} \tag{2-3}$$

[1] In some papers, dominance is represented in a search space, instead of an objective space.

In a Pareto front, each solution is a non-dominated point in $\mathbf{Y}$. Then, the problem of MOO is converted to how to find a Pareto front set $\mathcal{P}^*$. However, a Pareto front $\mathcal{P}^*$ is difficult to be obtained, especially for a high-dimensional and continuous black-box problem. This is because usually only a finite number of non-dominated points can be obtained. Commonly, a Pareto front approximation set, which contains only a subset of a Pareto front set, is used to be optimized in MOO. Then, the final question will become how to define a Pareto front approximation set $\mathcal{P}$ which can approximate the Pareto front set $\mathcal{P}^*$ best. A Pareto front approximation is defined as:

Definition 2.4 (Pareto front approximation) *Given a Pareto front set $\mathcal{P}^*$, a Pareto front approximation set $\mathcal{P}$ is any set of mutually non-dominated points and it is defined as:*

$$\mathcal{P} := \{\mathbf{y} \in \mathcal{Y}^{'} \subseteq \mathcal{Y} | \nexists \mathbf{z} \in \mathcal{Y}^{'} : \mathbf{z} \prec \mathbf{y}\} \tag{2-4}$$

Example 2.1 *Figure 2.1 illustrates the concept of Pareto* Dominance*,* Pareto front *and* Pareto front approximation *in 2-D case. Suppose the image of the decision space $\mathcal{X} \subset \mathbb{S}$ is $\mathcal{Y}$, then $\mathcal{Y}$ can be expressed by dots in Figure 2.1. The Pareto front $\mathcal{P}^*$ of $\mathcal{Y}$ is the non-dominated set of $\mathcal{Y}$ and it is represented by solid black curves. A Pareto front approximation $\mathcal{P}$, represented by the solid gray dots surrounded by dashed curves, is dominated by $\mathcal{P}^*$. The other gray dots are the dominated points, which are dominated by $\mathcal{P}$.*

2.3 Infill Criteria[2]

Given two Pareto front approximation sets, how to evaluate and compare the quality between the two Pareto front approximation sets? This section introduces some basic infill criteria, which will be used in later chapters, in MOO.

Hypervolume Indicator: The *Hypervolume Indicator*, proposed by Zitzler and Thiele [15], measures the size of the dominated subspace bounded from below[3]

[2]This section only considers maximization problems.

[3]The original definition was for minimization problems and the reference point bounds the set from above.

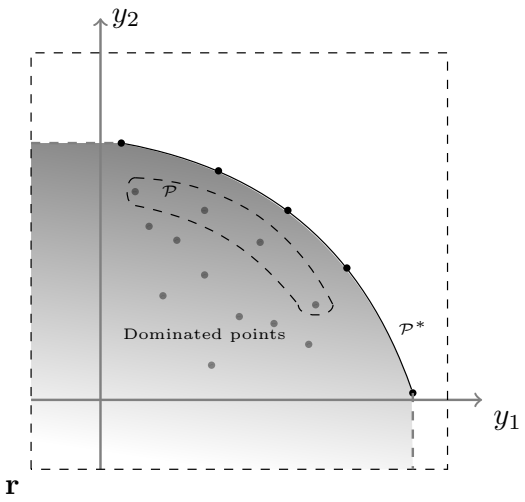

Figure 2.1: Example of 2-D Pareto front and Pareto front approximation.

by a reference point $\mathbf{r}$. The hypervolume indicates the performance of a Pareto-front approximation set $\mathcal{P} \subset (\mathbb{R}^d)^n$, where n stands for the number of the points in $\mathcal{P}$, and the maximization of HV can lead to a Pareto-front approximation set that is close to the true Pareto front. In 2-D and 3-D cases, the hypervolume indicator can be computed in time $\Theta(n \log n)$ [16]. In more than 3 dimensions, the algorithm proposed by Chan [17] achieves $O(n^{\frac{d}{3}} \operatorname{polylog} n)$ time complexity. The hypervolume indicator is defined as:

Definition 2.5 (Hypervolume Indicator) *Given a finite Pareto front approximation set, say* $\mathcal{P} = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)}\} \subset \mathbb{R}^d$*, the* Hypervolume Indicator *(HV) of* $\mathcal{P}$ *is defined as the d-dimensional Lebesgue measure of the subspace dominated by* $\mathcal{P}$ *and bounded below by a reference point* $\mathbf{r}$*:*

$$HV(\mathcal{P}) = \lambda_d(\cup_{\mathbf{y} \in \mathcal{P}}[\mathbf{r}, \mathbf{y}]) \tag{3-5}$$

with λ_d *being the Lebesgue measure on* $\mathbb{R}^d$*.*

The reference point needs to be provided by the user, and it should, if possible, be chosen in such a way that it is dominated by all elements of the Pareto-front approximation sets $\mathcal{P}$ that might occur during the optimization process.

