A Systems Approach to Evolutionary Multi-Objective Structural Optimization and Beyond

Yaochu Jin and Bernhard Sendhoff

***Abstract*— Multi-objective evolutionary algorithms (MOEAs) have shown to be effective in solving a wide range of test problems. However, it is not straightforward to apply MOEAs to complex real-world problems. This paper discusses the major challenges we face in applying MOEAs to complex structural optimization, including the involvement of time-consuming and multi-disciplinary quality evaluation processes, changing environments, vagueness in formulating criteria formulation, and the involvement of multiple sub-systems. We propose that the successful tackling of all these aspects give birth to a systems approach to evolutionary design optimization characterized by considerations at four levels, namely, the system property level, temporal level, spatial level and process level. Finally, we suggest a few promising future research topics in evolutionary structural design that consist in the necessary steps towards a life-like design approach, where design principles found in biological systems such as self-organization, self-repair and scalability play a central role.**

***Index Terms*— Evolutionary multi-objective optimization, structural optimization, efficiency and scalability, robustness, systems approach**

I. INTRODUCTION

Evolutionary multi-objective optimization has witnessed a great success in solving a wide range of scientific and engineering problems [1], [2]. One of the most successful yet challenging application of evolutionary optimization is structural optimization [3], such as aerodynamic optimization [4]. Structural optimization is often characterized by requirements at four levels. First, structural design optimization involves complex, multi-disciplinary processes, which are often computationally very expensive to simulate, or very costly to do experiment with. For example, in design optimization of micro heat exchangers, both thermodynamic efficiency and aerodynamic efficiency must be optimized. That is to say, a completeness of system properties must be taken into account. Second, structural design optimization requires the consideration of the performance of a designed structure in the whole lifetime, including robustness to a changing environment, its maintenance and disposal. Third, it is nontrivial to specify the criteria to be optimized in some complex structural design problems. In this sense, formulating the problem itself is a part of the problem-solving process. Finally, complex structural design can consist of multiple spatially distributed sub-systems, which however, must be taken into account simultaneously during the optimization. We believe that the fulfillment of the aforementioned four levels of requirements will lead to a novel and systematic approach to design complex structure, which we term it *a systems approach*.

Y. Jin and B. Sendhoff are with the Honda Research Institute Europe, Carl-Legien-Str. 30, 63073 Offenbach, Germany. Email: yaochu.jin@honda-ri.de.

One major concern in the systems approach to complex structural design is the scalability and efficiency of the MOEA. First, the design space is high-dimensional, which can be attributed to the requirements both at the system property level and the spatial level. For solving high-dimensional problems, developing scalable MOEAs is of paramount importance. In addition, involvement of multiple disciplines and difficulties in specifying the optimization criteria may result in a high number of objectives, where the scalability to the number of objectives comes into play. Second, assessments of the quality of candidate designs often involve computationally time-consuming simulations or expensive experiments. This problem becomes extraordinary severe, when additional fitness evaluations are needed to search for robust solutions to meet the temporal level of requirements. Thus, it is essential to enhance the efficiency of MOEAs for structural optimization. To enhance scalability and efficiency of MOEAs for structural design, a number of measures can be taken in geometry representation, genetic encoding, genetic operators and fitness evaluations.

The remainder of the paper is organized as follows. Section II gives a brief introduction to main components in evolutionary multi-objective structural optimization (EMOSO). In the sections that followed, main aspects of a systems approach to EMOSO are discussed. Section III presents various methods for efficiency enhancement in EMOSO, focusing on geometry representation and surrogate-based fitness evaluations. Approaches to the improvement of the scalability of EMOSO are discussed in Section IV, where algorithms that are able to capture problem structure during optimization are advocated. Bio-inspired representation is also considered a promising methodology for scalable EMOSO. In Section V, we discuss strategies for the search of robust solutions and life-long optimization to deal with temporal level of considerations in EMOSO. Requirements at the spatial and process levels are studied briefly in Section VI. Section VII suggests a few promising future research topics stressing features such as self-organization, self-repair and scalability inspired by biology, which we believe will lead to a life-like design methodology. A summary of the paper is provided in Section VIII.

II. EVOLUTIONARY MULTI-OBJECTIVE STRUCTURAL OPTIMIZATION (EMOSO)

Evolutionary algorithms (EAs) are meta-heuristic search methods inspired by biological evolution. Research on using evolution-like computational algorithms for solving engineering optimization problems and creating self-organizing intelligent systems can be traced back to the late 1950s [5].

Over the last few decades, evolutionary computation, including a number of variants of evolution-inspired algorithms such as genetic algorithms, evolution strategies, evolutionary programming, genetic programming, and other bio-inspired metaheuristics, including particle swarm intelligence, ant colony, and differential evolution, has grown to be a new discipline of computer sciences.

A. A generic diagram of EMOSO

The success of evolutionary computation is characterized not only with rapid advances in theory and algorithmic development, including the boom of research on multi-objective evolutionary algorithms (MOEAs) in the last decade, but also with a wide range of applications in almost all areas of science and technology, such as optimization and control, scheduling, decision-making, robotics, finance, game theory, artificial life and computational systems biology, just to name a few.

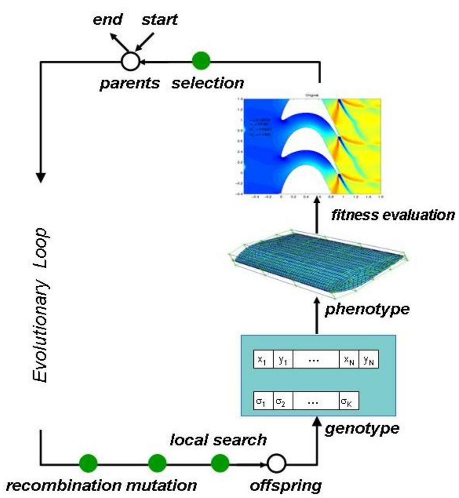

Fig. 1. A generic framework of EMOSO.

Among these applications, evolutionary structural optimization has attracted attention from both academia and industry. A generic framework for evolutionary multi-objective structural optimization (EMOSO) is illustrated in Fig. 1. From the figure, we can see that EMOSO consists of four main components, namely, genotype-phenotype mapping (including geometry representation and genetic encoding), fitness evaluation, selection, and genetic variations, such as crossover, mutation and local search. All these components play a role in the efficiency of EMOSO, where geometry representation and fitness evaluation are two components of particular importance in structural optimization.

