Swarm and Evolutionary Computation 1 (2011) 61–70

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Survey paper

Surrogate-assisted evolutionary computation: Recent advances and future challenges

Yaochu Jin

Department of Computing, University of Surrey, Guildford, Surrey, GU2 7XH, UK

ARTICLE INFO

Article history:
Received 6 February 2011
Received in revised form
20 April 2011
Accepted 14 May 2011
Available online 6 June 2011

Keywords:
Evolutionary computation
Surrogates
Meta-models
Machine learning
Expensive optimization problems
Model management

ABSTRACT

Surrogate-assisted, or meta-model based evolutionary computation uses efficient computational models, often known as surrogates or meta-models, for approximating the fitness function in evolutionary algorithms. Research on surrogate-assisted evolutionary computation began over a decade ago and has received considerably increasing interest in recent years. Very interestingly, surrogate-assisted evolutionary computation has found successful applications not only in solving computationally expensive single- or multi-objective optimization problems, but also in addressing dynamic optimization problems, constrained optimization problems and multi-modal optimization problems. This paper provides a concise overview of the history and recent developments in surrogate-assisted evolutionary computation and suggests a few future trends in this research area.

1. Introduction

In most evolutionary algorithms, it is often implicitly assumed that there exists a means for evaluating the fitness value of all individuals in a population. In general, the fitness value of an individual can be computed using an explicit fitness function, a computational simulation, or an experiment. In practice, however, fitness evaluations may become non-trivial. Such situations typically occur when evolutionary algorithms are employed to solve expensive optimization problems, where either the computational simulation for each fitness evaluation is highly time-consuming, or the experiments for fitness estimation are prohibitively costly, or an analytical function for fitness evaluations simply does not exist.

Surrogate-assisted evolutionary computation was mainly motivated from reducing computational time in evolutionary optimization of expensive problems, such as aerodynamic design optimization [1] or drug design [2], where complex computational simulations are involved.

In principle, surrogates should be used together with the real fitness function, as long as such a fitness function exists to prevent the evolutionary algorithm from being misled by a false minimum introduced by the surrogates [3]. A strategy for properly using the surrogates is often known as model management or evolution control. In surrogate-assisted evolutionary optimization of expensive problems, in particular when the problems are of high-dimension, the development of a model management strategy remains a challenging research topic.

The remainder of the paper is organized as follows. Section 2 takes a brief look back at the history of surrogate-assisted evolutionary computation starting from the late 1990s. Representative model management strategies are discussed in Section 3, which distinguish themselves into managing a single surrogate, homogeneous multiple surrogates, and heterogeneous multiple surrogates. Application of surrogates to addressing problems other than expensive optimization in evolutionary computation is presented in Section 4. Application examples of meta-model based evolutionary optimization are briefly accounted in Section 5. A few promising yet challenging research topics are suggested in Section 4. The paper concludes with a brief summary in Section 7.

2. A brief look back

Research on evolutionary optimization using approximate fitness evaluations was first reported in the mid-1980s [4], and sporadic yet increasing research results on evolutionary optimization using computational models for fitness estimation appeared after the mid-1990s [5–9]. The first event devoted to research on using surrogates in evolutionary optimization was a workshop held in 2002 within the Genetic and Evolutionary Computation Conference (GECCO) [10]. Since then, a series of special sessions and workshops have been organized on the major conferences including GECCO and IEEE Congress on Evolutionary

E-mail address: yaochu.jin@surrey.ac.uk.

2210-6502/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.swevo.2011.05.001

Computation, and journal special issues have also been edited. An overview of the research on surrogated-assisted evolutionary optimization reported in various fields was first presented in a conference paper [11], and then a journal paper in a special issue [12]. A first tutorial on fitness approximation on evolutionary optimization was given at the GECCO in 2005. Most recently, an edited book on the use of surrogates in evolutionary computation was also published [13].

In the review paper [12], the importance of managing surrogates was emphasized for the first time to prevent the evolutionary algorithms from being misled to a false optimum that can be introduced in a surrogate. In that review, methods for managing surrogates in evolutionary computation were divided into three categories, namely, individual-based, generation-based and population-based strategies. A variety of computational models, including polynomials (also known as response surface methodologies in the field of traditional design optimization), Gaussian processes (also known as Kriging in traditional design optimization), neural networks, together with data sampling techniques such as design of experiments, active learning and boosting were also presented. Practically, fitness inheritance from parents or fitness imitation from siblings [14–16] can be seen as a sort of simplified yet effective interpolation technique. General issues such as the global and local approximation, approximation of nonlinear constraints and the use of multiple surrogates having various fidelities were discussed. Theoretical analysis of the convergence properties was also raised.

Since the review paper [12], very encouraging research progresses have been made in many of the areas, whereas some issues remain unsolved, in particular with respect to a rigorous theoretical support for the benefit for using surrogates in evolutionary computation. Note that this paper focuses on surrogates in evolutionary computation. Readers interested in recent developments of surrogate-assisted design and analysis methods are referred to [17,18].

The next section provides a brief overview of recent advances in the research on surrogate-assisted evolutionary optimization, emphasizing on the progresses made after the review paper [12]. Research on using surrogates beyond solving expensive problems is discussed in Section 4. A few challenging topics for future research are suggested in Section 6. A summary of the paper is given in Section 7.

3. Strategies for managing surrogates

In most real-world optimization problems, no analytical fitness function exists for accurately evaluating the fitness of a candidate solution. Instead, there are only more accurate and less accurate fitness estimation methods, which often trade off accuracy with computational costs, as illustrated in Fig. 1. For example, in evolutionary optimization of aerodynamic structures [1], wind tunnel experiments may provide the most accurate estimation of the quality of candidate designs. The cost of such experiments is often prohibitively high. In addition, three-dimensional (3-D) computational fluid dynamic (CFD) simulations using Navier–Stokes equations may provide very accurate fitness evaluations. Unfortunately, such CFD simulations are highly time-consuming, which can take hours or even days for one single fitness evaluation. Computationally more efficient simulations can be achieved by 2-D full simulations or even incomplete simulations. By incomplete simulation, we mean that a simulation process is stopped before it converges. The computationally most efficient way for estimating fitness is the use of machine learning models, i.e., surrogates. Note, however that this graphic only shows a simplified version of actual levels of accuracy.

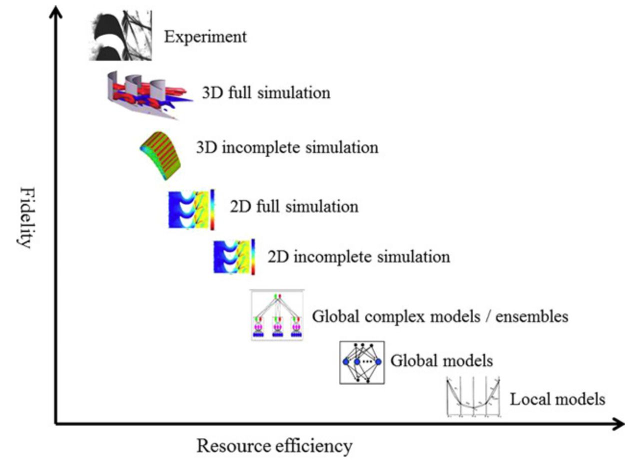

Fig. 1. An illustration of a trade-off between fidelity (approximation accuracy) and computational cost. Usually, high-fidelity fitness evaluations are more time-consuming. By contrast, low-fidelity fitness evaluations are often less time-consuming.

