



Model-based evolutionary algorithms: a short survey

Ran Cheng¹ · Cheng He¹ · Yaochu Jin² · Xin Yao^{1,3}

Received: 18 June 2018 / Accepted: 31 July 2018 / Published online: 7 August 2018
© The Author(s) 2018

Abstract

The evolutionary algorithms (EAs) are a family of nature-inspired algorithms widely used for solving complex optimization problems. Since the operators (e.g. crossover, mutation, selection) in most traditional EAs are developed on the basis of fixed heuristic rules or strategies, they are unable to learn the structures or properties of the problems to be optimized. To equip the EAs with *learning abilities*, recently, various model-based evolutionary algorithms (MBEAs) have been proposed. This survey briefly reviews some representative MBEAs by considering three different motivations of using models. First, the most commonly seen motivation of using models is to estimate the distribution of the candidate solutions. Second, in evolutionary multi-objective optimization, one motivation of using models is to build the inverse models from the objective space to the decision space. Third, when solving computationally expensive problems, models can be used as surrogates of the fitness functions. Based on the review, some further discussions are also given.

Keywords Model-based evolutionary algorithms · Estimation of distribution algorithms · Surrogate modelling · Inverse modelling

Part of the work was supported in part by an EPSRC Grant (No. EP/M017869/1), and in part by the Ministry of Science and Technology of China Grant (No. 2017YFC0804003) and the Science and Technology Innovation Committee Foundation of Shenzhen Grant (No. ZDSYS201703031748284).

✉ Ran Cheng
ranchengcn@gmail.com
Cheng He
chenghehust@gmail.com
Yaochu Jin
yaochu.jin@surrey.ac.uk
Xin Yao
xiny@sustc.edu.cn

- ¹ Shenzhen Key Laboratory of Computational Intelligence, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China
- ² Department of Computer Science, University of Surrey, Guildford GU2 7XH, UK
- ³ Center of Excellence for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK

Introduction

With the development of modern science and engineering, the optimization problems in various areas are becoming increasingly challenging. Without loss of generality, an optimization problem (for minimization, with box constraints) can be formulated as [1]

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})), \\ \text{s.t.} \quad & \mathbf{x} \in X, \quad \mathbf{f} \in Y, \end{aligned} \quad (1)$$

where $X \subset \mathbb{R}^n$ and $\mathbf{x} = (x_1, x_2, \dots, x_n) \in X$ denote the decision space and the decision vector, respectively; $Y \subset \mathbb{R}^m$ and $\mathbf{f} \in Y$ denote the objective space and the objective vector, respectively. The decision vector \mathbf{x} comprises n decision variables, and the objective vector \mathbf{f} comprises m objective functions which map \mathbf{x} from X to Y . If there is only one objective, i.e., $m = 1$, the problems are often known as single-objective optimization problem (SOPs); while if there is more than one objective function, i.e., $m > 1$, the problems are often known as *multi-objective optimization problems* (MOPs) [2]. For SOPs, there usually exist at least one global optimal solution that optimizes the given objective function. For MOPs, however, there does not exist a single solution that optimizes all the objectives simultaneously, and by contrast,

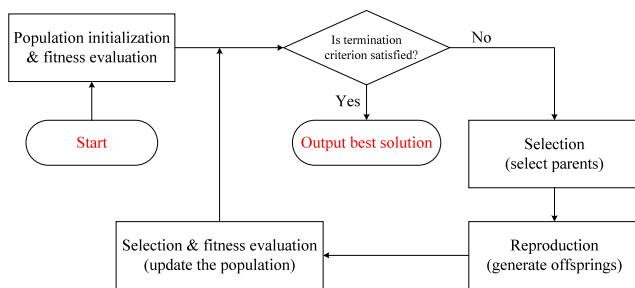


Fig. 1 The general framework of the evolutionary algorithms

there exist a set of optimal solutions that trade off between different objectives, where the image of the solution set is known as the Pareto set (PS) and the Pareto front (PF) in the decision space and objective space, respectively.

Due to the complex properties of real-world optimization problems, mathematical methods such as the Newton's method [3] and the hill climbing method [4] fail to work effectively on them. By contrast, the evolutionary algorithms (EAs) [5] show generally robust performance on those complex optimization problems. Generally, the family of EAs refers to the population-based stochastic algorithms inspired by natural evolution, include the genetic algorithm (GA) [6], the evolutionary programming (EP) [7,8], the evolution strategy (ES) [9] and the genetic programming (GP) [10], as well as the differential evolution (DE) [11]. Besides, the recently developed swarm intelligence (SI) algorithms such as the particle swarm optimization (PSO) [12] and ant colony optimization (ACO) [13] are also regarded as new members of the EA family.

In spite of the different technical details adopted in different EAs, most of them share a common framework as given in Fig. 1. Each generation in the main loop of a typical EA consists of the following components: reproduction, fitness evaluation and selection. To be more specific, the reproduction process, which generates new candidate solutions, often adopts the so-called genetic operators such as crossover and/or mutation; the fitness evaluation process indicates the quality of the candidate solutions in the current population by assigning fitness values; and the selection operator determines which candidate solutions can survive to the next generation. Traditionally, the operators in EAs are developed on the basis of some fixed heuristic rules or strategies, but do not interact with the environment.¹ However, during the evolution process, the environment can vary rapidly due to the complicated properties of the problem to be optimized. In this case, traditional operators may not work effectively due to the failure of adaptively adjusting the behaviors. In other

words, traditional EA operators are unable to *learn* from the environment.