B. Differences between single-objective optimization (SOO) and multi-objective optimization (MOO)

There seems to be no essential difference in the generic diagram of EMOSO compared to that of evolutionary single-objective structural optimization (ESOSO). However, intrinsic differences do exist between single-objective optimization (SOO) and multi-objective optimization (MOO), which are listed in Table I. These differences lead directly to different considerations in developing single-objective evolutionary algorithms (SOEAs) and MOEAs. Apart from the well understood difference in selection between SOEAs and MOEAs, the following differences are less attended, which, however, have significant influence on the efficiency and scalability of MOEAs, as first discussed in [6]:

- The concept of convergence is completely changed. In SOEAs, the evolutionary search is converged when all individuals of a population move to a local or global optimum, that is, when all individuals sit on the same point in the search space. After the population of an SOEA is converged to an optimum, the population will stay with the optimum when the crossover operator is applied to a genetic algorithm. For a single-objective evolution strategy, the population will remain converged even if the mutation operator is used. This can be attributed to the fact that the step-size used in evolution strategies will also converge to zero after the population is converged. By contrast, a population converged to the Pareto front will diverge again when crossover is applied to a genetic algorithm based MOEA. As a consequence, crossover must be constrained to neighboring individuals to have fine search in a genetic algorithm based MOEA [7]. An evolution strategy based MOEA is able to converge to the Pareto front when mutations are applied only. Such a difference implies that differences in genetic representation and genetic operators will result in a much larger difference in search dynamics of MOEAs [8], and thus the search efficiency.
- Some genetic operators may give rise to additional problems unknown to SOEAs. A good example is that a genetic algorithm with the bimodal normal distribution crossover (BNDX) may get trapped in a local Pareto front that is not resulted from the problem to be solved. As illustrated in Fig. 2, once the individuals in a population align into a linear curve, the population is no longer able to move toward the true Pareto front if population-centric genetic operators such as BNDX is used only to generate new offspring, because any newly generated offspring will stay on the line. The population is able to escape from the local Pareto front only if perturbations are deliberately added in generating new offspring in the direction orthogonal to the line.
- Model based EAs gain a few additional advantages in MOO compared to SOO for the following reasons. First, since most multi-objective optimization problems have an infinite number of Pareto-optimal solutions, model based EAs, such as the estimation of distribution algorithm (EDAs), are more efficient in representing Pareto-optimal solutions, because the number of solutions that can be represented by a model is not limited by the size of the population or of the archive. In addition, there are often regularities in the distribution of Pareto-optimal solutions [9]. Thus, model based algorithms are more capable of capturing these regularities.

TABLE I

FUNDAMENTAL DIFFERENCES BETWEEN SOO AND MOO

	SOO	MOO
Target	• Find the global optimal solution	• Achieve the Pareto-optimal solution set or a representative subset
Performance Indexes	• Accuracy • Efficiency	• Accuracy • Spread • Distribution • Efficiency
Problem Structure	• Fitness landscape – ruggedness – deceptiveness – multi-modality – correlation, etc.	• Fitness landscape – ruggedness – deceptiveness – multi-modality – correlation, etc. • Distribution of the Pareto-optimal solutions – finite/infinite – convex, concave – continuous, discrete – curve, surface, etc.

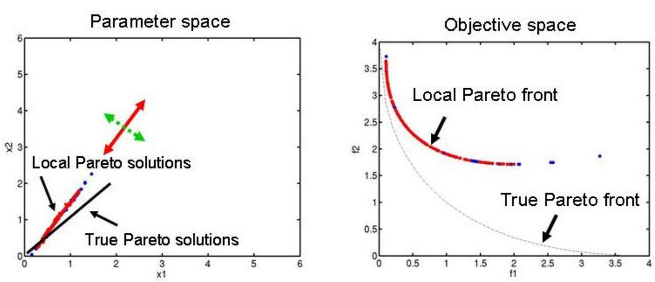

Fig. 2. Pseudo local Pareto front introduced by the BNDX on the 30-D Schaffer function.

We will discuss in more detail in Section III-C.1 and Section IV-A how to enhance the efficiency and scalability of EMOSO, being aware of the aforementioned differences in the search dynamics of SOEAs and MOEAs.

III. EFFICIENCY ENHANCEMENT IN EMOSO

As discussed in Section I, efficiency is one major concern in EMOSO. In this section, we will focus on two aspects of particular importance to EMOSO in improving the efficiency, namely, geometry representation and fitness evaluations. General aspects relevant to efficiency improvement such as genetic variations and knowledge incorporation will also be discussed.

A. Geometry representation

Geometry representation is the first step in EMOSO, which plays an essential role in the efficiency and scalability of evolutionary search. In this section, we first mention a few general requirements in choosing a geometry representation. We then introduce a few selected geometry representation methods and discuss their strengths and weaknesses. Following that, we point out that multiple representations or adaptive representations may be needed to overcome the weaknesses in a single geometry representation.

1) General requirements: An optimal geometry representation is always problem specific. Nevertheless, there are a few general requirements in choosing a geometry representation, which is closely related to the requirements on the genotype-phenotype mapping for an evolvable and robust evolutionary search.

- A geometry representation should be complete and compact. Completeness means that the geometry representation should be able to describe all feasible solutions. Meanwhile, the representation should be compact to reduce the dimension of the search space.
- A geometry representation should be unbiased unless a priori knowledge is available indicating that a particular space is more promising in finding the global optimum.
- A geometry representation should be causal and local. Causality means that the neighborhood in the genotype space should be conserved in the phenotype space. On the other hand, local changes in the genotype space should be allowed to make fine tuning.

Although the above requirements are true in general, two issues are worthy of further discussions. First, there may be a trade-off between completeness and compactness of the representation, because compact representations may limit the flexibility in the representation, thus harming the completeness of the representation. Second, causality can be a double-side sword. Causality is often required for a fine tuning, and is a condition for self-adaptation of search parameters in EAs. Nevertheless, strict causality may degrade the ability to search in an innovative way, and does not allow for neutrality. As we know, innovation and neutrality are two important yet related facets for evolvability of EAs [10]. The reader is referred to [11] for a more comprehensive and general discussion on genetic representations.

Two possible approaches can be taken to address the trade-off between completeness and compactness requirements in geometry representation, which will be discussed in the next sections.

2) Methods for geometry representation: Several method can be used for geometry representation for EMOSO. A few often used geometry representations are:

- Direct representation. In this method, the geometry is represented by a number of linearly connected points in terms of their coordinates in a Cartesian space. The merit of this method is that it is complete, however, this method is not compact. A large number of parameters are needed for representing a complex shape.
- Parametrized representation. Parametrized optimization often needs additional domain knowledge. For example, a two-dimensional turbine blade can be described by circles at the leading edge, in the middle and the trailing edge [12]. The main advantage of such representations is their compactness. Unfortunately, the flexibility of parametrized representations is very low and such representations do not meet the completeness requirements.
- Non-parametrized representations. Non-parametrized representations include Bezier curves, splines, and non-uniform B-splines (NURBS) [13]. These methods are rather flexible and relatively compact. In addition, when computational fluid dynamics (CFD) simulations are involved in fitness evaluations, which often happens in structural optimization, a re-meshing for the CFD calculation is needed, which is a highly time-consuming process [14], [15].
- Free form deformation (FFD) [16]. FFD-based geometry representation is more compact, compared to non-parametrized representations such as NURBS, since FFD accounts only for the changes in the geometry in modifying the geometry [17]. An additional advantage of FFD is that in FFD based design, deformation is applied to both the geometry and the CFD mesh for fitness evaluations, thus no re-meshing is needed. The major weakness with FFD-based geometry representations is that modifications are made to control points, which will affect the whole shape and everything inside the control volume, thus making it difficult to implement local modifications. An approach to addressing this weakness is the so-called direct manipulation of FFD (DM-FFD) [18], [19]. In DM-FFD, the points to be modified are chosen directly on the shape. The required changes in the control points for realizing the desired changes in the shape are then calculated using an optimization algorithm, e.g., the least square method.

Fig. 3 illustrates three widely used geometry representations, i.e., parametrized representation, NURBS, and FFD.