In the research of surrogate-assisted evolutionary optimization, most algorithms have been developed based on benchmark problems, where it is assumed that fully accurate fitness evaluations can be provided. Such fitness functions are often termed "real fitness function" or "original fitness function". In the following, we use surrogates for denoting computational models constructed with data, whereas other approximate fitness techniques such as full or incomplete 2-D CFD simulations are called problem approximations as termed in [12]. In addition, we do not distinguish between surrogate-assisted single objective optimization and surrogate-assisted multi-objective optimization if the method for model management does not differ.

In the early work on surrogate-assisted evolutionary optimization, the evolutionary search is based solely on a surrogate, assuming that the surrogate can provide sufficiently accurate fitness evaluations. However, such assumptions can give rise to serious problems if the surrogate introduces optima that do not exist in the original optimization problem. This issue was first explicitly raised in [3] to stress the importance of model management in surrogate-assisted evolutionary optimization, mainly by using the surrogate together with the real fitness function.

Surrogates can be applied to almost all operations of evolutionary algorithms, such as population initialization, cross-over, mutation, local search and fitness evaluations, as illustrated in Fig. 2. For instance, a surrogate can be used for filtering out poor solutions in population initialization, crossover [19] or mutation [20]. The use of surrogates in initialization, mutation or crossover [21] can reduce the randomness in the genetic operators, thus termed informed operators. Most recently, a similar approach is adopted for multi-objective optimization [22], where a single, aggregated meta-model is built to pre-screen candidate solutions before fitness evaluation. The requirement on the quality of surrogates is minimum, as an estimated fitness that is better than a random guess is adequate.

Techniques for managing surrogates for fitness evaluations can generally be divided into individual-based, generation-based and population-based [12]. By generation-based, we mean that surrogates are used for fitness evaluations in some of the generations, while in the rest of the generations, the real fitness function is used [8,23,24,7]. By contrast, in individual-based model management techniques, the real-fitness function is used for fitness evaluations for some of the individuals in a generation [25,3,23]. In population-based approaches, more than one sub-population co-evolves, each using its own surrogate for fitness

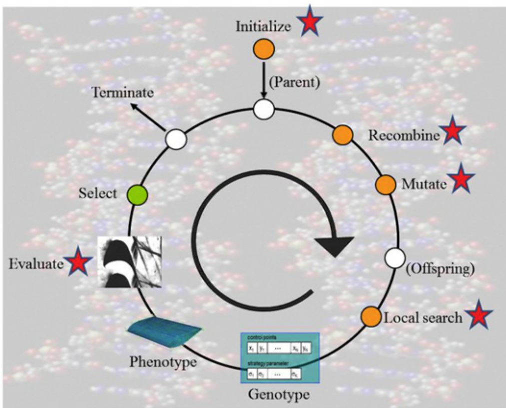

Fig. 2. A diagram for an evolutionary algorithm for optimization of a turbine blade. A star denotes an evolutionary operation where a surrogate can be helpful.

evaluations. Migration of individuals from one sub-population to another is allowed.

A strategy closely related to the above methods is the pre-selection strategy [26]. Pre-selection does not exactly fall in individual-based strategies. Assume the population size is λ. In pre-selection, an initial offspring population contains λ' individuals, where $\lambda' > \lambda$ are produced in each generation. Then, all λ' offspring individuals are evaluated using the surrogate. Based on the fitness value obtained using the surrogate, only λ offspring are kept and re-evaluated using the original fitness function. A main difference between individual-based strategies and pre-selection here is that in pre-selection, selection is always based on the real fitness value whereas in individual-based methods, selection may be conducted partly based on fitness values from the surrogate.

The main steps of one specific individual-based model management method, termed best strategy, and the pre-selection method for a (μ, λ) evolution strategy, (μ, λ)-ES, are illustrated in Fig. 3 (a) and (b), respectively. In a (μ, λ)-ES using best strategy, all λ offspring are first evaluated using the surrogate. Then, $\lambda^\star \leq \lambda$ best individuals according to the surrogate are re-evaluated using the expensive real fitness function. As a result, it can happen that the fitness value of some of the selected μ parents is based on the surrogate. Contrary to that, in a (μ, λ)-ES using pre-selection, $\lambda^\star \geq \lambda$ offspring are generated and then evaluated using the surrogate. Then, λ best individuals are re-evaluated using the expensive real fitness function. Consequently, all the selected μ parents for the next generation are evaluated by the real fitness function.

A large category of surrogate-assisted evolutionary algorithms use surrogates in local search only for both single [27,28] and multi-objective optimization [29]. In this case, sophisticated model management methods developed in traditional design optimization [30], such as the trust-region method [31] can be directly employed.

Recently, surrogates have also been used in stochastic search methods other than evolutionary algorithms, such as surrogate-assisted simulated annealing [32] or surrogate-assisted artificial immune systems [33].

In the following, we discuss a few interesting ideas for model management, which are divided into two major categories — use of a single surrogate and multiple surrogates.

3.1. Managing a single surrogate

The essential question to answer in surrogate-assisted evolutionary computation is which individuals should be chosen to be evaluated or re-evaluated using the real fitness function. As we

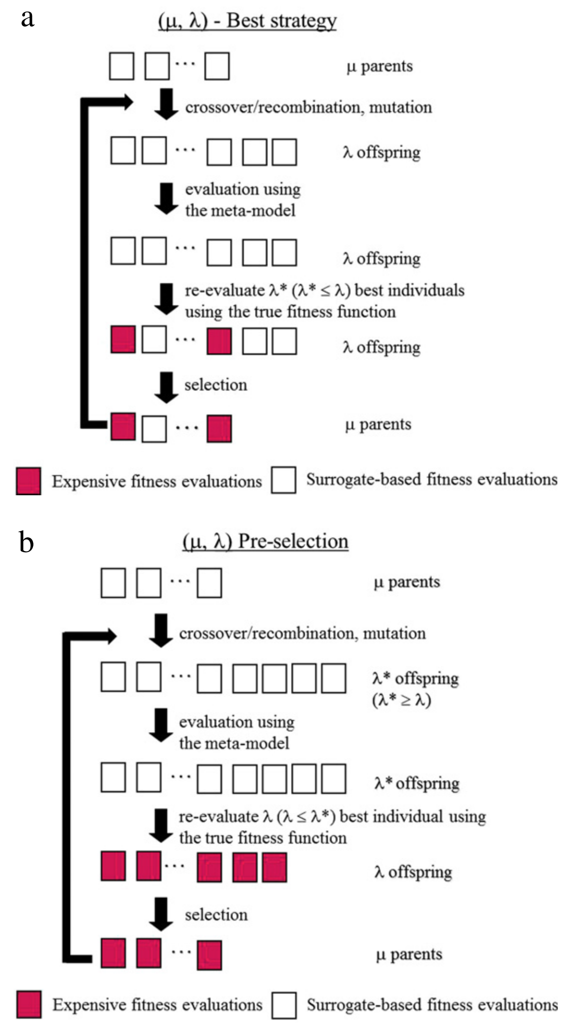

Fig. 3. Two individual-based model management strategies. (a) Best strategy, and (b) Pre-selection strategy.

assume fitness evaluations using the real fitness function is time-consuming, the next question is how to adapt the number of individuals to be evaluated with the real fitness function so that the time for fitness evaluations can be reduced as much as possible, while the evolutionary algorithm can still find the global optimum. In the following, we discuss a few issues related to the answer to the above questions.

3.1.1. Criteria for choosing individuals for re-evaluation

The most straightforward idea is to evaluate those individuals that potentially have a good fitness value and the higher the approximation accuracy, the more often the surrogate can be used [3,23]. A slightly different idea is that representative individuals can be chosen for re-evaluation by clustering the population into a number of crisp or fuzzy clusters. The individual closest to each cluster center [34,15,35] or the best individual in each cluster [36,37] can be chosen for re-evaluation.