Hypervolume Improvement *Hypervolume Improvement* (HVI) is also called *Improvement of Hypervolume* in [18]. The basic idea of HVI is the HV change of a Pareto front approximation set $\mathcal{P}$ before and after adding an evaluated point $\mathbf{y}$ in it. The definition of *Hypervolume Improvement* is:

Definition 2.6 (Hypervolume Improvement) *Given a finite collection of vectors $\mathcal{P} \subset \mathbb{R}^d$, the* Hypervolume Improvement *(HVI) of a vector $\mathbf{y} \in \mathbb{R}^d$ is defined as:*

$$HVI(\mathbf{y}, \mathcal{P}) = HV(\mathcal{P} \cup \{\mathbf{y}\}) - HV(\mathcal{P}) \tag{3-6}$$

In case we want to emphasize the reference point $\mathbf{r}$, the notation $HVI(\mathbf{y}, \mathcal{P}, \mathbf{r})$ will be used to denote the Hypervolume Improvement. *Note that $HVI(\mathbf{y}, \mathcal{P}) = 0$, in case $\mathbf{y} \in \mathcal{P}$.*

Hypervolume Contribution Another HV based criterion is *Hypervolume Contribution* (HVC). It is applied as a selection criterion in SMS-EMOA [19]. The most efficient algorithm to calculate HVC (one time) currently holds a time complexity $\Theta(n \log n)$ for $d = 2, 3$ as proposed by Emmerich and Fonseca in [20]. The basic idea behind HVI and HVC is the same, that is, to calculate the difference of the hypervolume between two Pareto front approximation sets. The *Hypervolume Contribution* is defined as:

Definition 2.7 (Hypervolume Contribution) *Given a finite collection of vectors $\mathcal{P} \subset \mathbb{R}^d$, the* Hypervolume Contribution *(HVC) of a vector $\mathbf{y} \in \mathbb{R}^d$ is defined as:*

$$HVC(\mathbf{y}, \mathcal{P}) = HV(\mathcal{P}) - HV(\mathcal{P} \setminus \{\mathbf{y}\}) \tag{3-7}$$

In case we want to emphasize the reference point $\mathbf{r}$, the notation $HVC(\mathbf{y}, \mathcal{P}, \mathbf{r})$ will be used to denote the Hypervolume Contribution.

Example 2.2 *Figure 2.2 illustrates the concept of the* Hypervolume Improvement *and the* Hypervolume Contribution. *For the 2-D case, suppose a Pareto front approximation set is $\mathcal{P}$, which is composed by $\mathbf{y}^{(1)} = (1, 2.5)^T$, $\mathbf{y}^{(2)} = (2, 1.5)^T$ and $\mathbf{y}^{(3)} = (3, 1)^T$. When a new point $\mathbf{y}^{(+)} = (2.8, 2.3)^T$ is added, the* Hypervolume Improvement *$HVI(\mathcal{P}, \mathbf{y}^{(+)})$ is the yellow area. The* Hypervolume Contribution *of $\mathbf{y}^{(1)}$ is the green area. The right figures illustrate the* Hypervolume Improvement *and the* Hypervolume Contribution *for the 3-D case. A Pareto-front approximation is $\mathcal{P} = \big(\ \mathbf{y}^{(1)} = (4, 4, 1)^T,\ \mathbf{y}^{(2)} = (1, 2, 4)^T,\ \mathbf{y}^{(3)} = (2, 1, 3)^T\big)$. The* Hypervolume Improvement *of $\mathbf{y}^{(+)} = (3, 3, 2)^T$ relative to $\mathcal{P}$ is given by the joint volume covered by the yellow slices. The* Hypervolume Contribution *of $\mathbf{y}^{(1)}$ is given by the joint volume covered by the green slices.*

Other infill criteria, for instance, *Expected Hypervolume Improvement*, *Probability of Improvement* and *Truncated Hypervolume Improvement* will be introduced in later chapters.