3) Multiple and adaptive representations: As we can see in the previous section, no single representation seems to be able to meet all the requirements for geometry representation. There are two possible approaches to dealing with this problem. First, multiple representations can be used simultaneously for a design. A good example is the design optimization of a micro heat exchanger, where the heat transfer rate is to be maximized and the pressure drop of the flow needs to be minimized [20]. For this problem, it is found that only part of the Pareto front can be achieved with a single geometry representation, as shown in Fig. 4. In this design example, the upper left segment of the Pareto front can be obtained only by using a sine function where the frequency and amplitude are optimized. On the other hand, the lower right part of the Pareto front can be obtained using a spline representation, which is missing if a sine representation is adopted.

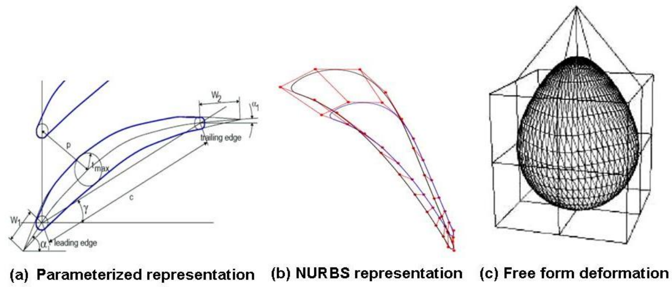

Fig. 3. Three typical geometry representations. (a) Parametrized representation, (b) NURBS, (c) FFD.

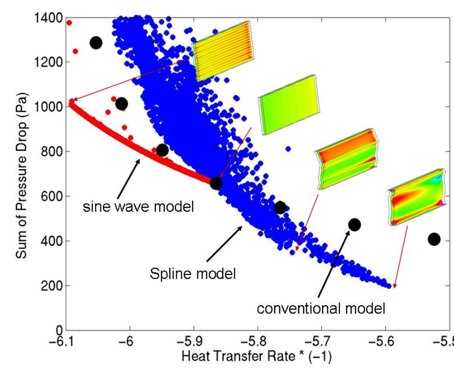

Fig. 4. Obtained solutions using different geometry representations in an evolutionary optimization of a micro heat exchanger.

A method to tackle the completeness and compactness trade-off in geometry representation is to introduce an adaptive representation [21], [22]. The main idea is to begin with a small number of design variables. As the evolutionary optimization proceeds, additional free design variables are introduced to increase the flexibility of the representation. A question that may arise is why not use a more flexible representation from the beginning. This can be attributed to the fact that the search efficiency of most evolutionary algorithms seriously degrades as the dimension of search space increases [23]. Thus, if a higher number of variables is adopted from the beginning, the EA may fail to find the optimum. With an adaptive representation where new variables are introduced step by step during the evolution, it is expected that it is easier for EAs to find the global optimum.

It should be noticed that several issues need to be addressed appropriately to ensure that an adaptive representation works better than a fixed representation. One important point is that the new variables should be introduced in such a way that the phenotype is neutral to the newly included variables. This kind of neutral mutation is key to the success of adaptive representations. Nevertheless, it is unclear where on the geometry and when during the evolution should the new design variables be introduced. The "where" problem can be solved to a certain degree by e.g., finding out the location of the shape where modifications are most likely to improve the quality of the design. As to the "when" problem, it has been found that it is a good strategy to introduce new variables long before the population converges.

B. Surrogate-based fitness evaluations

One most popular and effective approach to efficiency enhancement in EMOSO is to reduce the computational cost for fitness evaluations. To this end, various approximation methods can be used, including problem approximation, where, e.g., a full-scale simulation is replaced with a reduced-scale of simulations, functional approximation, where computationally efficient surrogates, or meta-models are used instead of an expensive simulation, and EA-specific approximation methods that estimate the fitness value of an individual from its parents or from other individuals. Among these approximation methods, surrogate-assisted evolutionary optimization is the most popular approach. However, it is highly recommendable to use different approximation methods simultaneously.

Surrogate-assisted evolutionary optimization has attracted increasing research interest in recent years [24]. The following three questions must be answered when surrogates are used in an attempt to enhance the efficiency of EMOSO.

- Where surrogates can be used in EAs. Generally speaking, surrogates can be used wherever a fitness evaluation is needed. For example, fitness approximation can be helpful in reducing randomness in reproduction [25]. During reproduction, instead of passing all individuals generated by crossover or mutation into the offspring population, candidate individuals are passed to the offspring generation only if they survive a pre-selection according to the surrogate. In addition, surrogates can also be employed in local search of a memetic evolutionary method [26]. The most common way to use a surrogate is in selection.
- How surrogates can be used. This is often known as model management, and is particularly important when surrogates are used directly in selection. Surrogates can mislead the evolutionary search if they are not used properly due to the fact that surrogates may introduce approximation errors and even false optima [27]. A basic principle is to use surrogates together with the exact fitness evaluations [1]. In general, we can categorize existing methods into three groups. First, individual-based methods. In these methods, only part of the individuals in a population at a certain generation are evaluated with the exact fitness function [28], [29], [30]. In these methods, the essential issue is which individuals should be chosen for re-evaluation using the exact fitness function. Second, generation-based methods. In these frameworks, the surrogate is used for fitness evaluations in some of the generations, and in the rest of the generations, the exact fitness function is used [31], [30], [27]. Third, population-based approaches. In these methods, a few sub-populations co-evolve, where different sub-populations use different surrogates [32], [33]. Usually, these surrogates are of different accuracy. Individuals are allowed to migrate among sub-populations.
- How the quality of the surrogates can be assessed and how to improve the quality of the surrogates. The first part of the question appears straightforward. In machine learning, the quality of a learning model can be judged by the approximation error on unseen data, i.e., the model's generalization ability [34]. This must not be true when the model is used for fitness evaluations as a surrogate of the exact fitness function. The reason is that in selection, we are not directly concerned with the approximation accuracy, rather a correct selection. Thus, various model assessment criteria have been proposed for comparing surrogates [35], [36], [37]. The second part of the question is concerned with the question about which kind of surrogate models, such as global models or local models, should be preferred for surrogates [24]. In general, an ensemble model is better than a single model, not only because an ensemble is able to provide a more accurate prediction, but also because an ensemble contains information about if a prediction is sufficiently reliable [38], [39], [40]. Recently, a dual surrogate memetic framework has been proposed [41], [42], where a combination of global and local models are used in surrogate-assisted local search. It has been shown that the dual surrogate framework performs better in that the global model can smooth the global fitness landscape, thus achieving the effect of *the bless of uncertainty*, whereas a local model is able to accurately predict in a local neighborhood of the fitness landscape, thus addressing the *curse of uncertainties*. Fig. 5 illustrates how approximation errors introduced by a surrogate influence the dominance comparison. It can be seen from the figure that approximation errors may slow down the convergence to the Pareto front in some cases (left panel), but can speed up the convergence and even enhance diversity as well (right panel).

  When a surrogate is used to predict a fitness value solved by a CFD simulation, where a large number of iterations must be carried out before convergence, a recurrent neural network can be switched on during the CFD iterations to reduce the number of needed CFD iterations, thus reducing the computational cost of fitness evaluations [43]. This is applicable to any iterative quality evaluation methods. A diagram illustrating how to predict the solution of CFD simulations in an evolutionary loop

[1] In solving complex engineering problems, it is almost impossible to calculate the fitness value of a design exactly. More often, there are more exact approximations, such as experiments or full-scale simulation, or less exact approximations, such as reduced-scale simulations or surrogates. In general, there is a trade-off between approximation accuracy and computational cost among different approximation methods. In this case, the exact fitness function means the most accurate approximation.

is given in Fig. 6.