It has also been suggested that individuals having a large degree of uncertainty in approximation can be good candidates for re-evaluation [25,26]. This idea can be justified by two arguments. First, a large degree of uncertainty in approximation of the individuals indicates that the fitness landscape around these solutions has not been well explored and therefore may provide a good chance of finding a better solution. Second, re-evaluation of these solutions may be the most effective in improving the

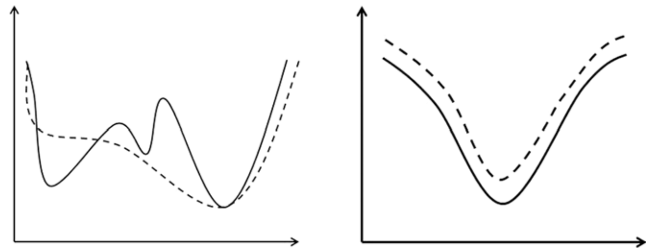

Fig. 4. Examples of surrogates that have a large approximation error but are adequately good for evolutionary search. Solid curves denote the original function and dashed curves are their approximation.

approximation accuracy of the surrogate, similar to the idea of active learning [38].

The estimation of the approximation error can be achieved with different methods. In [25], the degree of uncertainty is roughly set to be inversely proportional to the average distance to the closest data samples used for constructing the surrogate. Alternatively, an ensemble can be used for estimating the variance of the individual estimates given by an ensemble of surrogates. The most often used surrogate model for estimating model uncertainties is the Gaussian processes [39], also known as the Kriging model [40]. Unlike deterministic models, Gaussian processes provide an estimate of the fitness (mean) together with an estimate of the uncertainty (variance), which is a statistically sound boundary of the uncertainty in fitness estimation. Due to this property, Gaussian processes have increasingly been employed as surrogates in evolutionary single- and multi-objective optimization [41–44]. Note however, the computational cost for constructing Gaussian processes itself can be very high when the number of samples used is large and online learning of the Gaussian processes is non-trivial when new samples are available.

3.1.2. Metrics for evaluating surrogates and adaptation

Not much attention has been paid to adapting the frequency of using the surrogates. In [45], the model quality is estimated by calculating the average approximation error after re-evaluation, which is used to adapt the frequency of using the surrogate in a generation-based model management method. Based on empirical observations that large approximation errors must not mislead the evolutionary search, see e.g., Fig. 4, a few metrics other than approximation error have been proposed in [46,45] in evaluating the quality of surrogates. In the following, we present a few performance measures for surrogates in great detail.

The most common measure for model quality or model fidelity is the mean squared error between the individual's real fitness value and the predicted fitness by the meta-model. However, from the evolutionary perspective, selecting the right individuals for the next generation is the main concern. For instance in Fig. 4, the quality of the surrogate is poor in terms of approximation accuracy. However, an evolutionary algorithm searching on the surrogate only will nevertheless find the right optimum. Consider (μ, λ)-selection with $\lambda \geq 2\mu$, which is of particular relevance in evolutionary optimization of complex real-world problems, the number of individuals that have been selected correctly using the surrogate can be obtained by:

$$\rho^{(\mathrm{sel.})} = \frac{\xi - \langle \xi \rangle}{\mu - \langle \xi \rangle}, \tag{1}$$

where ξ $(0 \leq \xi \leq \mu)$ is the number of correctly selected individuals, i.e., the number of individuals that would have also been selected if the real fitness function was used for fitness evaluations. The expectation

$$\langle \xi \rangle = \sum_{m=0}^{\mu} m \frac{\binom{\mu}{m}\binom{\lambda-\mu}{\mu-m}}{\binom{\lambda}{\mu}} = \frac{\mu^2}{\lambda} \tag{2}$$

of ξ in case random selection is used as a normalization in (1). It can be seen that if all μ parent individuals are selected correctly, the measure reaches its maximum of $\rho^{(\mathrm{sel.})} = 1$, and that negative values indicate that the selection based on the surrogate is worse than a random selection.

The measure $\rho^{(\mathrm{sel.})}$ only evaluates the absolute number of correctly selected individuals. If $\rho^{(\mathrm{sel.})} < 1$, the measure does not indicate whether the $(\mu + 1)$-th or the worst offspring individual has been selected, which may have significant influence on the evolutionary process. Therefore, the measure $\rho^{(\mathrm{sel.})}$ can be extended to include the rank of the selected individuals, calculated based on the real fitness function. A surrogate is assumed to be good, if the rank of the selected individuals based on the model is above-average according to the rank based on the real fitness function.

The definition of the extended measure $\rho^{(\sim\mathrm{sel.})}$ is as follows: The surrogate achieves a grade of $\lambda - m$, if the m-th best individual based on the real fitness function is selected. Thus, the quality of the surrogate can be indicated by summing up the grades of the selected individuals, which is denoted by π. It is obvious that π reaches its maximum, if all μ individuals are selected correctly:

$$\pi^{(\mathrm{max.})} = \sum_{m=1}^{\mu} (\lambda - m) = \mu\left(\lambda - \frac{\mu+1}{2}\right). \tag{3}$$

Similar to (1) the measure $\rho^{(\sim\mathrm{sel.})}$ is defined by transforming π linearly, using the maximum $\pi^{(\mathrm{max.})}$ as well as the expectation $\langle \pi \rangle = \frac{\mu\lambda}{2}$ for the case of a purely random selection:

$$\rho^{(\sim\mathrm{sel.})} = \frac{\pi - \langle \pi \rangle}{\pi^{(\mathrm{max.})} - \langle \pi \rangle}. \tag{4}$$

Besides these two problem-dependent measures for evaluating the quality of the surrogate, two established measures – the rank correlation and the (continuous) correlation – partially fit the requirements formulated above. The rank correlation can be expressed by

$$\rho^{(\mathrm{rank})} = 1 - \frac{6 \sum_{l=0}^{\lambda} d_l^2}{\lambda(\lambda^2 - 1)}, \tag{5}$$

is a measure for the monotonic relation between the ranks of two variables. In our case, d_l is the difference between the ranks of the l-th offspring individual based on the original fitness function and on the approximate model. The range of $\rho^{(\mathrm{rank})}$ is the interval $[-1; 1]$. The higher the value of $\rho^{(\mathrm{rank})}$, the stronger the monotonic relation with a positive slope between the ranks of the two variables. In contrast to $\rho^{(\sim\mathrm{sel.})}$, the rank correlation does not only take the ranking of the selected individuals, but also the ranks of all individuals into account.

A slightly different quality measure can be defined by calculating the (continuous) correlation between the surrogate and the original fitness function.

Using the selection-based criterion [45] for evaluating surrogates, an adaptation scheme has been suggested for adapting the

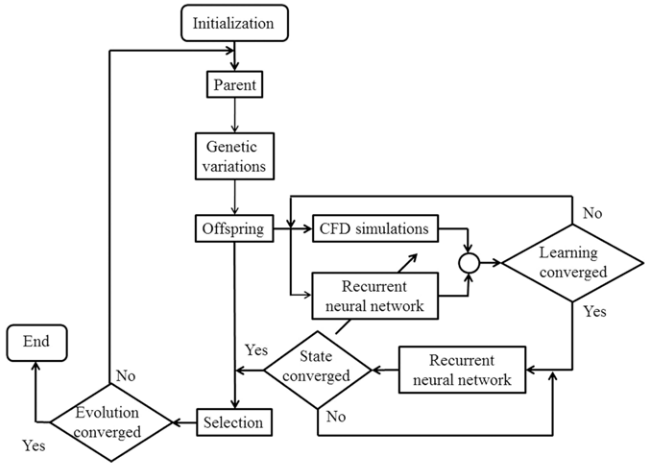

Fig. 5. An illustration of learning iterative fitness evolutionary process using recurrent neural networks for predicting converged fitness value.

number of individuals to be evaluated using the surrogate (λ') [47]. It has been shown that λ' increases as the evolution proceeds, indicating that the quality of the surrogates improves. Interestingly, when noise is introduced into the fitness data samples, λ' first decreases and then increases again. The various selection based criteria suggested in [45] have been benchmarked for adapting the number of individuals to be re-evaluated by the real fitness function [37]. The results, however, failed to show a clear advantage of any particular criterion.