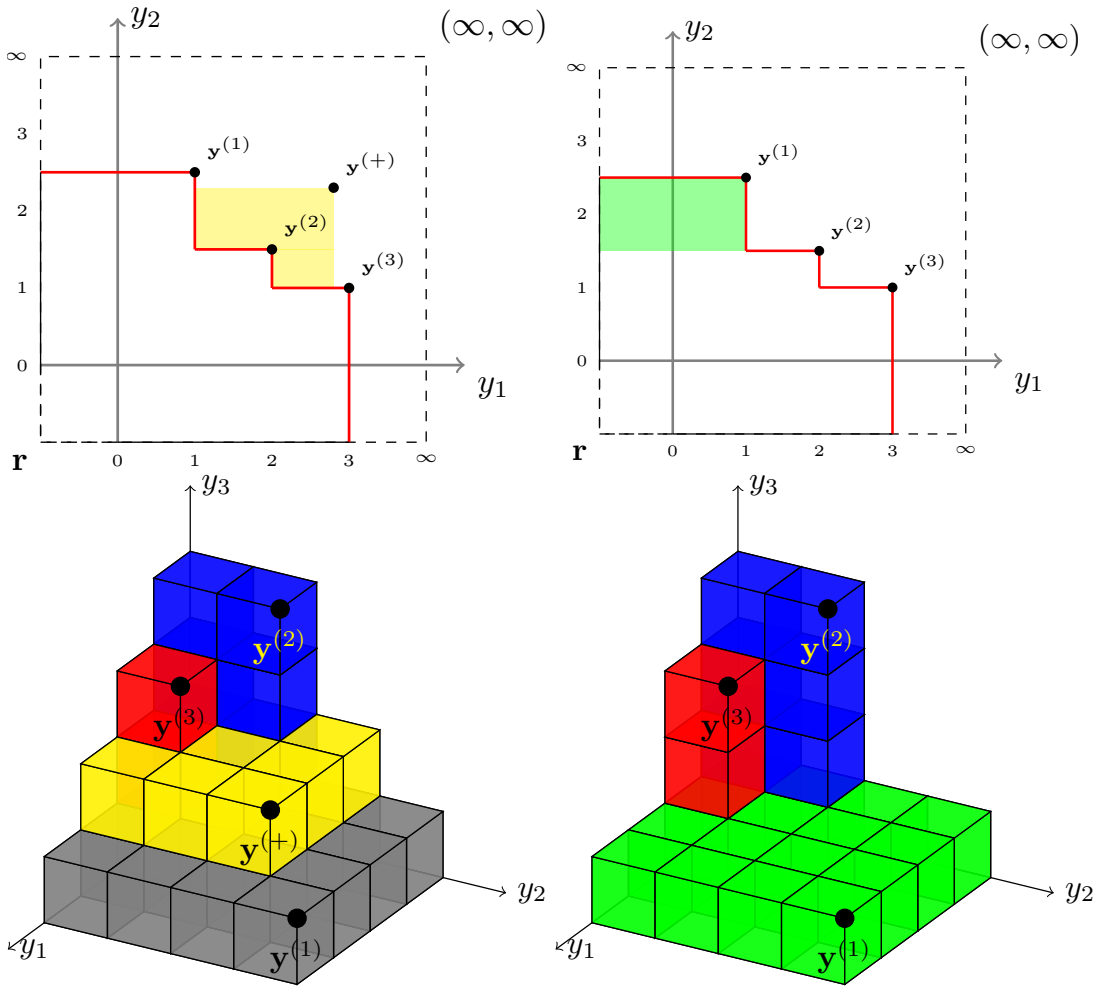

Figure 2.2: The left column illustrates *Hypervolume Improvement* for 2-D and 3-D cases. The right column illustrates *Hypervolume Contribution* for 2-D and 3-D cases. The yellow areas stand for the Hypervolume improvement of $y^{(+)}$, and green areas represent the Hypervolome contribution of $y^{(1)}$.

2.4 Evolutionary Multi-objective Optimization Algorithms

Evolutionary algorithms are population based metaheuristics optimization algorithms and they are inspired by biological paradigm of natural selection, recombination and mutation [21, 22]. The main advantage of evolutionary algorithms, when applied to solve multi-objective optimization problems, is the fact that they typically generate sets of solutions, allowing computation of an approximation of the entire Pareto front.

2.4.1 NSGA-II

NSGA-II is an improved version of NSGA (Nondominated Sorting Genetic Algorithm) [23], and it is a classical multi-objective algorithm proposed by Deb et al. [24, 25]. Being the most commonly applied evolutionary algorithm in the field of multi-objective optimization, NSGA-II serves as a reference algorithm in this dissertation. NSGA-II implements elitist mechanism,[2] where all the non-dominated solutions discovered are preserved from the beginning of the initial population. The selection mechanism of NSGA-II considers two factors: non-dominated rank of an individual in the population and its crowding distance (the average distance between two points on either side of this point along each of the objectives) for two objectives optimization. The priority between these two factors is the non-dominated rank. If two solutions are in the same non-dominated rank, the one that resides in the less crowded region is chosen. The basic structure of NSGA-II is shown in Algorithm 1.

2.4.2 SMS-EMOA

A popular algorithm that uses the hypervolume indicator as a selection criterion is SMS-EMOA [27]. In its two-dimensional instantiation, it can be viewed as a steady state variant of the NSGA-II algorithm [25] that replaces the crowding distance by hypervolume contributions, and thereby generates a sequence of approximation sets that grow according to the hypervolume indicator. The basic structure of SMS-EMOA is illustrated in Algorithm 2.

[2]This was claimed by the authors, but it is wrong and explained in Emmerich et al. [26].

Algorithm 1: NSGA-II

Input: Crossover rate p_c, mutation rate p_m, population size μ, offspring size λ, objective functions $\mathbf{y}$

Output: Pareto front approximation $\mathcal{P}$

1: Evaluate an initial set of μ points and store these in $\mathcal{P} = ((\mathbf{x}^{(1)}, \mathbf{y}^{(1)} = \mathbf{y}(\mathbf{x}^{(1)})), \ldots, (\mathbf{x}^{(\mu)}, \mathbf{y}^{(\mu)} = \mathbf{y}(\mathbf{x}^{(\mu)})))$;
2: **while** *termination criterion not satisfied* **do**
3: Select parents $\mathbf{X}^{'}$ from $\mathcal{P}$: $\mathbf{X}^{'} = Selection(\mathcal{P})$;
4: Crossover for $\mathbf{X}^{'}$: $\mathbf{X}^{'} = Crossover(\mathbf{X}^{'}, p_c)$;
5: Mutation for $\mathbf{X}^{'}$: $\mathbf{X}^{'} = Mutation(\mathbf{X}^{'}, p_m)$;
6: Evaluate the offspring $\mathbf{X}^{'}$ and store these in $\mathcal{P}^{'} = ((\mathbf{X}^{'(1)}, \mathbf{y}^{(1)} = \mathbf{y}(\mathbf{X}^{'(1)})), \ldots (\mathbf{X}^{'(\lambda)}, \mathbf{y}^{(\lambda)} = \mathbf{y}(\mathbf{X}^{'(\lambda)})))$;
7: Fast non-dominated sorting for $\mathcal{P} \cup \mathcal{P}^{'}$: $F = FastNonDominatedSorting(\mathcal{P} \cup \mathcal{P}^{'})$;
8: Update $\mathcal{P}$ by selecting best μ individuals from $\mathcal{P} \cup \mathcal{P}^{'}$;
9: Return $\mathcal{P}$

SMS-EMOA shows a slightly better performance on standard benchmarks than other commonly applied multi-objective optimization algorithms such as NSGA-II [27]. Therefore, SMS-EMOA was chosen as another reference algorithm in this research. We will compare its performance with that of multi-objective Bayesian global optimization defined in Chapters 3, 4, 5 and 8.

2.4.3 Example[1]

The utilization of evolutionary multi-objective algorithm is illustrated through a power distribution network reconfiguration problem (DNRP), which is served as a preliminary study in this dissertation. The network reconfiguration problem in a power distribution system aims at finding the best configuration of a radial network by changing the status of the switches in a power network system. There are two types of switches: normally closed switches and normally open switches. See Figure 2.3, for an example of a power distribution network configuration, the

[1]This example is a discrete optimization problem, and all the other parts of this dissertation consider only continuous optimization problems.