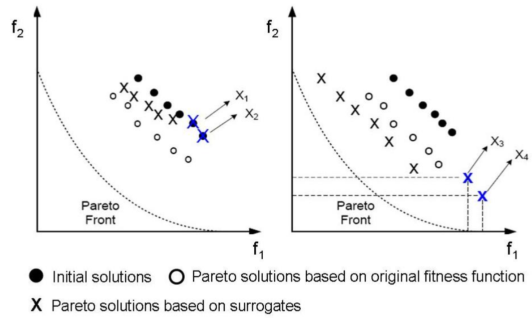

Fig. 5. Curse of uncertainties in dominance comparison (left panel) and bless of uncertainties (right panel).

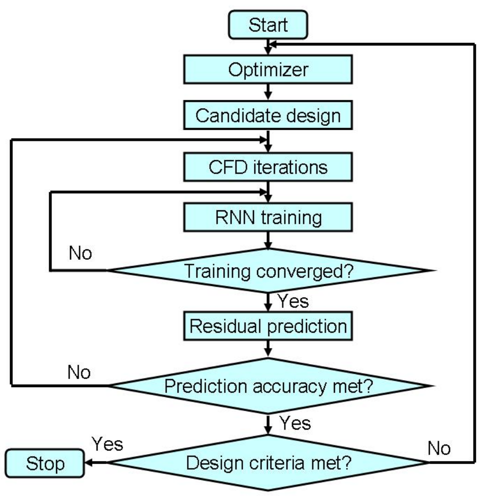

Fig. 6. A diagram showing how recurrent neural networks can be used for predicting results in CFD iterations.

In many cases, it is impossible to define a concrete fitness function when the desired quality cannot be described by an explicit mathematical function, for example in interactive evolution [44]. In this case, a surrogate is helpful in reducing the burden of a human user.

C. Other aspects

In addition to geometry representation and surrogate-assisted evolutionary optimization, a few other aspects can be considered in enhancing the efficiency of EMOSO, some of them are particular to multi-objective design, others are generally valid for evolutionary optimization.

1) Genetic representation and genetic operations: The difference in the search dynamics between a binary genetic algorithm with crossover and a real-coded evolution strategy with Gaussian mutations for SOO has been discussed [45]. For MOO, as indicated in Section II-B, the difference becomes even more significant. One main phenomenon is that a binary GA with crossover can never converge in MOO when no constraints on crossover are imposed. This means that a binary GA with crossover performs poorly for local search, even when it is converging to the Pareto front. An evolution strategy, in contrast to a binary GA, can still do fine local search if the step-size is properly self-adapted so that at the end of the evolution, it becomes sufficiently small.

For this reason, a hybrid representation can improve the search efficiency of EMOSO. In [46], an evolutionary algorithm with a hybrid representation has been proposed. In this algorithm each individual can potentially use either a binary representation or a real-coded representation. Once a binary representation is activated, crossover will be used for generating offspring. When a real-coded representation is in use, Gaussian mutations are used. Two questions must be answered. First, which representation should be activated. For this question, an additional gene, whose value can be zero or one, is included in the representation, where zero means that the binary representation is activated, and one means the real-valued representation. The second question is how to synchronize between the two representations. This is particularly important when a binary coding is switched to a real-valued coding, where the step-size self-adaptation must be performed. In [46], the step-size is estimated by calculating the distance between the parent and its offspring generated using crossover. It has been shown that an MOEA with a hybrid representation consistently outperform one with a single representation on the problems used in [46]. More interestingly, it has been shown that a binary representation using crossover is preferred at the beginning of search while a real-valued coding overwhelms at the end of evolution.

2) Use of parallel and grid computing: To enhance the efficiency of EMOSO from the computational point of view, parallel computing and grid computing techniques are almost a must for EMOSO. Fig. 7 shows a parallel computing paradigm, where two levels of parallelization are implemented. The first level is done on the population level, where individuals are evaluated in parallel on separate machines. The second level of parallelization is implemented during the fitness calculation of each individual. Since a three-dimensional (3D) CFD simulation takes several hours on a single computer, the simulation process is again parallelized on four processors.

This parallel computing architecture is satisfactory if the machines are of similar performance, in the same physical location, and if the fitness evaluations for all individuals take similar computation time. This assumption may be violated for two reasons. First, computational resources may be heterogeneous in computational speed, operation platform, communication protocol, and may be distributed at geographically different locations. In addition, the time taken to evaluate individuals can be very different, particularly when surrogate models of different fidelity are used in a population. If the standard master-slave parallelization architecture is adopted and a generational EA is used, the evaluation time of a generation depends on the slowest evaluation of a population, which may be a waste of computing resources. To address this problem, there are also two possible solutions. One is to use a steady-state evolutionary algorithm instead of a generational one. In a steady-state EA, only the worst individual or a few worst individuals are replaced, therefore it is not necessary to wait until the whole population is evaluated. Once an individual is

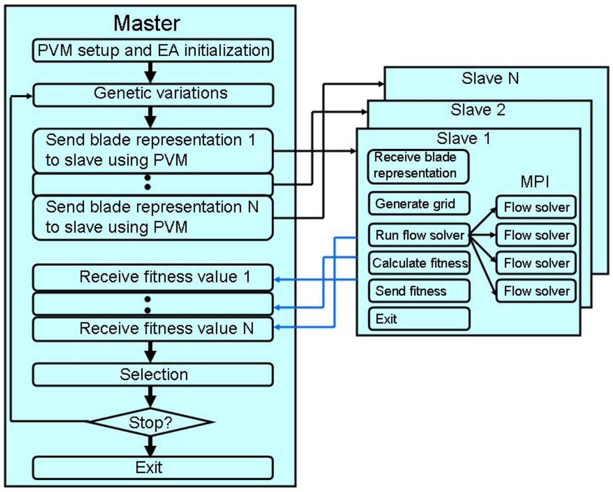

Fig. 7. A computing architecture with both population level and individual level of parallization.

evaluated, a reproduction can be done immediately [47].

However, steady-state EAs may not be applicable to multi-objective optimization without degrading search performance. A preferable solution is to develop a computing architecture that can still provide the speed-up regardless of the heterogeneity in the computing resources or surrogates while preserving the standard behavior of the parallel evolutionary search. A good example of such solutions is the decoupled grid-enabled hierarchical parallel genetic algorithm (DGE-HPGA) proposed in [48]. The DGE-HPGA consists of two-level parallelization. At level one, the island model is used for parallelization, whereas at level two, a master-slave parallelization scheme is adopted. The two levels of the hierarchical parallel genetic algorithm (PGA) are gridified to form the 'sub-population evolution' and 'chromosome evaluation' Grid services. Furthermore, the 'sub-population evolution' Grid service is decoupled into two separate services, namely, the 'evolutionary operations' at the client(master) and 'chromosome ensemble' at each computing cluster. In this way, the computing clusters are used for fitness evaluations only and genetic operations are processed at the client side. As a result, a constant sub-population size can be maintained at the client side while non-uniform ensembles are allocated to the computing clusters of chromosomes for fitness evaluations. The DGE-PGA has shown to be significantly more efficient than a non-decoupled grid-enabled PGA on an airfoil design optimization example.