3.1.3. Improving approximation accuracy

Although approximation quality is not the only criterion for surrogates for fitness prediction in evolutionary optimization, improving its approximation quality is desirable. Much work has been reported along this line. For example, in [3,23], regularization of the neural network model has been suggested to alleviate overfitting. Structure and parameter optimization of the surrogate can co-evolve with the original optimization problem [46,48].

One of the main difficulties in improving the approximation accuracy can be attributed to the high-dimensionality in the design space. To overcome this difficulty, the surrogate can be built up in a new space of a lower dimension using dimension reduction techniques [49,50].

It is noticed that in many expensive optimization problems, the fitness evaluation often consists of an iterative computation process, such as the numerical solution of differential equations in computational fluid dynamics simulations. In such cases, many intermediate data will be produced before the simulation converges. Such intermediate data can also be used for training a surrogate in the first iterations and then the surrogate can be used for predicting the converged fitness [51]. An example of such a process is illustrated in Fig. 5.

3.2. Managing multiple surrogates

Methods for multiple surrogates in evolutionary optimization distinguish themselves in type and fidelity of the surrogates. For example, a neural network ensemble has been used in [34], where all ensemble members are of the same type of feed-forward neural networks. Alternatively, multiple surrogates of different types, such as polynomials, support vector machines and neural networks can be used [52].

A step further is to use surrogates of different fidelities. Surrogates of different fidelities can be obtained by using models of different complexities or different data sets. For instance, different types of training samples used for constructing the surrogates can be generated from different problem approximations, such as wind-tunnel experiments, 3-D or 2-D CFD simulations. Another way of generating surrogates of heterogeneous fidelity is to control the complexity of the surrogates explicitly, e.g., by using a different number of training samples or by controlling the model complexity with regularized learning [3] or a Pareto-based multi-objective learning method [53].

In the following, we discuss the use of multiple surrogates in evolutionary optimization by dividing the methods into homogeneous and heterogeneous multiple surrogates. By homogeneous multiple surrogates, the fidelity of the surrogates are not explicitly controlled, even if different types of surrogates are used. On the contrary, heterogeneous surrogates vary in their fidelity due to an explicit control in model complexity or training data.

3.2.1. Homogeneous multiple surrogates

Use of ensembles for fitness approximation was suggested in [34], where it was shown that neural network ensembles can improve the performance of surrogate-assisted evolutionary optimization in two aspects. First, ensembles can improve the quality in fitness prediction. Second, the variance of the predicted fitness of the ensemble members can help identify large prediction errors so that false optima can be avoided.

The benefit of using multiple surrogates has also been shown empirically in many papers [54,55,52]. In this category of research, no explicit control of fidelity of the multiple surrogates is employed. For example in [56,57] multiple surrogates such as Kriging, polynomials, radial-basis-function networks (RBFN), and weighted average ensemble are used to demonstrate the improved robustness of optimization. Polynomial and RBFN surrogates are employed for multiobjective optimization and it was shown that each of the models performs better in different regions of the Pareto front.

Multiple surrogates have been used in evolutionary multi-objective optimization [58]. In that work, a co-evolutionary genetic algorithm for multiple-objective optimization based on surrogates was introduced. After some fixed search intervals, the surrogates that approximate different objectives are exchanged and shared among multiple sub-populations of genetic algorithms. Spatially distributed multiple surrogates have been used for fitness approximation in multi-objective optimization [59].

3.2.2. Heterogeneous multiple surrogates

As illustrated in Fig. 1, in many real-world optimization problems, various problem approximation techniques can be employed. For example, in aerodynamic optimization 3-D or 2-D numerical simulations can be used for estimating the quality of the designs in addition to wind tunnel experiments. In more extreme situations, incomplete simulations can also be used, where a numerical simulation is stopped earlier to reduce computation time. Data from all these different processes can also be applied for building up surrogates.

The motivation of explicitly controlling the fidelity of the surrogates can also be justified by taking the computational costs of constructing surrogates into account. To reduce the cost for building up surrogates, it makes good sense to use surrogates of a lower fidelity that can be obtained with less cost in the early stage of evolutionary optimization. Another more tricky motivation is to take advantage of approximation errors introduced by surrogates, hoping to smoothen a rugged fitness landscape, or to increase the diversity of the population, or simply to use data from incomplete simulations.

Early work that uses heterogeneous multiple surrogates was reported in [60,61], where a population-based model management strategy is used. In both papers, three sub-populations are used, each using a surrogate for fitness evaluations. In [60], individuals from a sub-population that use a surrogate of lower fidelity are allowed to migrate to the sub-population that uses a surrogate of higher fidelity. The method presented in [61] is a minor variant of [60], where migration is allowed between all sub-populations.

One approach to reducing the computational cost for constructing surrogates is to use coarse surrogates (of lower fidelity) in the early stage of the optimization and increase the quality of the surrogate gradually as the search proceeds [6]. This idea of using coarse-to-fine surrogates has been introduced into a surrogate-assisted evolutionary search in [42,62], where surrogates are used for evolutionary multi-objective optimization.

A more subtle way to control the fidelity of surrogates is to use surrogates of a sufficiently good fidelity based on a correlation based measure [63]. The fidelity control strategy was applied to a memetic algorithm in which the local search is based on surrogates of a changing fidelity. The proposed method was evaluated empirically on an aerodynamic airfoil design problem and demonstrated that the use of a dynamic fidelity is able to improve the search speed.

The idea of taking advantage of approximation errors introduced by surrogates was further exploited in [64]. In that work, two types of surrogates are used in the local search of an evolutionary multi-objective optimization: One for getting a reliable local prediction and the other for a higher degree of diversity. Empirical results show that an evolutionary search based on heterogeneous multiple models can considerably improve the search performance, compared to surrogate-assisted evolutionary algorithms that use a single surrogate or homogeneous multiple surrogates. Interestingly enough, the proposed algorithm also outperforms its counterpart that uses an artificial perfect surrogate. Detailed analysis of the search processes confirmed the hypothesis that controlled approximation errors introduced by surrogates can speed up the search process in both single- and multi-objective optimization.

3.3. *Which model management strategy?*

As we discussed above, surrogates can be used in population initialization, crossover, mutation and preselection to pre-screen candidate solutions. The advantage of these relatively conservative approaches to using surrogates is that they are less likely to mislead the search process. One concern might be that they may cause premature convergence. It is also less risky if a surrogate is used in a local search of memetic algorithms. The common feature of these approaches is that all individuals have been re-evaluated using the original fitness function before selection.

In addition, among the model management strategies, the individual-based model management may be more suited for steady state evolution, or generational evolution implemented on a single machine. By contrast, population-based and generation-based model management is better for parallel implementation on heterogeneous machines having different speeds. An optimization strategy may be desirable when multi-level surrogates having different computational complexities are used on machines having different computational powers.

4. Beyond evolutionary optimization of expensive problems

In addition to reducing the computation time in evolutionary optimization of expensive problems, surrogates can be useful in addressing other problems in evolutionary computation, such as the use of surrogates for reducing fitness evaluations in search of robust optimal solutions [65]. In addition, surrogates have been found helpful in improving the efficiency of evolutionary algorithms for solving optimization with noisy fitness evaluations [66] or for solving multi-modal optimization with a very rugged fitness landscape [6,67], where the purpose of using a surrogate is to smoothen the fitness landscape.