Algorithm 2: SMS-EMOA

Input: Objective functions $\mathbf{y}$, population size μ

Output: Pareto front approximation $\mathcal{P}$

1: Evaluate an initial set of μ points and store these in $\mathcal{P} = ((\mathbf{x}^{(1)}, \mathbf{y}^{(1)} = \mathbf{y}(\mathbf{x}^{(1)})), \ldots, (\mathbf{x}^{(\mu)}, \mathbf{y}^{(\mu)} = \mathbf{y}(\mathbf{x}^{(\mu)})))$;

2: **while** *termination criterion not satisfied* **do**

3: Generate a new solution $\mathbf{x}_{new}$ using PMX recombination and/or polynomial mutation operator (cf. [25]) on (some) solutions of $\mathcal{P}$;

4: Add point $\mathbf{x}_{new}$ to $\mathcal{P}$;

5: Compute dominance rank of each solution in $\mathcal{P}$ by means of non-dominated sorting ;

6: Determine R_{max} as the worst ranked layer of $\mathcal{P}$ and remove it from the solution with smallest hypervolume contribution ;

7: Return $\mathcal{P}$

119 bus system [28], where the solid black lines represent normally closed switches, while dashed red lines represent normally open switches. Network reconfiguration is the process of changing the topology of the power network by operating these switches for the purpose of minimization of the power loss. Since each switch has two conditions, a system which has N nodes should contain 2^{N-1} possible switch configurations. In order to ensure that all the customers can get electricity and no short circuit exists in the system, there are two constraints for network reconfiguration: no cycles (the radial structure of the network must be maintained in each new structure) and no islands (all the loads must be served).

The objective functions are the minimization of power loss and the maximization of the network's reliability–i.e. minimization of voltage deviation in this section. The objective function for the **minimization of power loss** can be described as [29]:

$$\min \quad f_{loss} = \sum_{i=1}^{b} k_i R_i \frac{P_i^2 + Q_i^2}{V_i^2} = \sum_{i=1}^{b} k_i R_i \mid I_i \mid^2 \tag{4-8}$$

subject to:

$$V_i^{min} \leq V_i \leq V_i^{max} \tag{4-9}$$

$$I_i \leq I_i^{max}, \; i = 1, ..., b \tag{4-10}$$

Here b is the number of branches and for each branch $i \in \{1, ..., b\}$, R_i is the

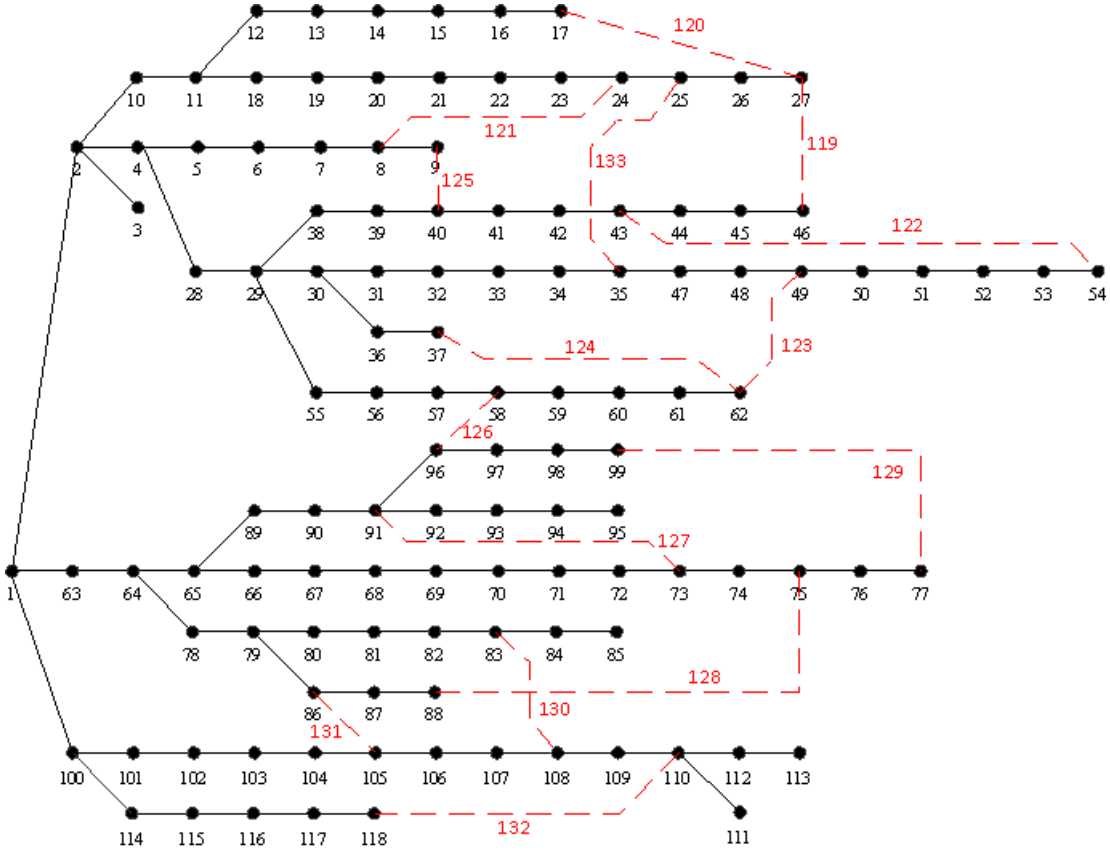

Figure 2.3: Initial configuration of the 119-bus test system.

branch resistance, P_i and Q_i are the active power and the inactive power of a branch terminal i, V_i is the terminal node voltage of branch i, V_i^{min} and V_i^{max} are the minimum and maximum bus voltage of branch i, respectively, k_i is the status variable of i-th switch. If k_i is 0, then switch i is open and if k_i is 1, then switch i is closed. I_i is the branch current and I_i^{max} is the maximum current in branch i.