To address the above issue, a number of recent works have been dedicated to proposing EAs with learning ability. The basic idea is to replace the heuristic operators with machine-learning models, where the candidate solutions are used as the training data sampled from the current environment in each generation. For different purposes, the machine-learning models can be embedded into any of the three main components in EAs, i.e., reproduction, fitness evaluation or selection. To be specific, the adopted machine-learning (ML) models can be regression models (e.g. the Gaussian process (GP) [10], artificial neural network (ANN) [14]), clustering models (e.g. the K-means [15]), classification models (e.g. the support vector machine (SVM) [16]), dimensionality reduction models (e.g. the principle component analysis (PCA) [17]), etc.

In spite of the various technical details, we find three main motivations of using ML models in EAs: (1) building estimation models in the decision space, (2) building inverse models to map from the objective space to the decision space, and (3) building surrogate models for the fitness functions. By considering the different motivations, we hope to conduct a short survey to not only provide a systematic summary of some representative works but also discuss the potential future research directions in the related field. Without loss of generality, we refer to the EAs using ML models as the *model-based evolutionary algorithms (MBEAs)* hereafter.

The rest of this survey is summarized as follows. “Estimation of distribution” reviews the MBEAs motivated to estimate the distribution in the decision space. “Inverse modeling” reviews the MBEAs motivated to build inverse models from the objective space to the decision space. “Surrogate modeling” reviews the MBEAs motivated to build surrogate models for the fitness functions. Finally, the last section summarizes this survey.

Estimation of distribution

Estimation of distribution algorithms (EDAs) refer to the MBEAs which estimate the distribution of the promising candidate solutions by training and sampling models in the decision space [18]. As given in Algorithm 1, EDAs still use the conventional framework of EAs, but the reproduction operators such as crossover and mutation are replaced by ML models. Ideally, the ML models in EDAs are iteratively refined as the evolution proceeds and eventually converged to the global optimum. In this section, we will present a brief overview of representative EDAs of three types: the univariate EDAs, the multivariate EDAs and the multi-objective EDAs.

¹ From the optimization point of view, the environment is also known as the fitness landscape.

Algorithm 1 Pseudo code for EDA approach.

```

1:  $D_0 \leftarrow$  Initialize  $M$  individuals (the initial population) at random
2: for  $l = 1, 2, \dots, T$  do
3:    $D_{l-1}^{Se} \leftarrow$  Select  $N \leq M$  individuals from  $D_{l-1}$  according to the
     selection method
4:    $p_l(\mathbf{x} \| D_{l-1}^{Se}) \leftarrow$  Estimate the probability distribution of an indi-
     vidual being among the selected individuals
5:    $D_l \leftarrow$  Sample  $M$  individuals (the new population) from  $p_l(\mathbf{x})$ 
6: end for

```

Univariate EDAs

To estimate the distribution of the candidate solutions in the decision space, the variable correlation is an essential factor to be taken into consideration in modeling. A simple approach is to adopt univariate models, where the assumption is that the decision variables are independent. Based on this assumption, the probability distribution of a candidate solution $\mathbf{x} = (x_1, x_2, \dots, x_n)$ can be decomposed as

$$p(\mathbf{x}) = p(x_1)p(x_2) \cdots p(x_n), \quad (2)$$

where $p(\mathbf{x})$ is the probability distribution of the candidate solution \mathbf{x} , and $p(x_i)$ is the probability distribution of decision variable x_i . EDAs adopting such univariate distribution models are known as the univariate EDAs.

As a classic univariate EDA, the univariate distribution algorithm (UMDA) was proposed to solve the well-known onemax problem [19]. UMDA adopts a binary-encoded probability model with a probability vector represented as $p = (p_1, p_2, \dots, p_n)$, where $p_i = 1$ indicates the probability of having 1 at position i of a candidate solution.

Another representative univariate EDA is known as the population-based incremental learning (PBIL) algorithm [20], which uses a similar binary-encoded probability model as in UMDA. However, a main difference lies in the fact that PBIL aims to incrementally improve the probability model by sampling a small number of candidate solutions in each generation, while UMDA maintains a full population of candidate solutions.

Despite that the univariate EDAs are of high computational efficiency by building univariate models, their performance may sharply deteriorate if there exist strong interactions between the decision variables. In the following subsection, we will briefly review some multivariate EDAs motivated to address this issue.

Multivariate EDAs

Among various multivariate EDAs, the most intuitive approach is to consider the pair-wise interactions between decision variables. Generally, given a candidate solution $\mathbf{x} = (x_1, x_2, \dots, x_n)$, the pair-wise interactions can be presented by the conditional probability model as

$$p(\mathbf{x}) = p(x_{i_1}|x_{i_2})p(x_{i_2}|x_{i_3}) \cdots p(x_{i_{n-1}}|x_{i_n})p(x_{i_n}), \quad (3)$$

where $p(\mathbf{x})$ is the probability distribution of the candidate solution \mathbf{x} , $p(x_{i_j}|x_{i_{j+1}})$ is the conditional probability distribution of x_{i_j} given $x_{i_{j+1}}$, and i_1, i_2, \dots, i_n denotes a permutation of the decision variables.

For example, De et al. proposed a mutual-information-maximizing input clustering (MIMIC) algorithm in [21]. In MIMIC, the conditional entropy of each decision variable is used as the information for building the conditional probability models, where a chain of dependencies is built according to the ascent order of the conditional entropy values of the candidate solutions. Another representative algorithm of this type is the bivariate marginal distribution algorithm (BMDA) [22], where the dependency model at each generation is built by considering Person's chi-square statistics as the dependence measure.