3) Domain knowledge discovery by mining history data: When using an adaptive representation, one concern is where the new variables should be included in the geometry. In this case, acquiring domain knowledge may be very helpful. One interesting example is the work reported in [49], where a displacement measure is proposed for informing the amount and direction of surface modifications. Such information, in combination with data mining techniques for rule extraction, can provide useful insights into the evolutionary search and thus improve its efficiency. Domain knowledge extracted during the optimization can also be taken advantage of to identify potentially high performance region to reduce search space [50], or to support decision-making [51]. A data mining method using self-organizing map has been proposed to identify the design variables that are responsible for performance improvement and trade-offs [52]. More general methods for discovering and reusing domain knowledge in evolutionary search can be found in [53].

IV. SCALABILITY OF EMOSO

A. Capturing problem structure

In EMOSO, evolutionary algorithms have to deal with high-dimensional search spaces due to the fact that complex structures often have a large number of design variables to be optimized. For this reason, scalability of the MOEA for complex structural design is of pivotal importance. We believe that a scalable MOEA should be able to discover and take advantage of the structure of the problem at hand. As discussed in Section II-B, being able to capture the problem structure is particularly important and also possible in MOO, because there are often regularities in the distribution of the Pareto-optimal solutions. However, most MOEAs do not pay sufficient attention to regularities in Pareto front [9]. By regularity, we mean two aspects. First, Pareto-optimal solutions are often connected. Second, according to the Karush-Kuhn-Tucker condition, the Pareto front is a (m-1)-dimensional piece-wise continuous manifold for continuous MOPs with m objectives. The connectedness of the Pareto-optimal solutions justifies the use of local search in MOO, either in combination with an evolutionary search [54] or the dynamic weighted aggregation method [55] and other similar approaches [56]. The second aspect is particularly useful in model-based evolutionary algorithms, such as the estimation of distribution algorithms [57], where reduction of dimension is key to the success of model building. Thus, taking advantage of regularity in multi-objective optimization is key to developing scalable MOEAs. Regularity model-based multi-objective estimation of distribution algorithm (RM-MEDA) is a good example of such scalable MOEAs [58], [59]. In RM-MEDA, a probabilistic model is built up to model the population, which consists of one deterministic part, a principal curve, and a probabilistic part, a Gaussian model. It has been shown that regularity modeling contributes significantly to the scalability of MOEAs for solving problems with strongly correlated design variables using a reasonably small population size. Fig. 8 shows the performance landscape in D-metric [60] with respect to population size and the search dimension of the RM-MEDA, of the univariate factorized Gaussian model (UGM), and of the marginalized multivariate Gaussian model (MGM). From the figure, we can see that the performance of RM-MEDA is highly scalable to the search dimension, and relatively insensitive to the population size. In contrast, the performance of both UGM and MGM deteriorates quickly as the search dimension increases.

The RM-MEDA has also been employed for 3D blade design optimization. The NURBS based geometry representation is shown is Fig. 9, and the optimization results are plotted in Fig. 10, together with solutions achieved from various single-objective or multi-objective optimization approaches. It can

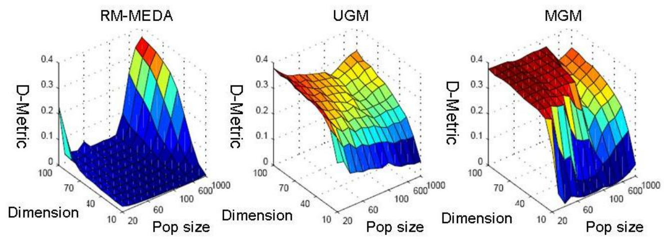

Fig. 8. Scalability of three MOEAs on the test problem ZDT2.2, whose design parameters are nonlinearly correlated. Left panel: RM-MEDA. Middle panel: UGM. Right panel: MGM.

be seen that the Pareto front achieved by the RM-MEDA outperforms those from other algorithms.

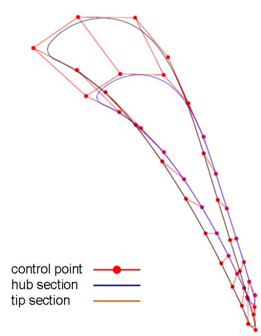

Fig. 9. NURBS representation of a 3D turbine blade of a gas turbine.

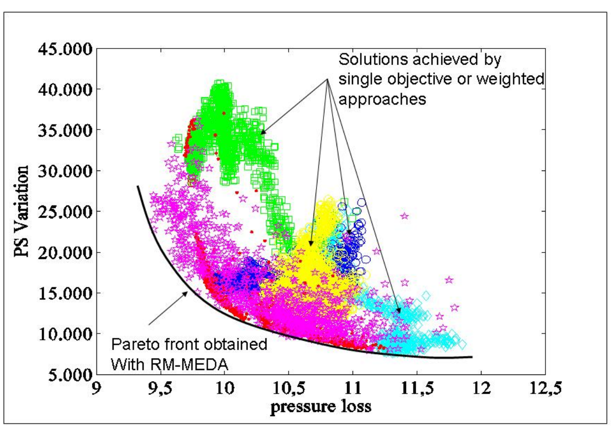

Fig. 10. Results from the 3D design optimization as represented by NURBS in Fig. 9.

Another issue in MOEAs that has received increasing attention is algorithms' scalability to the number of objectives, which is often known as many-objective optimization. Several approaches have been developed for solving many-objective optimization with evolutionary algorithms [61], such as modifying the dominance definition, using performance indicators as fitness function, using preference information, or reducing the number of objectives. Nevertheless, one fact seems to become clear that the Pareto-ranking based approach to sorting the population will dramatically decrease the efficiency of MOEAs, thus worsening the scalability of MOEAs to the number of objectives.

B. Bio-inspired scalable representations

Most MOEAs for structural optimization use a direct encoding, where the genotype-to-phenotype mapping in these algorithms are explicit and static. On the other hand, it is believed that developmental encodings are more scalable, particularly when there are repeated structures in the target design, although it is non-trivial to show the benefit of developmental representations in optimization [62]. In [63], a multi-cellular model is proposed for designing lightweight and stable structures. In that cellular model, there are two types of cells, namely, material cells that have a mass and can bear physical forces, and void cells that occupy a space but do not have a mass and cannot bear any force. Each cell, no matter whether it is a material cell or a void cell, contains a virtual DNA that forms a gene regulatory network determining whether a cell divides, if yes, the type of the daughter cells. In addition to the genetic control of the cellular system, physical cell-cell interactions including cell adhesion are also modeled.

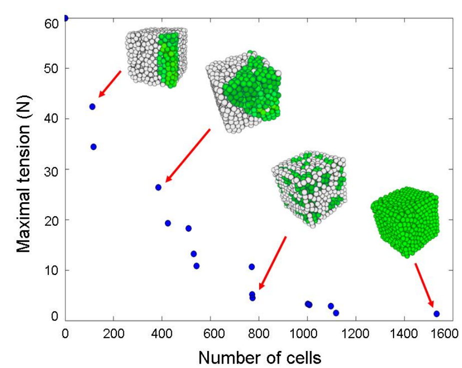

Fig. 11. Pareto-optimal solutions for lightweight stable structures obtained from a cellular growth model.