The objective function for **minimization of voltage deviation** can be expressed as follows [30][31]:

$$\min \quad f_{VDI} = \max\{\mid 1 - U_{min} \mid, \mid 1 - U_{max} \mid\} \tag{4-11}$$

where U_{min} and U_{max} are respectively the lowest and highest values of bus voltage which is divided by rated voltage to normalize them to value in $[0, 1]$. In this dissertation, Newton's Method based on MATPOWER [32], which is a power flow calculation toolbox on MATLAB, is applied to calculate power loss and

voltage deviation. The parameters in Newton's methods are: maximum number of iterations is 20 and termination tolerance on per unit is 1e-8.

In this problem, a multi-objective optimization algorithm is achieved by replacing the selection scheme of a previously single objective optimizer by that of a multi-objective algorithm, namely the $(\mu+\mu)$ selection of NSGA-II [25] and the $(\mu+1)$ selection of SMS-EMOA [19] (which has earlier been used also in Pareto archivers [33]). As a second objective voltage deviation is minimized (see equation 4-11).

As an adaptation, we introduce a variant of SMS-EMOA and NSGA-II with a self-adaptive single step size. Whenever more than five mutations per individual were unsuccessful, the step size was multiplied by a constant factor of $1/1.2$ (following the $1/5$th success rule). The success of a generation was registered if a new non-dominated solution entered the archive of non-dominated solutions among all solutions encountered so far.

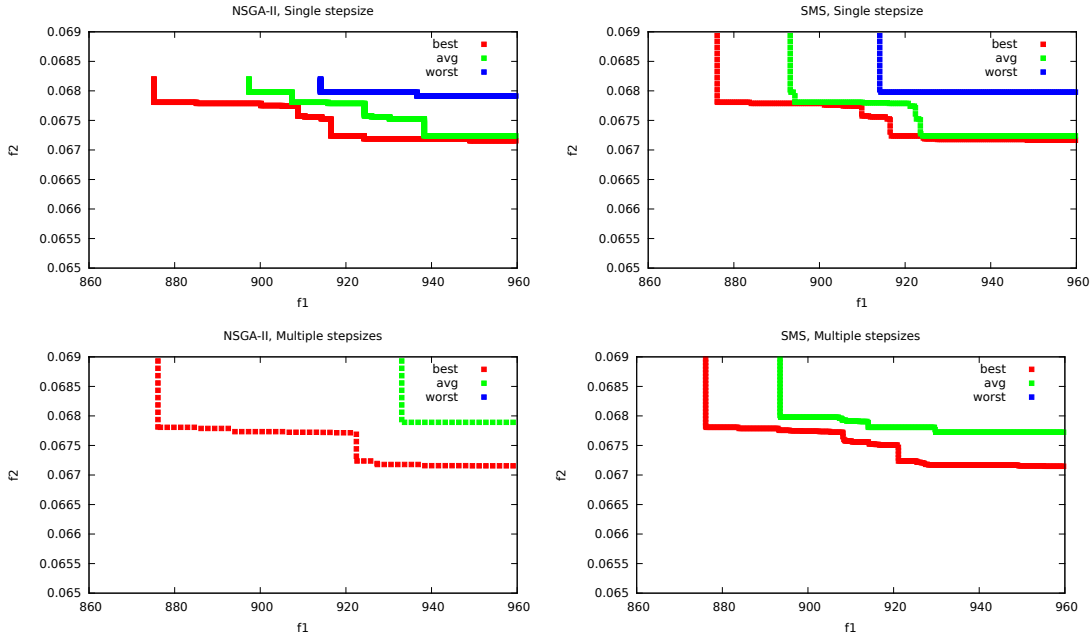

Figure 2.4: Best, worst, and average attainment curves for the multi-objective optimization of 119 DNRP. The worst attainment curves of the multiple stepsizes are missing in the below pictures, because they are too far away from the best curves.

The results (attainment curves) for a population size of $\mu = 30$ and 11 runs per algorithm are shown in Figure 2.4, where $f1$ and $f2$ represent voltage deviation

and power loss, respectively. Here, attainment curve[1] is a useful tool to assess the statistical performances of stochastic multi-objective optimizers, and visualize the outcome of a series of optimization trails [34, 35, 36].

In the following discussion of Pareto front, we mean the archive of all non-dominated solutions encountered in a single run. SMS-EMOA with a single step size provides the best Pareto fronts in the best case and also in the average case. Interestingly, all strategies find a Pareto front with a concave part, which is interpreted that locally there is a strong conflict between power loss minimization and voltage deviation minimization in this problem. However, the range of voltage deviation is relatively small, so that 'from a distance' the Pareto front has an apparent knee point region. Solutions in this region are recommended as good compromise solutions, whereas points located on the flanks of the Pareto front are not recommended, as small improvements in one objective will cause a large deterioration of the other objective.

2.5 Multi-objective Bayesian Global Optimization

In multi-objective optimization, the objective function evaluations are usually costly and evolutionary multi-objective optimization algorithms are usually not efficient to solve these expensive function evaluation problems. This is because EMOAs typically require a large number of function evaluations, furthermore, the Pareto optimality of the solutions cannot be guaranteed with fewer function evaluations. Therefore, EMOA is not recommended when function evaluations are expensive. In such cases, a better idea is perhaps to utilize information from all previous evaluations. This kind of algorithm is called *Bayesian Global Optimization* (BGO), which was proposed by Mockus et al. in [37].