While the conditional probability model as given in (3) is only capable of presenting the pair-wise interactions, some problems may contain more complicated interactions between the decision variables. To model such complicated interactions, a classic approach is the Bayesian optimization algorithm (BOA) [23] which adopts the Bayesian networks as the multivariate models. A Bayesian network is an acyclic-directed graph, where each decision variable is represented by a node, and the conditional dependencies between the decision variables are presented by the edges. Given a decision vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$, a Bayesian network can be formulated as

$$p(\mathbf{x}) = \prod_{i=1}^n p(x_i | x_{\text{edge}(i)}), \quad (4)$$

where $\text{edge}(v)$ is the set of variables having edges connected to x_i . To build a Bayesian network, BOA starts from a single node and iteratively adds edges to the network according to the Bayesian–Dirichlet metric [24].

Since the Bayesian networks are able to capture complex variable interactions, many other representative multivariate EDAs are also developed on the basis of it. In the estimation of Bayesian network algorithm (EBNA), the Bayesian information criterion [25] is adopted in the iterative construction of the Bayesian networks. In the hierarchical Bayesian Optimization Algorithm (hBOA) [26], a problem is decomposed into a group of subproblems, and a hierarchical structure is adopted to deal with different subproblems in multiple levels.

There are also some other multivariate EDAs using different types of models, such as the Markovianity-based optimization algorithm (MOA) using the Markov networks [27], the affinity propagation EDA (AffEDA) using the affinity propagation clustering method [28], etc.

Multi-objective EDAs

Apart from the single-objective EDAs as discussed above, there are also EDAs tailored for solving MOPs, known as the multi-objective EDAs (MEDAs). Instead of obtaining one global optimum, the MEDAs are expected to obtain a set of optimal solutions as an approximation to the PF (as well as PS).

To approximate the PF an MOP, most MEDAs adopt special mechanisms to balance the convergence and diversity of the candidate solutions. In the Bayesian multi-objective optimization algorithm (BMOA) [29], the selection operator is based on a ϵ -archive, where a minimal set of candidate solutions that ϵ -dominates all the others is maintained over generations. In the naive mixture-based multi-objective iterated density estimation evolutionary algorithm (MIDEA) [30], a two-phase selection pressure is adopted, where the selection pressure is tuned by a parameter δ . In the multi-objective Bayesian optimization algorithm (mBOA) [31], the selection operator is directly borrowed from the NSGA-II algorithm [32]. The multi-objective hierarchical BOA (mohBOA) also adopts the selection operator in NSGA-II, combined with a k-meas clustering method.

Different from most MEDAs that adopt new selection operators, the regularity model-based multi-objective estimation of distribution algorithm (RM-MEDA) adopts a new reproduction operator [33]. Since the PS is a piecewise continuous manifold under the Karush–Kuhn–Tucker optimality conditions (aka the regularity property) [34], RM-MEDA reduces the dimensionality of the decision vectors using the local PCA method and then samples new candidate solutions in the latent space.

Discussion

As the most commonly seen MBEAs, the EDAs have achieved considerable advances over the past decade. As a main advantage, the EDAs have potential abilities to adapt to the fitness environment and learn the problem structures. This is helpful when the problems to be optimized have some special properties. Nevertheless, some challenges still remain to be addressed.

First, compared to using heuristic strategies (e.g. two-point crossover), it is generally more time consuming to build ML models. It should be well traded off in practice whether it is worth the computational cost to apply EDAs, maybe only for incremental performance improvement.

Second, most EDAs have strict requirement of the training data (i.e. candidate solutions to have the models adequately trained). This can hardly be guaranteed during the optimization process of an EA. Consequently, ill-trained models may lead to poor performance of EDAs.

Third, most EDAs suffer a serious curse of dimensionality. With the increase of the decision variables, the performance of EDAs may deteriorate sharply due to the failure of the ML models adopted therein. This has limited the robustness and applicability of EDAs in practice.

Fourth, EDAs focus on the estimation of the distribution in the decision space, but they pay little attention to the correlation among the decision variables. By contrast, the covariance matrix adaptation (CMA)-based algorithms [35], e.g., the CMA-based evolutionary strategy [36] and the multi-objective CMA (MO-CMA) [37], utilize the correlation and variance quotients of the distribution to enhance the convergence of the algorithm. A promising future work is the combination of EDA and CMA to take full advantage of the statistical information for accelerating the convergence rate of EDAs.

Inverse modeling

As discussed in the previous subsection, the target of multi-objective optimization is to obtain a set of candidate solutions as trade-offs between the different objectives. Hence, an algorithm should maintain a good balance between the convergence and diversity of the population, such that, ideally, the candidate solutions can be uniformly distributed on the true PF. Despite that the target is to approximate the PF (in the objective space), most MEDAs still build models in the decision space and sample candidate solutions. However, as illustrated in Fig. 2, a uniformly distributed solution set in the decision space may not necessarily mean that their image set is also uniformly distributed on the PF. To directly control the distribution of the candidate solutions on the PF, some researchers have proposed to first sample points in the decision space and then build inverse models to map them back to the decision space. In this section, we will introduce several representative MBEAs of this type.

Given an MOP $\mathbf{f}(\mathbf{x})$, the inverse modeling process is to build a model that maps from the objective space to the decision space as

$$g(\mathbf{f}(\mathbf{x})) = \mathbf{x}, \quad (5)$$

where $g(\cdot)$ denotes the inverse mapping function to be modeled. Strictly speaking, $g(\cdot)$ can be precisely modeled if and only if it is a one-to-one mapping from the PF to the PS. In practice, however, $g(\cdot)$ still can be approximated even if this condition does not hold. From the machine-learning point of view, building an inverse model $g(\cdot)$ can be seen as a regression task.