4.1. *Surrogates in interactive evolutionary computation*

In interactive evolutionary computation, the fitness value of each individual is evaluated by human user subjectively [68]. Human fitness evaluations are necessary where no fitness function is available. For instance, when evolutionary algorithms are used for aesthetic product design or art design. One main challenge of interactive evolutionary computation is the issue of human fatigue. To address this problem to a certain degree, surrogates can be used to replace in part human evaluations. The main idea is to use a machine learning model to predict the fitness value the human may assign to a design based on history data [69–71].

4.2. *Surrogated-assisted evolution for solving dynamic optimization*

Evolutionary optimization of dynamic optimization problems has become a popular research topic recently [12]. The primary goal is to develop an evolutionary search strategy that can follow a moving optimum or a moving Pareto front. To this end, a certain degree of diversity in the population should be maintained or a memory mechanism must be embedded in the evolutionary algorithm. Memory mechanisms include sub-populations, archives of optimal solutions found so far, or multiploidy in genetic representation.

In addition to memory and diversity based strategies, anticipation and prediction of the change in the fitness function can be helpful in solving dynamic problems more efficiently. In such strategies, a surrogate can be helpful in learning the changing fitness function [72–74].

4.3. *Surrogates for robust optimization*

In evolutionary optimization of real-world problems, one is concerned not only with the performance of the obtained optimal solution, but also the sensitivity of the performance to small changes in the design variables or in the environment. If an optimal solution is insensitive to such changes, the solution is known as robust optimization.

To obtain robust optimal solutions using evolutionary algorithms, either implicit averaging or explicit averaging can be used [12], wherein an assumption on the probability distribution of the noise is often made. By contrast, one can predefine the allowed performance decrease and then search for an optimum that has the maximum tolerance of changes in the design variables, which is termed inverse robust optimization [75]. In both explicit averaging based or inverse robust optimization, additional fitness evaluations are needed. To enhance the efficiency, some of these additional fitness evaluations can be done based on a surrogate [76–78].

4.4. *Surrogates for constrained optimization*

Many optimization problems are subject to constraints. To judge if a candidate solution is feasible, the constraint functions need to be frequently evaluated. Therefore, if the evaluations of constraint functions are time-consuming, it is desirable to replace the constraint functions with computationally efficient approximate models [79].

In some real-world applications, an explicit constraint is not available. For example in aerodynamic optimization, some of the candidate designs may result in unstable computational fluid dynamic (CFD) simulations. In order to reduce the number of unnecessary, time-consuming CFD simulations, it is very helpful to judge whether a solution is feasible (e.g., converges in a CFD simulation) before it is evaluated in a CFD simulation. Surrogates can be used for this purpose [80,81].

An interesting idea of using surrogates in constrained optimization has been recently reported in [82], where surrogates are

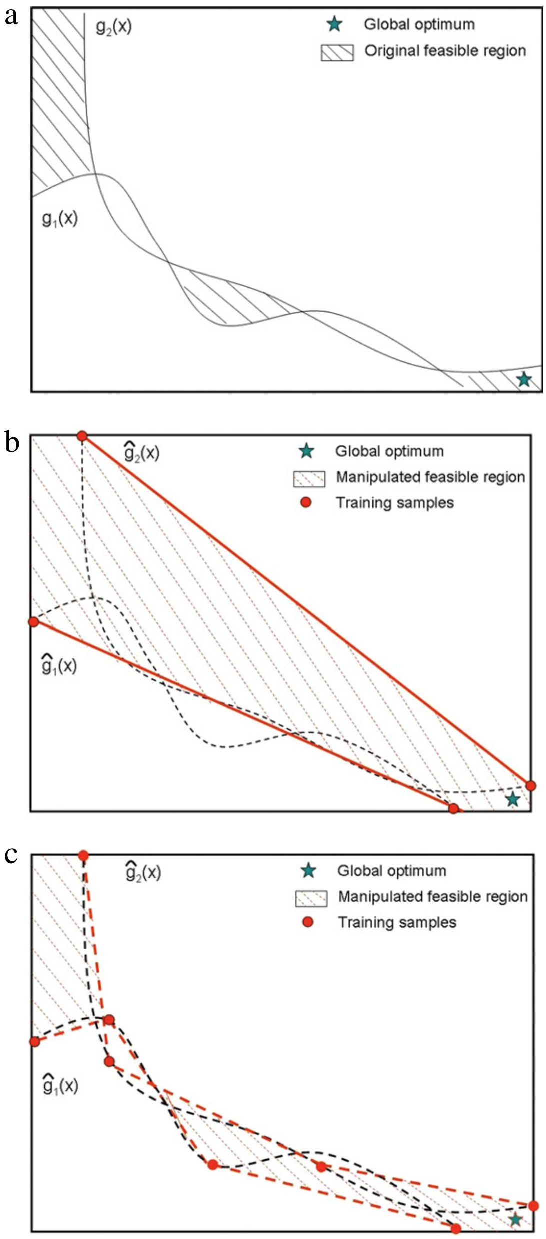

Fig. 6. An illustrative example of manipulating the constraints to facilitate evolutionary search [82]. (a) The true feasible region (shaded), which consists of three separate sub-regions. (b) A linear approximation of the original constraints by using two data points, resulting in an enlarged single feasible region. (c) A more accurate approximation of the constraint functions and the resulting feasible region is close to real one.

applied to manipulate the shape and size of the feasible region to ease the solution of highly constrained optimization problems. The basic idea is to deliberately enlarge the feasible region by building up a very simple surrogate for each constraint function. As the evolutionary optimization proceeds, the complexity of the surrogates increases gradually so that the approximated feasible region can converge to the real feasible region. An illustration of this basic idea is given in Fig. 6. Simulation results on a set of benchmark problems and a few structural design problems demonstrated that the idea works well. Genetic programming based generation of increasingly complex surrogates has also been reported [83].

5. Real-world applications

Surrogate-assisted evolutionary optimization is more application driven. Thus, the effectiveness of surrogate-assisted evolutionary algorithms need to be demonstrated in real-world applications. One intensively researched area is surrogate-assisted design optimization, such as turbine blades [9,23,84,85], airfoils [27,86], forging [87], vehicle crash tests [88], multi-processor systems-on-chip design [89] and injection systems [90]. Other applications include drug design [2], protein design [5], hydroinformatics [91] and evolutionary robotics [92]. We must note that not many substantial successful applications of meta-model based evolutionary optimization have been reported, which, however does not necessarily mean no such work has been done. Some work carried out in industry has not been published. We also want to note that meta-model based evolutionary optimization has been included in a few commercial design software tools.

6. Future challenges

Surrogate-assisted evolutionary computation has achieved considerable advances over the past decade, not only in algorithm design, but also in real-world applications. Nevertheless, many challenges remain to be addressed. In the following, we discuss a few of these challenges and hope that these discussions will trigger more research efforts devoted to approaching these challenges.

6.1. Theoretic work

A wide range of trust-region methods have shown to converge to the global optimum [93] when a gradient-based method is used to search on the surrogate. Unfortunately, a convergence proof for surrogate-assisted evolutionary algorithms to the global optimum or to a local optimum is not straightforward, as a proof of any stochastic search algorithm to a global optimum is non-trivial. Meanwhile, approximation errors introduced by surrogates can usually neither be described by a Gaussian nor a uniform distribution, which makes a quantitative analysis of the search dynamics on a surrogate very difficult, if not impossible.

If we go one step back, we may raise the question of whether we can guarantee that a surrogate-assisted evolutionary algorithm converges faster than its counterpart without using a surrogate using the same number of expensive fitness evaluations. Again, no theoretical work has been reported to show this.