There are only six parameters in total in the cellular model that regulates the cell growth process. Nevertheless, complex structures can be evolved with this cellular model through a developmental process. The system starts with a few cells of both material and void cells. During the development, cells will replicate themselves when the gene for cell division is activated. The gene regulatory network is evolved using the non-dominated sorting genetic algorithm (NSGA-II) [64] for designing lightweight and stable structures, where the two objectives are optimized. One is the total mass of the structure denoted by the number of material cells in the structure and the other is the stability of the structure denoted by the maximal internal tension when a force is applied on the surface of the structure. Fig. 11 shows the Pareto-optimal solutions of one evolutionary run of the cellular approach to structural design. These solutions have a quite complex inner structure. These results suggest that a bio-inspired representation and design approach may be promising in improving the scalability of a class of EMOSO.

V. Robust and Life-long Optimization: Temporal Level Considerations

A systems approach to EMOSO considers not only the performance of the design at a certain time or under a certain condition, but also in the whole life of the design, e.g., the maintenance and sustainability of the design. Different measures can be taken to address these concerns. For instance, for easier maintenance, the performance of the design should be insensitive to small changes in the design due to deformation or worn-out effect, and changes in the environment during the life time. This is known as the robustness requirement on the design. Robust solutions, however, are not sufficient to tackle large changes in the design or in the environment. In this case, a life-long optimization will be indispensable. The simplest form of life-long optimization is dynamic optimization, where the changes in the environment are embodied in the changes of the fitness over time, which is usually known as dynamic optimization. We will elaborate on these two aspects in the following.

A. Search for robust solutions

Robustness and sustainability is a temporal level of requirement in a systems approach to structural design that not only takes the performance of the structure at one operation point, but also that during its entire lifetime, its maintenance or even its disposal into account. In this section, we mainly discuss evolutionary methodologies for the search of robust solutions when the evolved structure is subject to mild changes in the design and environment. Several measures are available for defining robust solutions [65]. In evolutionary optimization, four methods can be employed for achieving robust solutions [66]. The first method is known as implicit averaging, where random perturbations are added into the phenotype during fitness evaluations [67]. It has been shown that for genetic algorithms with an infinite population size, the implicit averaging method is able to achieve the expected fitness (or effective fitness) so that the robust solution can be obtained. The second method, also known as the explicit averaging, calculates the fitness of a design by averaging over the fitness of a number randomly sampled solutions (effective fitness) in the neighborhood [68]. It should be pointed out that whether the implicit or explicit averaging strategy should be adopted depends on the property of the noise and that of the fitness landscape [69]. The third approach is to treat robustness as an additional objective, and it has been found that a trade-off between optimality and robustness often exists [70]. In that work, robustness is defined to be the variance in the objective value scaled by the average variance in the design variables, estimated from the individuals in a population. Another method for estimating the robustness is termed the inverse multi-objective robust evolutionary (IMORE) design [71], where a maximum of tolerable degradation in performance is predefined. The robustness is thus measured by the maximal allowed variation in the design variables for the given maximum performance degradation.

Search for robust solutions may require additional fitness evaluations, e.g., in the explicit averaging method and the IMORE. Since additional fitness evaluations can be very expensive, surrogates may also be adopted in estimating the robustness of a candidate design. A simple approach is to use history data near the candidate solution [68]. A more sophisticated method has been suggested in [72], where a local approximation model of the fitness function is built for estimating the expected fitness and the variance. Linear interpolation or regression and quadratic interpolation or regression models are compared with respect to the reliability in estimation, and results showed that the regression models always outperform the models using interpolation. An example of the Pareto fronts obtained by surrogates compared to the one achieved using the real fitness function (denoted by the solid line) is presented in Fig. 12.

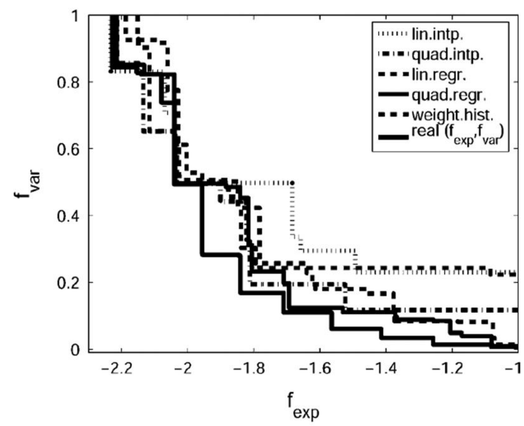

Fig. 12. Pareto fronts obtained from the real fitness function and various surrogates.

Search for robust Pareto-optimal solutions has attracted much research attention in the recent years. In [73] two approaches to finding robust Pareto-optimal solutions are suggested. The first approach, similar to SOO, is to perform the evolutionary search based on the effective objective values instead of the original ones. In the second approach, the optimization will be performed with the original objective functions, however, subject to the condition that the relative performance degradation compared to the effective fitness value or the worst one in the neighborhood is less than a predefined value.

Including the robustness measure in the objectives is less attractive in MOO in that an additional objective not only reduces the performance of an MOEA, but also makes it difficult to visualize. An interesting idea suggested in [74] is to measure the robustness of the Pareto-optimal solutions using a discrete degree according to the allowed size of neighborhood for a predefined performance degradation. One main merit of this method is that it can provide helpful and sufficient information on robustness of each Pareto-optimal solution to the user in decision-making without making the problem more complicated. The reader is referred to Part IV in [75] for more detailed discussions and recent advances in search for robust solutions in MOO.

B. Life-long optimization

While searching for robust optimal solutions is a practical and efficient strategy for tacking mild changes in the system and in the environment that can occur in the lifetime of a design, it is not sufficient for robust optimal solutions to deal with large changes. When the speed of change is relatively slow but the severity of change is large accumulated over time, life-long optimization (LLO) is necessary. In evolutionary computation, LLO is also known as dynamic optimization [76], [66]. The goal of LLO is to track the optimum, or the Pareto front that is changing over time. The change in the optimum can a result of a change in the system, or a change in the environment. For tracking a moving optimum for SOO, strategies such as maintenance of diversity in the population, introduction of memory, and maintenance of multiple sub-population (species) can be used. The idea of maintaining diversity is to prevent the population from fully converging to the current optimum and thus is able to locate a new optimum. This strategy becomes less useful for MOO because MOEAs have intrinsic population diversity. In other words, the population for MOO is supposed to be able to follow a slowly moving Pareto front. Memory mechanisms are of limited use for both SOO and for MOO when the change is not cyclic, or when the new optimum or Pareto front is weakly correlated to the old ones. The strategy of using multiple sub-populations is not attractive for MOO either. While it still makes sense to have different small populations for tracking multiple peaks, it is no longer practical to use multiple small populations to track multiple Pareto fronts. One possibility is to extend the idea described in [77] to use a sub-population to track only a segment of a moving population. Another idea is to predict the trajectory of a moving Pareto front based on history data using prediction methods so that the population can anticipate the changes in the future to track the moving Pareto front more efficiently [73], [78], [79].

VI. Spatial and Process Level Considerations

Complex structural design often involves a large number of correlated subsystems. Optimization of one single sub-system does not lead to the optimization of the whole system. For example, the optimization of gas turbine engine includes the stator blades, rotor blades, the end-wall contour etc. [80]. Another good example of complex structural design is the optimization of the aerodynamic performance of a racing car, which may involve the optimization of the diffusor, front wing, rear wing, the chassis etc. These parts are spatially distributed, however, are strongly correlated in optimization of aerodynamics such as down force and drag.