In [38], Giagkiozis and Fleming propose to use the Radial Basis Function Neural Networks (RBFNNs) [39] to build the inverse models mapping from the objective space to the

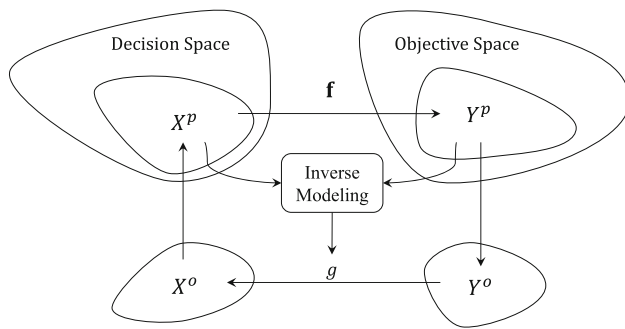


Fig. 2 Illustration of one generation of the IM-MOEA algorithm. X^p and Y^p denote the decision vectors and the objective vectors of the parent population, and X^o and Y^o denote the decision vectors and the objective vectors of the offspring population

decision space for multi-objective optimization, known as the Pareto estimation method. In this method, an existing multi-objective evolutionary algorithm is first run to obtain a set of candidate solutions as an approximation to the PF. Then, the solution set is used for training the RBFNN mapping from the objective space to the decision space. Using the trained RBFNNs, the decision makers are able to sample more solutions in the region of interest on the PF without performing additional exhaustive search.

While the Pareto estimation method in [38] only works for off-line training using a solution set obtained by another existing algorithm, recently, Cheng et al. have proposed a multi-objective evolutionary algorithm using Gaussian process-based inverse modeling (IM-MOEA) [40,41], which adopts on-line training of the inverse models. As illustrated in Fig. 2, the inverse modeling process is embedded into the reproduction operator of the algorithm to sample new candidate solutions. To simplify the modeling process, the whole multivariate inverse mapping model is approximately decomposed into a number of univariate models:

$$P(\mathbf{x}|\mathbf{f}(\mathbf{x})) \approx \prod_{i=1}^n \prod_{j=1}^m P(x_i|f_j). \tag{6}$$

Since the decomposition does not strictly hold due to the variable correlations, the Gaussian process (GP) [10] models are used to present the uncertainty information (i.e. errors of the decomposition). In addition, a random grouping method is used to increase the probability that the correlated variables are considered together when training and sampling the inverse models. Compared to the RBFNN-based method in [38], the IM-MOEA shows more robust performance, and more importantly, it can not only be used to sample additional solutions in the region of interests, but also work as an independent EA to approximate the PF. Following the success of IM-MOEA, the idea has also been extended for solving MOPs with irregular PFs [42] as well as MOPs in dynamic environment [43]. Instead of adopting the GP model, an sim-

ple linear model is adopted in [43] to simplify the inverse modeling process.

Apart from the aforementioned inverse modeling-based approaches, there are also some approaches focused on the PF modeling only. For example, in the Pareto-adaptive ϵ -dominance-based algorithm (*pa* λ -MyDE) [44,45] and the reference indicator-based MOEA (RIB-EMOA) [46], each PF is associated with one curve in the family:

$$\left\{ f_1^p + f_2^p + \dots + f_M^p = 1 : 0 \leq f_1, \dots, f_M \leq 1, p > 0 \right\},$$

where f_i denotes the i th objective value and M denotes the number of objectives. Recently, Tian et al. proposed a robust Pareto front modeling method [47] by training a generalized simplex model in consideration of both the scale and curvature of the PF. However, despite that these approaches are capable of capturing the approximate structures of the PFs, the models cannot be used to obtain the candidate solutions in the decision space directly, which is a major difference from the inverse modeling-based approaches.

Discussion

While the EDAs are focused on the estimation of the distribution in the decision space, the inverse modeling works as a bridge between the objective space and decision space. It is particularly useful to build such inverse models when there is a decision-making processes involved in multi-objective optimization. Nevertheless, the development of inverse modeling is still at the infancy and there is much to be explored and studied in the future.

First, inverse modeling is based on the assumption that the mapping from the objective space to the decision space is one-to-one mapping. In practice, however, it is very likely that one objective vector can correspond to more than one decision vectors. It is of particular interest to see how to build more robust inverse models for such problems.

Second, just as most other MBEAs, the inverse modeling-based algorithms also suffer from the curse of dimensionality. This issue is twofold. On one hand, the ML models such as GP can be extremely time consuming if there is a large number of variables. On the other hand, the training data required for building the inverse models exponentially increase with the number of variables, which, however, cannot be met due to the limited population size and fitness evaluations.

Surrogate modeling

One great challenge in solving many real-world optimization problems is that one single fitness evaluation (FE) is computationally and/or financially expensive, since it requires

time-consuming computer simulation or physical experiments [48]. For instances, the computational fluid dynamic (CFD) simulation is used to estimate the quality of a design scheme in the field of structural design, where a single simulation may take from minutes to hours [49–51]. Conventional model-free EAs cannot afford such expensive function evaluations, as they typically require tens of thousands real-objective FEs. To overcome this challenge, the surrogate-assisted evolutionary algorithms (SAEAs) have been developed, where the computationally efficient models are introduced for replacing the computationally expensive models [52–55]. Generally speaking, the SAEAs are also a class of typical MBEAs where the models are used in the fitness evaluation component.

In this section, we will present a brief overview of representative SAEAs of two types: the single-objective SAEAs and the multi-objective SAEAs.

Single-objective SAEAs

In expensive single-objective optimization, the surrogate model typically aims to approximate the objective function or a fitness function of a candidate solution \mathbf{x}

$$\hat{f}(\mathbf{x}) = f^*(\mathbf{x}) + \xi(\mathbf{x}), \quad (7)$$

where f^* is the true value of the objective or fitness value of the solution, \hat{f} is the approximated value, and ξ is the error function which reflects the degree of “uncertainty” of the approximation of the model [56].