6.2. Multi-level, multi-fidelity heterogeneous surrogates

Use of multi-level, multi-fidelity surrogates has already been suggested in [12]. The heterogeneity can include the model type of the surrogates and the degree of fidelity (modeling accuracy). On the one hand, various surrogates, ranging from the deterministic linear model (e.g., linear interpolation) to nonlinear models (e.g. feedforward neural networks, support vector machines) and to stochastic models, such as Gaussian processes (Kriging) and to dynamic models such as recurrent neural networks. Meanwhile, multi-fidelity models can be used either by using data from different problem approximations (e.g., 2D Navier–Stokes simulations and 3D Navier–Stokes simulations) or experiments, or different degrees of incomplete simulations, or by deliberately controlling the complexity of the models.

When heterogeneous surrogates are used, the computational times for fitness evaluations using different models can be very different. To further improve the computational efficiency of the whole evolutionary process, non-generational evolutionary algorithms with grid-based or asynchronous computing structure may be preferred [86,56].

6.3. Surrogate-assisted combinatorial optimization

Surrogate-assisted evolutionary algorithms have been studied extensively for continuous optimization. In real-world applications, however, there are also many computationally intensive combinatorial optimization problems, such as job shop scheduling and wireless network or mobile sensor network optimization. In such cases, discrete modeling techniques must be employed, e.g., binary neural networks [94]. In [95], an RNF neural network is applied to assist a mixed integer evolution strategy for intravascular ultrasound image analysis [95]. Recently, an integrated Kriging model is used for mobile network optimization [96].

6.4. Surrogate-assisted dynamic optimization

If an expensive optimization is time-varying, evolutionary algorithms for solving dynamic optimization problems must be adopted to track the moving optima or moving Pareto front [97]. Practically, an optimal solution that is robust over time may be more preferred [74]. In either case, the surrogate must be updated online. Therefore, it may be of interest to introduce incremental learning techniques [98] for efficient online learning when the objective functions change over time.

6.5. Rigorous benchmarking and test problems

Although many surrogate-assisted evolutionary algorithms have been proposed and demonstrated to be more efficient than their counterpart without using a surrogate, no rigorous comparative studies on surrogate-assisted evolutionary algorithms have been reported. This may be attributed to two reasons. First, no widely accepted performance index for benchmarking surrogate-assisted evolutionary algorithms has been suggested. Second, no benchmark problems dedicated to surrogate-assisted evolutionary algorithms have been proposed. Most work on surrogate-assisted evolutionary algorithms uses either standard test functions such as the Ackley function [99] or specific real-world applications for empirical evaluations. However, design of test problems relevant to real-world applications is non-trivial. Ideally, such test problems should reflect the major difficulties in real-world applications yet tractable for intensive empirical comparisons. As indicated in [1], expensive optimization problems such as aerodynamic design optimization not only involve highly time-consuming fitness evaluations, the fitness landscape is often multi-modal as well. In addition, the CFD simulations may be unstable, resulting in many isolated infeasible solutions. Finally, the design space is very high and geometry representation may be critical for the efficiency of the whole evolutionary design optimization.

7. Summary

Surrogate-assisted evolutionary algorithms are motivated from real-world applications. As evolutionary algorithms are increasingly applied to solving complex problems, research interests in surrogate-assisted evolutionary algorithms have considerably increased in recent years. This paper provides a brief overview of recent advances in this research area and suggests a few challenging issues that remain to be resolved in the future. We expect that successful resolution of these challenges heavily depends on the progress in both optimization and learning, and new computing techniques such as grid computing [100] and cloud computing [101], with which more computing resources will be made available to common users via computer networks.

References

[1] Y. Jin, B. Sendhoff, A systems approach to evolutionary multi-objective structural optimization and beyond, IEEE Computational Intelligence Magazine 4 (3) (2009) 62–76.
[2] D. Douguet, e-LEA3D: A computational-aided drug design web server, Nucleic Acids Research 38 (2010) w615–w621.
[3] Y. Jin, M. Olhofer, B. Sendhoff, On evolutionary optimization with approximate fitness functions, in: Genetic and Evolutionary Computation Congress, 2000, pp. 786–793.
[4] J.J. Grefenstette, J.M. Fitzpatrick, Genetic search with approximate fitness evaluations, in: Proceedings of the International Conference on Genetic Algorithms and Their Applications, 1985, pp. 112–120.
[5] G. Schneider, J. Schuchhardt, P. Wrede, Artificial neural networks and simulated molecular evolution are potential tools for sequence-oriented protein design, CABIOS 10 (6) (1994) 635–645.
[6] D. Yang, S.J. Flockton, Evolutionary algorithms with a coarse-to-fine function smoothing, in: IEEE International Conference on Evolutionary Computation, 1995, pp. 657–662.
[7] A. Ratle, Accelerating the convergence of evolutionary algorithms by fitness landscape approximation, in: Parallel Problem Solving from Nature, 1998, pp. 87–96.
[8] L. Bull, On model-based evolutionary computation, Soft Computing 3 (1999) 76–82.
[9] S. Pierret, Turbomachinery blade design using a Navier–Stokes solver and artificial neural network, ASME Journal of Turbomachinery 121 (3) (1999) 326–332.
[10] Y. Jin, S.J. Louis, K.M. Rasheed, Approximation and learning in evolutionary computation. GECCO Workshop, July 2002.
[11] Y. Jin, B. Sendhoff, Fitness approximation in evolutionary computation — a survey, in: Genetic and Evolutionary Computation Conference, 2002, pp. 1105–1112.
[12] Y. Jin, A comprehensive survey of fitness approximation in evolutionary computation, Soft Computing 9 (1) (2005) 3–12.
[13] Y. Tenne, C.-K. Goh (Eds.), Computational Intelligence in Expensive Optimization Problems, Springer, 2009.
[14] M.M. Davarynejad, C.W. Ahn, J. Vranckena, J. van den Berga, C.A. Coello Coello, Evolutionary hidden information detection by granulation-based fitness approximation, Applied Soft Computing 10 (3) (2010) 719–729.
[15] H.S. Kim, S.B. Cho, An efficient genetic algorithms with less fitness evaluation by clustering, in: Congress on Evolutionary Computation, 2001, pp. 887–894.
[16] M. Salami, T. Hendtlass, The fast evaluation strategy for evolvable hardware, Genetic Programming and Evolvable Machines 6 (2) (2005) 139–162.
[17] A. Forrester, A. Sobester, A. Keane, Engineering Design via Surrogate Modelling: A Practical Guide, John Wiley & Sons, 2008.
[18] J.P.C. Kleijnen, Design and analysis of simulation experiments, in: International Series in Operations Research & Management Science, Springer, 2009.
[19] K. Anderson, Y. Hsu, Genetic crossover strategy using an approximation concept, in: IEEE Congress on Evolutionary Computation, 1999, pp. 527–533.
[20] K. Abboud, M. Schoenauer, Surrogate deterministic mutation: Preliminary results, in: Artificial Evolution, in: LNCS, 2002, pp. 919–954.
[21] K. Rasheed, H. Hirsh, Informed operators: speeding up genetic-algorithm-based design optimization using reduced models, in: Genetic and Evolutionary Computation Conference, Morgan Kaufmann, 2000, pp. 628–635.
[22] I. Loshchilov, M. Schoenauer, S. Sebag, A mono surrogate for multiobjective optimization, in: Genetic and Evolutionary Computation Conference, 2010, pp. 471–478.
[23] Y. Jin, M. Olhofer, B. Sendhoff, A framework for evolutionary optimization with approximate fitness functions, IEEE Transactions on Evolutionary Computation 6 (5) (2002) 481–494.
[24] D. Lim, Y.-S. Ong, Y. Jin, B. Sendhoff, Trusted evolutionary algorithms, in: IEEE Congress on Evolutionary Computation, 2006, pp. 456–463.
[25] J. Branke, C. Schmidt, Fast convergence by means of fitness estimation, Soft Computing 9 (1) (2005) 13–20.
[26] M. Emmerich, A. Giotis, M. Uezdenir, T. Baeck, K. Giannakoglou, Metamodel-assisted evolution strategies, in: Parallel Problem Solving from Nature, in: LNCS, Springer, 2002, pp. 371–380.
[27] Y.S. Ong, P.B. Nair, A.J. Keane, Evolutionary optimization of computationally expensive problems via surrogate modeling, AIAA Journal 41 (4) (2003) 687–696.
[28] Z. Zhou, Y.-S. Ong, M.-H. Lim, B.-S. Lee, Memetic algorithm using multi-surrogates for computationally expensive optimization problems, Soft Computing 11 (10) (2007) 957–971.
[29] S.Z. Martinez, C.A. Coello Coello, A memetic algorithm with non gradient-based local search assisted by a meta-model, in: Parallel Problem Solving from Nature, in: LNCS, Springer, 2010, pp. 576–585.
[30] J.-F.M. Barthelemy, Approximation concepts for optimum structural design — A review, Structural Optimization 5 (1993) 129–144.
[31] M. Celis, J.E. Dennis, R.A. Tapia, A trust region strategy for nonlinear equality constrained optimization, in: P. Boggs, R. Byrd, R. Schnabel (Eds.), Numerical Optimization 1984, SIAM, Philadelphia, 1985, pp. 71–82.
[32] H.K. Singh, T. Ray, W. Smith, Surrogate assisted simulated annealing (SASA) for constrained multi-objective optimization, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–8.
[33] H.S. Bernardino, H.J.C. Barbosa, L.G. Fonseca, A faster clonal selection algorithm for expensive optimization problems, in: Artificial Immune Systems, Springer, 2010, pp. 130–143.
[34] Y. Jin, B. Sendhoff, Reducing fitness evaluations using clustering techniques and neural network ensembles, in: Genetic and Evolutionary Computation Conference, 2004, pp. 688–699.
[35] F. Mota, F. Gomide, Fuzzy clustering in fitness estimation models for genetic algorithms and applications, in: IEEE International Conference on Fuzzy Systems, 2006, pp. 1388–1395.
[36] L. Graening, Y. Jin, B. Sendhoff, Efficient evolutionary optimization using individual-based evolution control and neural networks: A comparative study, in: European Symposium on Artificial Neural Networks, 2005, pp. 273–278.