The basic idea of BGO is to use a surrogate model based on Kriging or Gaussian process. A surrogate model reflects the relationship between decision vectors and their corresponding objective values. This surrogate model is learnt from the previous evaluations. For multi-objective problems, the family of these algorithms is called *Multi-Objective Bayesian Global Optimization* (MOBGO). The scheme of a MOBGO algorithm is sequentially updating a surrogate model, instead of 'true' objective functions, by an optimizer and its corresponding objective function value. An optimizer in MOBGO is utilized to search for a promising point $\mathbf{x}^*$ by maximizing/minimizing an infill criterion according to surrogate models.

[1] Attainment curve is also called attainment surface for more than two objectives.

2.5.1 Kriging

Kriging is a statistical interpolation method. Being a Gaussian process based modelling method, it is cheap to evaluate [38]. Kriging has been proven to be a popular surrogate model to approximate noise-free data in computer experiments, where Kriging models are fitted on previously evaluated points and then replace the real time-consuming simulation model [39]. Given a set of n decision vectors $\mathbf{X} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(n)})^T$ in m dimensional search space, and associated function values $y(\mathbf{X}) = (y(\mathbf{x}^{(1)}), y(\mathbf{x}^{(2)}), \cdots, y(\mathbf{x}^{(n)}))^T$, Kriging assumes y to be a realization of a random process Y and it is of the form [40, 41]:

$$Y(\mathbf{x}) = \mu(\mathbf{x}) + \epsilon(\mathbf{x}) \tag{5-12}$$

where $\mu(\mathbf{x})$ is estimated mean value over all given sampled points, and $\epsilon(\mathbf{x})$ is a realization of a normally distributed Gaussian random process with zero mean and variance σ^2. The regression part $\mu(\mathbf{x})$ approximates globally the function Y and Kriging/Gaussian process $\epsilon(\mathbf{x})$ takes local variations into account. Moreover, as opposed to other regression methods, such as supported vector machine (SVM), Kriging/GP also provides an uncertainty qualification of a prediction. The correlation between the deviations at two points ($\mathbf{x}$ and $\mathbf{x}'$) is defined as:

$$Corr[\epsilon(\mathbf{x}), \epsilon(\mathbf{x}')] = R(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^{d} R_i(x_i, x_i') \tag{5-13}$$

Here $R(.,.)$ is the correlation function, which can be cubic or spline function. Commonly, a Gaussian function (also known as squared exponential) is chosen:

$$R(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^{d} \exp(-\theta_i (x_i - x_i')^2) \qquad (\theta_i >= 0)$$

where θ are parameters of correlation model and they can be interpreted as measuring the importance of the variable. Then the covariance matrix can be expressed by the correlation function:

$$Cov(\boldsymbol{\epsilon}) = \sigma^2 \boldsymbol{\Sigma}, \qquad \text{where} \qquad \boldsymbol{\Sigma}_{i,j} = R(\mathbf{x_i}, \mathbf{x_j})$$

When $\mu(\mathbf{x})$ is assumed to be an unknown constant, this unbiased prediction is called ordinary Kriging (OK). In OK, the Kriging model determines the hyperparameters $\theta = [\theta_1, \theta_2, \cdots, \theta_n]$ by maximizing the likelihood on the observed dataset. The expression of the likelihood function is:

$$L = -\frac{n}{2} \ln(\sigma^2) - \frac{1}{2} ln(|\boldsymbol{\Sigma}|)$$

The maximum likelihood estimates of the mean $\hat{\mu}$ and the variance $\hat{\sigma}^2$ are given by:

$$\hat{\mu} = \frac{\mathbf{1}_n^T \mathbf{\Sigma}^{-1} \mathbf{y}}{\mathbf{1}_n^T \mathbf{\Sigma}^{-1} \mathbf{1}_n}$$
$$\hat{\sigma}^2 = \frac{1}{n}(\mathbf{y} - \mathbf{1}_n \hat{\mu})^T \mathbf{\Sigma}^{-1} (\mathbf{y} - \mathbf{1}_n \hat{\mu})$$

Then the predictor of mean and variance at point x^t can be derived and they are shown as follows [41]:

$$\mu(\mathbf{x}^t) = \hat{\mu} + \mathbf{c}^T \mathbf{\Sigma}^{-1} (\mathbf{y} - \hat{\mu} \mathbf{1}_n)$$
$$\sigma^2(\mathbf{x}^t) = \hat{\sigma}^2 [1 - \mathbf{c}^T \mathbf{\Sigma}^{-1} \mathbf{c} + \frac{1 - \mathbf{c}^T \Sigma^T \mathbf{c}}{\mathbf{1_n^T \Sigma^{-1} 1_n}}]$$

where $\mathbf{c} = (Corr[Y(x^t), Y(x_1)], \cdots, Corr[Y(x^t), Y(x_n)])^T$.

2.5.2 Structure of MOBGO

Compared to multi-objective evolutionary algorithms, MOBGO requires only a small budget of function evaluations [10]. As a result, it has already been implemented in the real world optimization for expensive evaluation problems. According to the authors' knowledge, it was for the first time used in the context of airfoil optimization [42]. Later, it was applied in the field of biogas plant controllers [43], in the detection of water quality management [44], in the structural design optimization [45] and could be implemented in the other real-world optimization problems with expensive evaluations.