The model management plays a key role in making the most use of the surrogate models [57]. Existing model management strategies can be roughly classified into three categories, namely, the generation-based, the population-based and the individual-based strategies [52]. Most earlier model management strategies employ a generation-based method [58,59], where the key question is to adopt the frequency in which the real fitness function is used. For example, Nair et al. used the average approximation error of the surrogate during one control cycle to adjust the frequency of using the real objective function [60]. In the population-based approaches, more than one subpopulation co-evolves, each using its own surrogate for fitness evaluations and the migration of individuals from one subpopulation to another is allowed. For example, Sefrioui et al. proposed a Hierarchical Genetic Algorithm (HGA) using multiple models [61]. By contrast, the individual-based model management [57,62] focuses on determining which individuals need to be evaluated within each generation. The most straightforward criterion is to evaluate solutions that have the best fitness according to the surrogate [57]. Emmerich et al. proposed a criterion to select solutions whose estimated fitness was the

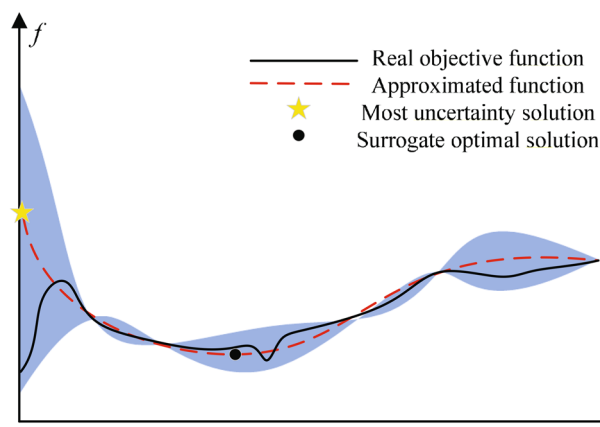


Fig. 3 An example illustrating the utility of the “uncertainty” information. The star point is the most uncertain candidate solution and the circle point is the optimal candidate solution found by the surrogate model, while the shaded area characterizes the “uncertainty” degree of the approximated function

most uncertain [62,63], which could encourage exploration of the algorithm.

Usually, researchers tend to select those “certain” candidate solutions to achieve relatively accurate prediction. However, some “uncertain” candidate solutions may be potentially good [64]. An example of this situation is given in Fig. 3, the most uncertain solution (the star point) corresponds to the global optimal solution of the problem, while the predicted best solution (the circle point) corresponds to a local optimum.

To estimate the uncertainty in fitness approximation, the average distance between a solution and the training data can be adopted [62]. Since Kriging models are able to provide uncertainty information in the form of a confidence level of the predicted fitness [48], they have recently become very popular in SAEAs. To take advantage of the uncertainty information provided by the Kriging models, various model management criteria, also known as infill sampling criteria in the Kriging-assisted optimization, have been proposed in SAEAs, such as the probability of improvement (PoI) [65,66], the expected improvement (ExI) [67], the lower confidence bound (LCB) [64], and the heterogeneous ensemble-based infill criterion to enhance the reliability of ensembles for uncertainty estimation [68].

Apart from the aforementioned single-objective SAEAs that work in the context of genetic algorithm, there are also some SAEAs working in stochastic search methods rather than genetic algorithms, such as the surrogate-assisted artificial immune systems [69], the neural network-assisted evolution strategy [57], the feasibility structure modeling-assisted memetic algorithm [70], the classification-assisted memetic algorithm [71], the surrogate-assisted cooperative particle swarm optimization [72], and the committee-based active learning based surrogate-assisted particle swarm optimizer [73].

Multi-objective SAEAs

In recent years, computationally expensive MOPs have drawn increasing attention in the area of expensive optimization as they are difficult for existing SAEAs [74].

Different from the single-objective SAEAs using surrogate models for approximating the objective functions or a fitness function, a variety of different targets can be approximated in multi-objective SAEAs. The most intuitive idea is to approximate the objective functions by multiple surrogate models [59,75,76]. For instances, Singh et al. proposed a surrogate-assisted simulated annealing algorithm (SASA) for constrained multi-objective optimization [77], Ahmed and Qin proposed a non-dominated sorting based SAEA for multi-objective aerothermodynamic design [78]. Recently, Chugh et al. proposed a reference vector-guided surrogate-assisted evolutionary algorithm for solving expensive MOPs with more than three objectives [79,80], which was also applied to design the air intake ventilation system [81].

Another basic idea is to construct a single model of an aggregation function of the objectives. A typical algorithm is the hybrid algorithm with on-line landscape approximation (ParEGO) [82], where the Kriging model is adopted to surrogate the weighted sum fitness function. Similarly, the performance metrics can be used as the fitness function. In the \mathcal{S} -metric selection-based SAEA (SMS-EGO), the Kriging model is used to surrogate the \mathcal{S} metric. By contrary, there are also some SAEAs using the surrogate models for classification to learn the Pareto dominance relationship or the Pareto rankings [83]. In 2014, Bandaru et al. trained a multi-class surrogate classifier to determine the dominance relationship between two candidate solutions [84]. In 2015, Bhattacharjee and Ray proposed a support vector machine-based surrogate to learn the ranking of solutions for constrained multi-objective optimization problems [85]. Later in 2017, Zhang et al. trained a classifier based on a regression tree or a k-nearest-neighbour (KNN) to distinguish good solutions from bad ones [86]. Recently, Pan et al. proposed a classification-based surrogate-assisted evolutionary algorithm (CSEA) to learn the dominance relationship between a candidate solution and a set of reference solutions [87].