[37] L. Graening, Y. Jin, B. Sendhoff, Individual-based Management of Meta-models for Evolutionary Optimization with Applications to Three-dimensional Blade Optimization, Springer, 2007, pp. 225–250 (chapter 6).

[38] S. Tong, Active learning: theory and applications. Ph.D. Thesis, Department of Computer Science, Stanford University, 2001.

[39] D.J.C. MacKay, Introduction to gaussian processes, in: C.M. Bishop (Ed.), Neural Networks and Machine Learning, Springer, 1998, pp. 133–165.

[40] J. Sacks, W.J. Welch, T.J. Mitchell, H.P. Wynn, Design and analysis of computer experiments, Statistical Science 4 (1989) 409–435.

[41] D. Buche, N.N. Schraudolph, P. Koumoutsakos, Accelerating evolutionary algorithms with gaussian process fitness function models, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 35 (2) (2005) 183–194.

[42] M. Emmerich, K.C. Giannakoglou, B. Naujoks, Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels, IEEE Transactions on Evolutionary Computation 10 (4) (2006) 421–439.

[43] J. Knowles, Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems, IEEE Transactions on Evolutionary Optimization 10 (1) (2006) 50–66.

[44] Q. Zhang, W. Liu, E. Tsang, B. Virginas, Expensive multiobjective optimization by MOEA/D with gaussian process model, IEEE Transactions on Evolutionary Computation 14 (3) (2010) 456–474.

[45] Y. Jin, M. Huesken, B. Sendhoff, Quality measures for approximate models in evolutionary computation, in: Proceedings of GECCO Workshops: Workshop on Adaptation, Learning and Approximation in Evolutionary Computation, 2003, pp. 170–174.

[46] M. Huesken, Y. Jin, B. Sendhoff, Structure optimization of neural networks for aerodynamic optimization, Soft Computing 9 (1) (2005) 21–28.

[47] H. Ulmer, F. Streichert, A. Zell, Evolution strategies with controlled model assistance, in: Congress on Evolutionary Computation, 2004, pp. 1569–1576.

[48] M. Schmidt, H. Lipson, Coevolution of fitness predictors, IEEE Transactions on Evolutionary Computation 12 (6) (2008) 736–749.

[49] Y. Tenne, K. Izui, S. Nishiwaki, Dimensionality-reduction frameworks for computationally expensive problems, IEEE Congress on Evolutionary Computation, 2010, pp. 1–8.

[50] Z. Zhou, Y.S. Ong, P.B. Nair, A.J. Keane, K.Y. Lum, Combining global and local surrogate models to accelerate evolutionary optimization, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 37 (1) (2007) 66–76.

[51] Y. Cao, Y. Jin, M. Kowalczykiewicz, B. Sendhoff, Prediction of convergence dynamics of design performance using differential recurrent neural networks, in: International Joint Conference on Neural Networks, 2008, pp. 529–534.

[52] Y. Tenne, S.W. Armfield, A framework for memetic optimization using variable global and local surrogate models, Soft Computing 13 (2009) 81–793.

[53] Y. Jin, B. Sendhoff, Pareto-based multi-objective machine learning: an overview and case studies, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 38 (3) (2008) 397–415.

[54] T. Goel, R.T. Haftka, W. Shyy, N.V. Queipo, Ensemble of surrogates, Structural and Multidisciplinary Optimization 33 (3) (2007) 199–216.

[55] E. Sanchez, S. Pintos, N.V. Queipo, Toward an optimal ensemble of kernel-based approximations with engineering applications, Structural and Multidisciplinary Optimization 36 (3) (2008) 247–261.

[56] D. Lim, Y.S. Ong, Y. Jin, B. Sendhoff, A study on metamodeling techniques, ensembles, and multi-surrogates in evolutionary computation, in: Genetic and Evolutionary Computation Conference, 2007, pp. 1288–1295.

[57] A. Samad, K.-Y. Kim, Multiple surrogate modeling for axial compressor blade shape optimization, Journal of Propulsion Power 24 (2) (2008) 302–310.

[58] D. Chafekar, L. Shi, K. Rasheed, J. Xuan, Constrained multi-objective ga optimization using reduced models, IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews 35 (2) (2005) 261–265.

[59] A. Isaacs, T. Ray, W. Smith, An evolutionary algorithm with spatially distributed surrogates for multiobjective optimization, in: The 3rd Australian Conference on Progress in Artificial Life, 2007, pp. 257–268.

[60] D. Eby, R. Averill, W. Punch, E. Goodman, Evaluation of injection island model ga performance on flywheel design optimization, in: Third Conference on Adaptive Computing in Design and Manufacturing, 1998, pp. 121–136.

[61] M. Sefrioui, J. Periaux, A hierarchical genetic algorithm using multiple models for optimization, in: Parallel Problem Solving from Nature, 2000, pp. 879–888.

[62] P.K.S. Nain, K. Deb, Computationally effective search and optimization procedure using coarse to fine approximation, in: IEEE Congress on Evolutionary Computation, 2003, pp. 2081–2088.

[63] D. Lim, Y.S. Ong, Y. Jin, B. Sendhoff, Evolutionary optimization with dynamic fidelity computational models, in: International Conference on Intelligent Computing, in: LNCS, Springer, 2008, pp. 235–242.