Multi-Objective Bayesian Global Optimization is assuming that d objective functions are mutually independent in an objective space. In MOBGO, the Kriging method or Gaussian process can approximate Kriging models M with respect to objective functions and the uncertainties of the prediction, from the existing evaluated data $D = \big((\mathbf{x}^{(1)}, \mathbf{y}^{(1)} = Y(\mathbf{x}^{(1)})), \ldots, (\mathbf{x}^{(\mu)}, \mathbf{y}^{(\mu)} = Y(\mathbf{x}^{(\mu)}))\big)$. Each objective function at a given point $\mathbf{x}^{(t)}$ is approximated by a one-dimensional normal distribution, with mean μ and standard deviation σ. Then MOBGO can predict the multivariate outputs by means of an independent joint normal distribution with parameters $\mu_1, \ldots, \mu_d$ and $\sigma_1, \ldots, \sigma_d$ at the point $\mathbf{x}^{(t)}$.

These predictive means and standard deviations can be used to calculate infill criteria. An *infill criterion* measures how promising a new point is, when compared to a current Pareto-front approximation. With the assistance of a single

objective optimization algorithm, 'optimal' solution $\mathbf{x}^*$ can be found according to the score of the infill criterion. This score of the infill criterion is calculated by the predictions of the Kriging models, instead of by the direct objective functions. Then, the 'optimal' solution $\mathbf{x}^*$ is evaluated, and both the dataset D and the Pareto-front approximation set $\mathcal{P}$ are updated.

Algorithm 3: MOBGO algorithm

Input: Objective functions $\mathbf{y}$, initialization size μ, termination criterion T_c

Output: Pareto-front approximation $\mathcal{P}$

1: Initialize μ points $(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(\mu)})$;
2: Evaluate the initial set of μ points: $(\mathbf{y}^{(1)} = \mathbf{y}(\mathbf{x}^{(1)}), \ldots, \mathbf{y}^{(\mu)} = \mathbf{y}(\mathbf{x}^{(\mu)}))$;
3: Store $(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(\mu)})$ and $(\mathbf{y}^{(1)} = \mathbf{y}(\mathbf{x}^{(1)}), \ldots, \mathbf{y}^{(\mu)} = \mathbf{y}(\mathbf{x}^{(\mu)}))$ in D: $D = ((\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(\mu)}, \mathbf{y}^{(\mu)}))$;
4: Compute the non-dominated subset of D and store it in $\mathcal{P}$;
5: $g = 1$;
6: **while** $g <= T_c$ **do**
7: Train surrogate models M based on D;
8: Use an optimizer to find the promising point $\mathbf{x}^*$ based on surrogate models M, with the infill criterion C;
9: Update D: $D = D \cup (\mathbf{x}^*, \mathbf{y}(\mathbf{x}^*))$;
10: Update $\mathcal{P}$ as non-dominated subset of D;
11: $g = g + 1$;
12: **end while**
13: Return $\mathcal{P}$.

The basic structure of the MOBGO algorithm is shown in Algorithm 3. It mainly contains three parts: initialization, updating and searching, and returning.

Firstly, the dataset D is initialized and the Pareto-front approximation set $\mathcal{P}$ is calculated, as shown in Algorithm 3 from Step 1 to Step 5. The initialization of D contains generation of the decision vectors (Step 1), calculation of the corresponding objective values (Step 2) and storage of this information in data set D (Step 3). This data set D will be utilized to build the Kriging models in the second part.

The second part of MOBGO is the main loop, as shown in Algorithm 3 from Step 6 to Step 12. In this main loop, it is started by training the Kriging models M based on data set D (Step 6). Please note that M contains d independent models for

each objective function, and these models would be used as temporary objective functions instead of 'true' objective functions at Step 7. Then, an optimizer can find the promising point $\mathbf{x}^*$ by maximizing or minimizing an infill criterion C (Step 7). Here, an infill criterion is calculated by its corresponding calculation formula, as the inputs of Kriging models M, Pareto-front approximation $\mathcal{P}$, decision vector $\mathbf{x}$, etc. In Step 8, a single-objective optimization algorithm is required to find the promising point $\mathbf{x}^*$ for each temporary objective function – i.e., surrogate model. In this dissertation, the BI-Population CMA-ES has been chosen as the optimizer to find the promising point $\mathbf{x}^*$, considering its favorable theoretical properties [46]. After finding the promising point $\mathbf{x}^*$, Step 9 and Step 10 will update the dataset D by adding $(\mathbf{x}^*, \mathbf{y}(\mathbf{x}^*))$ into D and update the Pareto-front approximation $\mathcal{P}$, respectively. The main loop from Step 6 to Step 12 will not stop until g meats the termination criterion T_c.

The last part of MOBGO is the return of Pareto-front approximation $\mathcal{P}$.

In single objective Bayesian Global Optimization, some common infill criteria include the *Expected Improvement* (EI) [41, 47], *Probability of Improvement* (PoI) [48, 49], and *Lower Confidence Bounds* (LCB) [50, 51, 52, 53]. In *Multi-Objective Bayesian Global Optimization*, some common *infill criteria* are: *Hypervolume Indicator* (HV) [54], *Probability of Improvement* (PoI) [48, 49, 55], *Hypervolume Improvement* (HVI) [27][1], *Euclidean distance-based EI* [55], *Hypervolume Contribution* (HVC) [19], *Expected Hypervolume Improvement* (EHVI) [47, 56], *Tchebycheff aggregation based EI* (EA-EI)[57], *Hypervolumme based PoI* [58], *Truncated Expected Hypervolume Improvement* (TEHVI) [6, 7], and *EI of penalty-based boundary intersection* (PBI-EI) [59].

2.5.3 Example

The behavior of the BGO based on the expected hypervolume improvement will be illustrated by a single numerical experiment.