In [81], *multi-multi-objective optimization* has been discussed, which is concerned with the multi-objective optimization of a family of designs sharing common components, and thus sharing a common search subspace. The shared components are termed communal components and the design variables in the communal components are termed communal variables. It is assumed that improvement in one MOP by tuning the communal variables will lead to the deterioration of other MOPS. In that paper, a sequential approach to the multi-multi-objective optimization is proposed, where the multiple MOPs are solved sequentially, as the name of the approach suggests. The basic idea is to solve one MOP first, achieving a Pareto front. In searching for the Pareto front for the second MOP, the communal variables achieving the first Pareto solutions will be set to be constant.

In addition to the spatial-level of considerations, the process level of completeness needs to be taken into account for complex structural design. This requirement can be attributed to the fact that for many complex EMOSOs, it is difficult to define clearly the quality functions to be optimized. In this case, iterative discussions between application experts such as aerodynamic engineers and optimization experts are needed to formulate the optimization problem. During this process, knowledge from the previous optimization runs can be very helpful for both application and optimization experts to define the problem, refer to Fig. 13. In a sense, defining the problem becomes part of the problem.

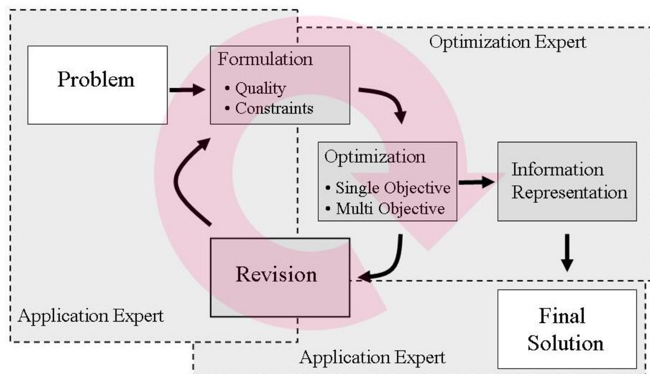

Fig. 13. A loop for formulation of the optimization problem.

VII. From A Systems Approach to Life-like Design

A systems approach to EMOSO stresses a holistic view of structural design including four levels of requirements. Compared to traditional top-down engineering design methodologies, a systems approach to EMOSO has the following features. First, it is no longer a fully top-down method. The temporal level requirements combined with an evolutionary methodology lead to more sustainable designs that a purely top-down approach cannot achieve. Nevertheless, several weaknesses still remain with a systems approach to EMOSO compared to design approach by nature, i.e., living organisms, which shows unique features such as self-assembly, self-organization, self-repair, scalability, robustness and evolvablity [82], [83]. In summary, a life-like design approach should have the following basic features.

- A life-like design adopts a developmental approach. A developmental approach starts with a simple design. Based on the pre-defined blueprint, similar to the DNA in biological systems, the system increases its functionality by interacting with the environment. The increased functionality, often in terms of increased ability to sense the environment and to act on the environment, is shaped by both the blueprint and the environment as well. In other words, for a given blueprint, the final design can be different in different environments.

  In a developmental design, not every detail of the design is specified in the blueprint, and a full global control for the developmental process is not possible. Instead, the interpretation of the blueprint depends heavily on the local interactions between the components of the system in development, as well as the interaction between the system and environment.

- A life-like design adopts an evolutionary approach. The blueprint that defines the global control of the development is not obtained by a top-down process either. Instead, the blueprint is evolved by duplicating functional modules accompanied by a strong positive selection and afterwards a strong purifying selection to generate the needed new functionality, which is known as a specialization process in biology [84].
- A life-like design is much more than search for a robust solution or life-long optimization only. Life-like design is self-organizing and self-repairing, which is mainly endowed by the developmental process driven by the blueprint-based global control and the rich interaction among the components of the system in design and with the environment. Growing when in use (growing by adding on) and developmental plasticity contribute significantly to the system's robustness to changes in the system and in the environment. On the other hand, the reuse of existing functional modules by duplication and specialization during the evolutionary process improves the efficiency and scalability [85].

From the above descriptions, it can be seen that a life-like approach to complex structural design is a large step forward from a systems approach. Yet, all requirements at the four levels of a systems approach are embodied implicitly in the design process of a life-like design.

Research results in the direction of life-like design have been reported in the recent years. One line of research was originated from computational model of artificial embryology that aims at modeling early morphological development of biological organisms [86] in computational environments [87], [88], [89], [90]. Computational models for morphological development have been used for circuit design [91], truss design [92] or stable and lightweight structural design [63], [93]. Another line of research has been initiated from robotics, where cellular mechanisms have been employed for multi-robot shape construction [94], [95] and self-reconfigurable robots [96], [97], [98]. These results indicate that genetic and cellular approaches are promising due to their self-organizing and self-repairing properties.

VIII. Summary and Conclusion

This paper discusses the main elements of a systems approach to EMOSO, which includes requirements at four levels. The first level, the system property level, is concerned with the properties of the system to be designed, such as the multiple disciplines involved in the design. In addition, the time-consuming fitness evaluations and the high-dimensional search space involved in EMOSO are also two main factors that require the MOEA should be efficient and scalable. The second level of a systems approach requires that the designer should consider performance during the whole lifetime of the design, in particular in terms of changes in the system and in the environment. A practical strategy to tackle with small changes in the system and in the environment is to look for solutions that are robust to these changes. If the severity of the change is large, a life-long optimization should be carried out. The third level of a systems approach takes the spatially distributed yet coupled sub-systems into consideration, and the fourth level considers the optimization itself as an interactive process rather than a one-shot open loop process.

A systems approach to design of complex structures is not the final goal. To better meet the requirements proposed in a systems approach is to learn from biology and make the design process more like the generation of life. Such an approach to complex system design is termed life-like design, which should be both developmental and evolutionary. An developmental and evolutionary approach, as life has been created in nature, is both self-organizing and self-repairing, evolvable and robust. Cellular models for structural design, multi-robot self-organization and self-reconfiguration could be starting points in this direction.

Acknowledgment

The authors would like to thank their colleagues, M. Olhofer, M. Hasenjäger, S. Menzel, L. Graening, T. Steiner, and collaborators, Y.-S. Ong, Q. Zhang, Y. Cao, Y. Meng, and X. Yao for their contribution to the work described in this paper.

References

[1] C. Coello Coello and G.B. Lamont, Ed., *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific, 2004.
[2] Y. Jin, Ed., *Multi-objective Machine Learning*. Berlin Heidelberg: Springer, 2006.
[3] R. Haftka and Z. Gürdal, *Elements of Structural Optimization*, 3rd ed. Springer, 1992.
[4] J. Sobieszczanski-Sobieski and R. Haftka, "Multidisciplinary aerospace design optimization: Survey of recent developments," *Structural Optimization*, vol. 14, pp. 1–23, 1997.
[5] D. Fogel, *Evolutionary Computation*. Piscataway, NJ: IEEE Press, 1995.
[6] Y. Jin, "Rethinking multi-objective evolutionary algorithms," Dagstuhl Seminar on Theory of Evolutionary Algorithms, February 2005.
[7] H. Ishibuchi and K. Narukawa, "Recombination of similar parents in emo algorithms," in *EMO 2005*, ser. LNCS 3410. Springer, 2005, pp. 265–279.
[8] T. Okabe, Y. Jin, and B. Sendhoff, "On the dynamics of multiobjective optimization," in *Genetic and Evolutionary Computation Conference,*, 2002, pp. 247–256.
[9] Y. Jin and B. Sendhoff., "Connectedness, regularity and the success of local search in evolutionary multi-objective optimization," in *Congress on Evolutionary Computation*, vol. 3. IEEE Press, 2003, pp. 1910–1917.
[10] S. Ciliberti, O. Martin, and A. Wagner, "Innovation and robustness in complex regulatory gene networks," *Proceedings of the National Academy of Sciences*, vol. 104, no. 34, pp. 13 591–13 596, 2007.
[11] F. Rothlauf, *Representations for Genetic and Evolutionary Algorithms*. Springer, 2006.