Discussion

Different from the other two types of MBEAs, SAEAs are proposed for solving the computationally expensive optimization problems. They are effective in obtaining a set of acceptable candidate solutions with limited computational resources. Nevertheless, there are still some challenges to be addressed.

First, the choice of surrogate models is not straightforward. There are many different types of surrogates but there is no simple rule for determining which type should be chosen.

It is crucial to balance the fitting ability and the computational efficiency of a surrogate model for different problems, i.e., a simple/powerful model should be used for a simple/complex problem.

Second, SAEAs also suffer from the curse of dimensionality. For example, the computation time for training a Kriging model with a high-dimensional input data (in terms of the dimensionality of a sample and the size of the samples) is unaffordable. It is necessary to use some dimensionality reduction methods or powerful surrogate models to deal with this issue.

Third, it is non-trivial to determine what should be predicted by the surrogate. This issue is more serious for expensive MOPs due to the existence of multiple objectives. It is interesting to design new features to distinguish the quality of two candidate solutions, as the advanced abstract feature may filter the local optima to improve the performance of SAEAs.

Fourth, the utilization of the “uncertainty” information should be further investigated. While existing SAEAs mainly use a single type of “uncertainty” information, their performance may degenerate sharply if the trained surrogate models have the syntropic prediction biases.

Summary

While the evolutionary algorithms (EAs) have witnessed a rapid development during the past two decades, the development of the model-based evolutionary algorithms (MBEAs) is attracting increasing interests. In contrast to providing a comprehensive review of every single method in the literature, this survey tries to shed light on the different motivations of using models in EAs: estimation of distribution, inverse modeling, and surrogate modeling. Among the three types of MBEAs, the estimation of distribution algorithms (EDAs) and the surrogate-assisted evolutionary algorithms (SAEAs) are more widely studied and applied, while the development of the inverse modeling-based EAs is still at the infancy.

From the machine-learning point of view, the working mechanism in MBEAs is twofold, where *data* and *models* are also key to the MBEAs, just as to the machine-learning algorithms. On one hand, the learning models are iteratively built and updated using the fitness values as training data. On the other hand, the models are iteratively sampled to generate the candidate solutions as the reproduction. Therefore, it is very important that the suitable models should be trained using the suitable candidate solutions, and there is still much to be studied along this direction.

To the best of our knowledge, this is the first survey of MBEAs in the literature. We hope that it not only helps better understand how models enable EAs to learn, but also

provides a systematic taxonomy of the related methods in this field.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Cheng R (2016) Nature inspired optimization of large problems. Ph.D. thesis, University of Surrey (United Kingdom)
- Wang H, Olhofer M, Jin Y (2017) A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges. *Complex Intell Syst* 3(4):233–245
- Kelley CT (2003) Solving nonlinear equations with Newton's method. Society for Industrial and Applied Mathematics, Philadelphia
- Tsamardinos I, Brown LE, Aliferis CF (2006) The max–min hill-climbing Bayesian network structure learning algorithm. *Mach Learn* 65(1):31–78
- Back T, Emmerich M, Shir O (2008) Evolutionary algorithms for real world applications (application notes). *IEEE Comput Intell Mag* 3(1):64–67
- Holland JH (1992) *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, Cambridge
- Fogel DB (2006) *Evolutionary computation: toward a new philosophy of machine intelligence*, vol 1. Wiley, USA
- Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. *IEEE Trans Evol Comput* 3(2):82–102
- Schwefel HPP (1993) *Evolution and optimum seeking: the sixth generation*. Wiley, USA
- Koza JR (1994) Genetic programming as a means for programming computers by natural selection. *Stat Comput* 4(2):87–112
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 11(4):341–359
- Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: *Proceedings of the sixth international symposium on micro machine and human science*. IEEE, pp 39–43
- Dorigo M, Maniezzo V, Colomi A (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern Part B (Cybernetics)* 26(1):29–41
- Svozil D, Kvasnicka V, Pospichal J (1997) Introduction to multi-layer feed-forward neural networks. *Chemometr Intell Lab Syst* 39(1):43–62
- Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY (2002) An efficient k-means clustering algorithm: analysis and implementation. *IEEE Trans Pattern Anal Mach Intell* 24(7):881–892
- Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297
- Hyvärinen A, Karhunen J, Oja E (2004) *Independent component analysis*, vol 46. Wiley, USA
- Larrañaga P, Lozano JA (2001) *Estimation of distribution algorithms: a new tool for evolutionary computation*, vol 2. Springer, Berlin
- Mühlenbein H, Paass G (1996) From recombination of genes to the estimation of distributions i. binary parameters. In: *International conference on parallel problem solving from nature*. Springer, Berlin, pp 178–187
- Baluja S (1994) *Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning*. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science
- De Bonet JS, Isbell Jr CL, Viola PA (1997) MIMIC: finding optima by estimating probability densities. In: *Proceedings of the 1997 conference on advances in neural information processing systems (NIPS'97)*. MIT Press, Cambridge, pp 424–431
- Pelikan M, Mühlenbein H (1999) The bivariate marginal distribution algorithm. In: *Advances in soft computing*. Springer, Berlin, pp 521–535
- Pelikan M, Goldberg DE, Cantu-Paz E (2000) Linkage problem, distribution estimation, and Bayesian networks. *Evol Comput* 8(3):311–340
- Heckerman D, Geiger D, Chickering DM (1995) Learning Bayesian networks: the combination of knowledge and statistical data. *Mach Learn* 20(3):197–243
- Etzeberria R (1999) Global optimization using Bayesian networks. In: *Proceeding of the 2nd symposium on artificial intelligence (CIMAFA-99)*, pp 332–339
- Pelikan M, Sastry K, Goldberg DE (2005) Multiobjective hBOA, clustering, and scalability. In: *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. ACM, pp 663–670
- Pelikan M (2005) Hierarchical Bayesian optimization algorithm. In: *Hierarchical Bayesian optimization algorithm*. Studies in fuzziness and soft computing. Springer, Berlin, pp 105–129
- Santana R, Larrañaga P, Lozano JA (2010) Learning factorizations in estimation of distribution algorithms using affinity propagation. *Evol Comput* 18(4):515–546
- Laumanns M, Ocenasek J (2002) Bayesian optimization algorithms for multi-objective optimization. In: *International conference on parallel problem solving from nature*. Springer, Berlin, pp 298–307
- Bosman PA, Thierens D (2006) Multi-objective optimization with the naive MIDEA. In: *Towards a new evolutionary computation*. Advances in estimation of distribution algorithms. Springer, Berlin, pp 123–157
- Ocenasek J, Kern S, Hansen N, Koumoutsakos P (2004) A mixed Bayesian optimization algorithm with variance adaptation. In: *International conference on parallel problem solving from nature*. Springer, Berlin, pp 352–361
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
- Zhang Q, Zhou A, Jin Y (2008) RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm. *IEEE Trans Evol Comput* 12(1):41–63
- Jin Y, Sendhoff B (2003) Connectedness, regularity and the success of local search in evolutionary multi-objective optimization. In: *The 2003 Congress on Evolutionary Computation*, vol 3. IEEE, pp 1910–1917
- Hansen N, Ostermeier A (1996) Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In: *Proceedings of IEEE international conference on evolutionary computation*. IEEE, pp 312–317
- Hansen N, Müller SD, Koumoutsakos P (2003) Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evol Comput* 11(1):1–18
- Igel C, Hansen N, Roth S (2007) Covariance matrix adaptation for multi-objective optimization. *Evol Comput* 15(1):1–28
- Giagkiozis I, Fleming PJ (2014) Pareto front estimation for decision making. *Evol Comput* 22(4):651–678
- Chen T, Chen H (1995) Approximation capability to functions of several variables, nonlinear functionals, and operators by radial