The numerical example is visualized in the plots of Figure 2.5. The bicriteria optimization problem from Emmerich et al. [60] is: $f_1(\mathbf{x}) = ||\mathbf{x} - \mathbf{1}|| \to \min$, $f_2(\mathbf{x}) = ||\mathbf{x} + \mathbf{1}|| \to \min$, $\mathbf{x} \in [-2, 2] \times [-2, 2] \subset \mathbb{R}^2$. The Pareto front is the line segment from $(0, 2 \cdot \sqrt{2})$ to $(2 \cdot \sqrt{2}, 0)$, the efficient set is the line segment that connects $(-1, -1)$ and $(1, 1)$. The metamodel used is a Gaussian random field model with Gaussian correlation function $\exp(-\theta||\mathbf{x}^{(1)} - \mathbf{x}^{(2)}||^2)$, for $\mathbf{x}^{(1)} \in \mathbb{R}^m$

[1] The HVI was called the most likely improvement (MLI) in [27].

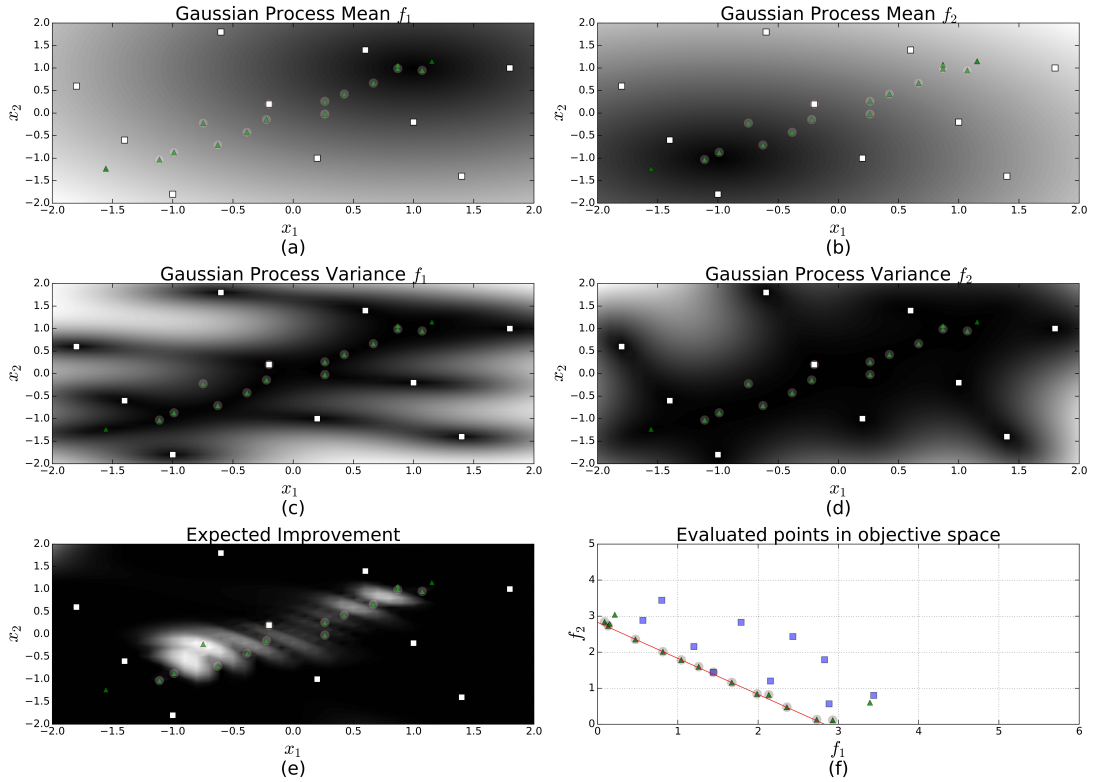

Figure 2.5: Example Run of Multicriteria Bayesian Global Optimization.

and $\mathbf{x}^{(2)} \in \mathbb{R}^m$, here $m = 2$. We set $\theta = 0.0001$, which was estimated by maximum likelihood method for initial sample. An initial set of 10 points was evaluated indicated by the dark blue squares in Figure 2.5 (f). From this starting set 15 new points were generated using the expected hypervolume improvement. The maximizer of the expected improvement was found using a uniform grid. In total, each objective function was evaluated 25 times.

The results of the experiment are depicted in plots. In all pictures, points that have been evaluated are indicated by triangles. The points from the initial set are additionally marked by squares. Efficient points are surrounded by circles. The figures in the the top row, Figure 2.5 (a) and (b), depict the mean value of the Gaussian random field model at $\mathbf{x} \in [-\mathbf{2}, \mathbf{2}] \times [-\mathbf{2}, \mathbf{2}]$ for f_1 and f_2, respectively. Likewise, the figures in the middle row, Figure 2.5 (c) and (d), depict the variance of the Gaussian random field model at $\mathbf{x} \in [-\mathbf{2}, \mathbf{2}] \times [-\mathbf{2}, \mathbf{2}]$ for f_1 and f_2, respectively. The expected hypervolume improvement values after 25 iterations are shown in Figure 2.5 (e). The final set of points in the objective space and the Pareto front approximation can be found in Figure 2.5 (f). After

only *25 evaluations of the original objective functions*, the algorithm finds a good approximation to the Pareto front.

2.6 Summary

Most practical optimization problems involve more than one objective, simply due to the fact that no product, process, or system can be assessed with a single criterion. Since, frequently, there are conflicting criteria, such as minimizing cost and maximizing the quality of a product, Multi-Objective Optimization problems give rise to a set of trade-off optimal (known as Pareto optimal) solutions.

In this chapter, we defined vital terms that are used in MOBGO research, introduced some infill criteria in MOO, described two state-of-the-art EMOAs with a practical problem, and introduced the structure of MOBGO, together with a simple example for the illustration. This chapter only defines the fundamental terminologies of this research. Other related terminologies are represented in each chapter.