[64] D. Lim, Y. Jin, Y.S. Ong, B. Sendhoff, Generalizing surrogate-assisted evolutionary computation, IEEE Transactions on Evolutionary Computation 14 (3) (2010) 329–355.

[65] J. Branke, Creating robust solutions by means of evolutionary algorithms, in: Parallel Problem Solving from Nature, in: LNCS, Springer, 1998, pp. 119–128.

[66] M. Bhattacharya, Reduced computation for evolutionary optimization in noisy environment, in: Genetic and Evolutionary Computation Conference, 2008, pp. 2117–2122.

[67] K.-H. Liang, X. Yao, C. Newton, Evolutionary search of approximated n-dimensional landscapes, International Journal of Knowledge-Based Intelligent Engineering Systems 4 (3) (2000) 172–183.

[68] H. Takagi, Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation, Proceedings of IEEE 89 (2002) 1275–1296.

[69] J.A. Biles, P.G. Anderson, L.W. Loggi, Neural network fitness functions for a musical IGA, in: The International ICSC Symposium on Intelligent Industrial Automation and Soft Computing, 1996.

[70] R.R Kamalian, A.M. Agogino, H. Takagi, Use of interactive evolutionary computation with simplified modeling for computationally expensive layout design optimization, in: IEEE Congress on Evolutionary Computation, 2007, pp. 4124–4129.

[71] X. Sun, D. Gong, S. Li, Classification and regression-based surrogate model - Assisted interactive genetic algorithm with individual fuzzy fitness, in: Genetic and Evolutionary Computation Conference, 2009, pp. 907–914.

[72] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, E. Tsang, Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization, in: The Fourth International Conference on Evolutionary Multi-Criterion Optimization, in: LNCS, Springer, 2007, pp. 832–846.

[73] I. Hatzakis, D. Wallace, Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach, in: Genetic and Evolutionary Computation Conference, 2006, pp. 1201–1208.

[74] X. Yu, Y. Jin, K. Tang, X. Yao, Robust optimization over time — a new perspective on dynamic optimization problems, in: Congress on Evolutionary Computation, 2010, pp. 3998–4003.

[75] D. Lim, Y.-S. Ong, Y. Jin, B. Sendhoff, B.S. Lee, Inverse multi-objective robust evolutionary optimization, Genetic Programming and Evolvable Machines 7 (4) (2006) 383–404.

[76] D. Lim, Y.-S. Ong, M.-H. Lim, Y. Jin, Single/multi-objective inverse robust evolutionary design methodology in the presence of uncertainty, in: S. Yang, Y.S. Ong, Y. Jin (Eds.), Evolutionary Computation in Dynamic and Uncertain Environments, Springer, 2007, pp. 437–456.

[77] Y.-S. Ong, P.B. Nair, K.Y. Lum, Max–min surrogate-assisted evolutionary algorithm for robust design, IEEE Transactions on Evolutionary Computation 10 (4) (2006) 392–404.

[78] I. Paenke, J. Banke, Y. Jin, Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation, IEEE Transactions on Evolutionary Computation 10 (4) (2006) 405–420.

[79] T.P. Runarsson, Constrained evolutionary optimization by approximate ranking and surrogate models, in: Parallel Problem Solving from Nature, in: LNCS, Springer, 2004, pp. 401–410.

[80] S.D. Handoko, C.K. Kwoh, Y.S. Ong, Feasibility structure modeling: an effective chaperon for constrained memetic algorithms, IEEE Transactions on Evolutionary Computation 14 (5) (2010) 740–758.

[81] Y. Tenne, K. Izui, S. Nishiwaki, Handling undefined vectors in expensive optimization problems, in: Applications of Evolutionary Computation, in: LNCS, Springer, 2010, pp. 582–591.

[82] Y. Jin, S. Oh, M. Jeon, Incremental approximation of nonlinear constraint functions for evolutionary constrained optimization, in: IEEE Congress on Evolutionary Computation, 2010, pp. 2966–2973.

[83] S. Oh, Y. Jin, M. Jeon, Approximate models for constraint functions in evolutionary constrained optimization, International Journal of Innovative Computing, Information and Control 7 (10) (2011).

[84] M.K. Karakasis, K.C. Giannakoglou, Metamodel-assisted multi-objective evolutionary optimization, in: Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems, EUROGEN, 2005.

[85] A. Shahrokhi, A. Jahangirian, A surrogate assisted evolutionary optimization method with application to the transonic airfoil design, Engineering Optimization 42 (6) (2010) 497–515.

[86] V.G. Asouti, I.C. Kampolis, K.C. Giannakoglou, A grid-enabled asynchronous metamodel-assisted evolutionary algorithm for aerodynamic optimization, Genetic Programming and Evolvable Machines 10 (4) (2009) 373–389.

[87] M.H.A. Bonte, L. Fourment, T.-T. Do, A.H. van den Boogaard, J. Huetink, Optimization of forging processes using finite element simulations, Structural Multidisciplinary Optimization 42 (2010) 797–810.

[88] K. Hamza, K. Saitou, Crashworthiness design using meta-models for aprroximating the response of structural members, in: Cairo University Conference on Mechanical Design and Production, 2004.

[89] G. Mariani, G. Palermo, C. Silvano, V. Zaccaria, Meta-model assisted optimization for design space exploration of multi-processor systems-on-chip, in: 12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools, 2009, pp. 383–389.

[90] G. Dellino, P. Lino, C. Meloni, A. Rizzo, Kriging metamodel management in the design optimization of a cng injection system, Mathematics and Computers in Simulation 79 (8) (2009) 2345–2360.

[91] S.-T. Khu, D. Savic, Z. Kapelan, Evolutionary-based meta-modelling v the relevance of using approximate models in hydroinformatics, in: Hydroinformatics in Practice: Computational Intelligence and Technological Developments in Water Applications, Springer, 2008.

[92] I. Dahm, J. Ziegler, Using artificial neural networks to construct a meta-model for the evolution of gait patterns, in: Proceedings of the 5th International Conference on Climbing and Walking Robots, 2002.

[93] M.J. Powell, On the convergence of a wide range of trust region methods for unconstrained optimization, IMA Journal of Numerical Analysis 30 (1) (2010) 289–301.

[94] V.J. Hodge, K.J. Lees, J.L. Austin, A high performance k-nn approach using binary neural networks, Neural Networks 17 (3) (2004) 441–458.

[95] R. Li, M.T.M. Emmerich, J. Eggermont, E.G.P. Bovenkamp, T. Bäck, J. Dijkstra, J.H.C. Reiber, Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis, in: IEEE Congress on Evolutionary Computation, 2008, pp. 2764–2771.
[96] C. Ling, J.-H. Liang, B. Wu, Z.-W. Hu, A metamodel-based optimization method for mobile ad hoc networks, in: 2010 International Conference on Computer Application and System Modeling, ICCASM, 2010, pp. 412–416.
[97] Y. Jin, J. Branke, Evolutionary optimization in uncertain environments-a survey, IEEE Transactions on Evolutionary Computation 9 (3) (2005) 303–317.
[98] R. Polikar, L. Upda, S.S. Upda, V. Honavar, Learn++: An incremental learning algorithm for supervised neural networks, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 31 (4) (2001) 479–508.
[99] D.H. Ackley, A Connectionist Machine for Genetic Hillclimbing, Kluwer Academic Publishers, Norwell, MA, 1987.
[100] W. Hendrickx, D. Gorisson, T. Dhaene, Grid enabled sequential design and adaptive metamodeling, in: Proceedings of the 2006 Winter Simulation Conference, 2006.
[101] M. Miller, Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online, Que Publishing, 2008.