- basis function neural networks. *IEEE Trans Neural Networks* 6(4):904–910
40. Cheng R, Jin Y, Narukawa K, Sendhoff B (2015) A multiobjective evolutionary algorithm using gaussian process-based inverse modeling. *IEEE Trans Evol Comput* 19(6):838–856
 41. Corriveau G, Guilbault R, Tahan A, Sabourin R (2016) Bayesian network as an adaptive parameter setting approach for genetic algorithms. *Complex Intell Syst* 2(1):1–22
 42. Cheng R, Jin Y, Narukawa K (2015) Adaptive reference vector generation for inverse model based evolutionary multiobjective optimization with degenerate and disconnected pareto fronts. In: *International conference on evolutionary multi-criterion optimization*. Springer, Berlin, pp 127–140
 43. Gee SB, Tan KC, Alippi C (2017) Solving multiobjective optimization problems in unknown dynamic environments: an inverse modeling approach. *IEEE Trans Cybern* 47(12):4223–4234
 44. Hernández-Díaz AG, Santana-Quintero LV, Coello Coello CA, Molina J (2007) Pareto-adaptive ϵ -dominance. *Evol Comput* 15(4):493–517
 45. Tian Y, Wang H, Zhang X, Jin Y (2017) Effectiveness and efficiency of non-dominated sorting for evolutionary multi- and many-objective optimization. *Complex Intell Syst* 3:247–263
 46. Martínez SZ, Hernández VAS, Aguirre H, Tanaka K, Coello CAC (2014) Using a family of curves to approximate the Pareto front of a multi-objective optimization problem. In: *International conference on parallel problem solving from nature*. Springer, Berlin, pp 682–691
 47. Tian Y, Zhang X, Cheng R, He C, Jin Y (2018) Guiding evolutionary multi-objective optimization with robust front modeling. *IEEE Trans Cybern* (**accepted**)
 48. Wilson B, Cappelleri D, Simpson TW, Frecker M (2001) Efficient Pareto frontier exploration using surrogate approximations. *Optim Eng* 2(1):31–50
 49. Jin Y, Sendhoff B (2009) A systems approach to evolutionary multiobjective structural optimization and beyond. *IEEE Comput Intell Mag* 4(3):62–76
 50. Wang H, Jin Y, Doherty J (2017) A generic test suite for evolutionary multi-fidelity optimization. *IEEE Trans Evol Comput*. <https://doi.org/10.1109/TEVC.2017.2758360>
 51. Tesfahunegn YA, Koziel S, Leifsson LT (2015) Surrogate-based airfoil design with multi-level optimization and adjoint sensitivity. In: *53rd AIAA aerospace sciences meeting*. Kissimmee, Florida, p 0759
 52. Jin Y (2005) A comprehensive survey of fitness approximation in evolutionary computation. *Soft Comput* 9(1):3–12
 53. Jin Y (2011) Surrogate-assisted evolutionary computation: recent advances and future challenges. *Swarm Evol Comput* 1(2):61–70
 54. Tabatabaei M, Hakanen J, Hartikainen M, Miettinen K, Sindhya K (2015) A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods. *Struct Multidiscip Optim* 52(1):1–25
 55. Chugh T, Sindhya K, Hakanen J, Miettinen K (2017) A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Comput*. <https://doi.org/10.1007/s00500-017-2965-0>
 56. Zhan D, Cheng Y, Liu J (2017) Expected improvement matrix-based infill criteria for expensive multiobjective optimization. *IEEE Trans Evol Comput* 21(6):956–975
 57. Jin Y, Olhofer M, Sendhoff B (2002) A framework for evolutionary optimization with approximate fitness functions. *IEEE Trans Evol Comput* 6(5):481–494
 58. Jin Y, Olhofer M, Sendhoff B (2000) On evolutionary optimization with approximate fitness functions. In: *Genetic and evolutionary computation conference*, pp 786–793
 59. Lim D, Jin Y, Ong YS, Sendhoff B (2010) Generalizing surrogate-assisted evolutionary computation. *IEEE Trans Evol Comput* 14(3):329–355
 60. Nair P, Keane A, Shimpi R (1998) Combining approximation concepts with genetic algorithm-based structural optimization procedures. In: *39th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference and exhibit*, p 1912
 61. Sefrioui M, Périaux J (2000) A hierarchical genetic algorithm using multiple models for optimization. In: *International conference on parallel problem solving from nature*. Springer, Berlin, pp 879–888
 62. Branke J, Schmidt C (2005) Faster convergence by means of fitness estimation. *Soft Comput* 9(1):13–20
 63. Emmerich M, Giotis A, Özdemir M, Bäck T, Giannakoglou K (2002) Metamodel-assisted evolution strategies. In: *International conference on parallel problem solving from nature*. Springer, Berlin, pp 361–370
 64. Liu B, Zhang Q, Gielen GG (2014) A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Trans Evol Comput* 18(2):180–192
 65. Emmerich MT, Giannakoglou KC, Naujoks B (2006) Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Trans Evol Comput* 10(4):421–439
 66. Zhou Z, Ong YS, Nair PB, Keane AJ, Lum KY (2007) Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Trans Syst Man Cybern Part C (Appl Rev)* 37(1):66–76
 67. Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *J Global Optim* 13(4):455–492
 68. Guo D, Jin Y, Ding J, Chai T (2018) Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems. *IEEE Trans Cybern*. <https://doi.org/10.1109/TCYB.2018.2794503>
 69. Bernardino HS, Barbosa HJ, Fonseca LG (2010) A faster clonal selection algorithm for expensive optimization problems. In: *International conference on artificial immune systems*. Springer, Berlin, pp 130–143
 70. Handoko SD, Kwok CK, Ong YS (2010) Feasibility structure modeling: an effective chaperone for constrained memetic algorithms. *IEEE Trans Evol Comput* 14(5):740–758
 71. Handoko SD, Kwok C, Ong Y (2011) Classification-assisted memetic algorithms for equality-constrained optimization problems with restricted constraint function mapping. In: *2011 IEEE congress on evolutionary computation (CEC)*. IEEE, pp 391–400
 72. Sun C, Jin Y, Zeng J, Yu Y (2015) A two-layer surrogate-assisted particle swarm optimization algorithm. *Soft Comput* 19(6):1461–1475
 73. Wang H, Jin Y, Doherty J (2017) Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE Trans Cybern* 47(9):2664–2677
 74. Allmendinger R, Emmerich M, Hakanen J, Jin Y, Rigoni E (2017) Surrogate-assisted multicriteria optimization: complexities, prospective solutions, and business case. *J Multi-Criteria Decis Anal* 24(1–2):5–24
 75. Brownlee AE, Woodward JR, Swan J (2015) Metaheuristic design pattern: surrogate fitness functions. In: *Proceedings of the companion publication of the 2015 annual conference on genetic and evolutionary computation*. ACM, pp 1261–1264
 76. Allmendinger R, T M Emmerich M, Hakanen J, Jin YJ, Rigoni E (2017) Surrogate-assisted multicriteria optimization: complexities, prospective solutions, and business case. *J Multi-Crit Decis Anal* 24(1–2):5–24
 77. Singh HK, Ray T, Smith W (2010) Surrogate assisted simulated annealing (SASA) for constrained multi-objective optimization. In: *2010 IEEE congress on evolutionary computation (CEC)*. IEEE, pp 1–8

78. Ahmed M, Qin N (2012) Surrogate-based multi-objective aerodynamic design optimization of hypersonic spiked bodies. *AIAA J* 50(4):797–810
79. Chugh T, Jin Y, Miettinen K, Hakanen J, Sindhya K (2016) A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Trans Evol Comput* 22(1):129–142
80. Cheng R, Li M, Tian Y, Zhang X, Yang S, Jin Y, Yao X (2017) A benchmark test suite for evolutionary many-objective optimization. *Complex Intell Syst* 3(1):67–81
81. Chugh T, Sindhya K, Miettinen K, Jin Y, Kratky T, Makkonen P (2017) Surrogate-assisted evolutionary multiobjective shape optimization of an air intake ventilation system. In: 2017 IEEE congress on evolutionary computation (CEC). IEEE, pp 1541–1548
82. Knowles J (2006) ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Trans Evol Comput* 10(1):50–66
83. Seah CW, Ong YS, Tsang IW, Jiang S (2012) Pareto rank learning in multi-objective evolutionary algorithms. In: 2012 IEEE congress on evolutionary computation (CEC). IEEE, pp 1–8
84. Bandaru S, Ng AH, Deb K (2014) On the performance of classification algorithms for learning Pareto-dominance relations. In: 2014 IEEE congress on evolutionary computation (CEC). IEEE, pp 1139–1146
85. Bhattacharjee KS, Ray T (2015) A novel constraint handling strategy for expensive optimization problems. In: 11th world congress on structural and multidisciplinary optimization, pp 1–6
86. Zhang J, Zhou A, Tang K, Zhang G (2017) Preselection via classification: A case study on evolutionary multiobjective optimization. arXiv preprint [arXiv:1708.01146](https://arxiv.org/abs/1708.01146)
87. Pan L, He C, Tian Y, Wang H, Zhang X, Jin Y (2018) A classification based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. *IEEE Trans Evol Comput*. <https://doi.org/10.1109/TEVC.2018.2802784>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.