[12] A. Oyama, S. Obayashi, and T. Nakamura, "Real-coded adaptive range genetic algorithms applied to transonic wing optimization," in *Parallel Problem Solving from Nature*. Springer, 2000.

[13] L. Piegl and W. Tiller, *The NURBS Book*. Springer, 1997.

[14] D. Poueymirou-Bouchet, C. Tribes, and J.-Y. Trepanier, "A nurbs-based shape optimization method for hydraulic turbine stay vane," in *Computational Fluid Dynamics*. Springer, 2004, pp. 415–421.

[15] M. Hasenjäger, B. Sendhoff, T. Sonoda, and T. Arima, "Three dimensional evolutionary aerodynamic design optimization with CMA-ES," in *GECCO 2005: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, H.-G. B. et al., Ed. New York, NY: ACM Press, 2005, pp. 2173–2180.

[16] E. Perry, S. Benzley, M. Landon, and R. Johnson, "Shape optimization of fluid flow systems," in *AMSE Fluids Engineering Summer Conference*, 2000.

[17] S. Menzel, M. Olhofer, and B. Sendhoff, "Evolutionary design optimisation of complex systems integrating fluent for parallel flow evaluation," in *Proceedings of the 2nd EACC - 2005 European Automotive CFD Conference*, K. Seibert and R. Hanna, Eds. Frankfurt: FLUENT Inc., 2005, pp. 279–289.

[18] ——, "Direct manipulation of free form deformation in evolutionary design optimization," in *Parallel Problem Solving from Nature*, ser. LNCS 4193. Springer, 2006, pp. 352–361.

[19] S. Menzel and B. Sendhoff, "Representing the change - free form deformation for evolutionary design optimisation," in *Evolutionary Computation in Practice*, T. Yu, D. Davis, C. Baydar, and R. Roy, Eds. Berlin: Springer, 2008, ch. 4, pp. 63–86.

[20] K. Foli, T. Okabe, M. Olhofer, Y. Jin, and B. Sendhoff, "Optimization of micro heat exchanger: Cfd, analytical approaches and multi-objective evolutionary algorithms," *International Journal of Heat and Mass Transfer*, vol. 49, no. 5-6, pp. 1090–1099, 2006.

[21] M. Olhofer, Y. Jin, and B. Sendhoff, "Adaptive encoding for aerodynamic shape optimization using evolution strategies," in *Congress on Evolutionary Computation*, vol. 1. IEEE Press, 2001, pp. 576–583.

[22] P. Zhang, X. Yao, L. Ja, B. Sendhoff, and T. Schnier, "Target shape design optimization by evolving splines," in *Congress on Evolutionary Computation*. IEEE Press, 2007, pp. 2009–2016.

[23] Y. Jin, M. Olhofer, and B. Sendhoff, "On evolutionary optimization of large problems with small populations," in *Int. Conf. on Natural Computation*, ser. LNCS 3611. Springer, 2005, pp. 1145–1154.

[24] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing*, vol. 9, no. 1, pp. 3–12, 2005.

[25] K. Rasheed and H. Hirsh, "Informed operators: Speeding up genetic-algorithm-based design optimization using reduced models," in *Genetic and Evolutionary Computation Conference*, 2000, pp. 628–635.

[26] Y. S. Ong, P. Nair, and A. Keane, "Evolutionary optimization of computationally expensive problems via surrogate modeling," *American Institute of Aeronautics and Astronautics Journal*, vol. 41, no. 4, pp. 687–696, 2003.

[27] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 481–494, 2002.

[28] J. Branke and C. Schmidt, "Faster convergence by means of fitness estimation," *Soft Computing*, vol. 9, no. 1, pp. 13–20, 2005.

[29] M. Emmerich, A. Giotis, M. zdemir, T. Bäeck, and K. Giannakoglou, "Metamodel-assisted evolution strategies," in *Parallel Problem Solving from Nature*, ser. LNCS 2439, 2002, pp. 361–370.

[30] Y. Jin, M. Olhofer, and B. Sendhoff, "On evolutionary optimization with approximate fitness functions," in *Genetic and Evolutionary Computation Conference*, 2000, pp. 786–793.

[31] L. Bull, "On model-based evolutionary computation," *Soft Computing*, vol. 3, no. 2, pp. 76–82, 1999.

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

[33] M. Sefrioui and J. Periaux, "A hierarchical genetic algorithm using multiple models for optimization," in *Parallel Problem Solving from Nature*, ser. LNCS 1917, vol. 1917. Springer, 2000, pp. 879–888.

[34] R. Reed and R. Marks II, *Neural Smithing*. Cambridge, MA: The MIT Press, 1999.

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

[36] M. Huesken, Y. Jin, and B. Sendhoff, "Structure optimization of neural networks for evolutionary design optimization," *Soft Computing*, vol. 9, no. 1, pp. 21–28, 2005.

[37] L. Gräning, Y. Jin, and B. Sendhoff, "Individual-based management of meta-models for evolutionary optimization with application to three-dimensional blade optimization," in *Evolutionary Computation in Dynamic and Uncertain Environments*, ser. Studies in Computational Intelligence, S. Yang, Y.-S. Ong, and Y. Jin, Eds. Springer, 2007, pp. 225–250.

[38] Y. Jin and B. Sendhoff, "Reducing fitness evaluations using clustering techniques and neural network ensembles," in *Genetic and Evolutionary Computation Conference.*, ser. LNCS 3102. Springer, 2004, pp. 688–699.

[39] T. Goel, R. Haftka, W.Shyy, and N. Queipo, "Ensemble of surrogates," *Structural Multidisciplinary Optimization*, vol. 33, pp. 199–216, 2007.

[40] D. Marjavaara, S. Lundström, and W. Shyy, "Hydraulic turbine diffuser shape optimization by multiple surrogate model approximations of Pareto fronts," *AMSE Journal of Fluids Engineering*, vol. 129, pp. 1228–1240, 2007.

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

[42] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Transactions on Evolutionary Computation*, 2009, in press.

[43] Y. Cao, Y. Jin, M. Kowalczykiewicz, and 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.

[44] H. Takagi, "Interactive evolutionary computation: Fusion of the capacities of ec optimization and human evaluation," *Proceesings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.

[45] W. Spears, "The role of mutation and recombination in evolutionary algorithms," Ph.D. dissertation, George Mason University, Fairfax, Virginia, 1998.

[46] T. Okabe, Y. Jin, and B. Sendhoff, "Evolutionary multi-objective optimization with a hybrid representation," in *Congress on Evolutionary Computation*, vol. 4, 2003, pp. 2262–2269.

[47] R. Enache, B. Sendhoff, M. Olhofer, and M. Hasenjäger, "Comparison of steady-state and generational evolution strategies for parallel architectures," in *Parallel Problem Solving from Nature – PPSN VIII*, ser. Lecture Notes in Computer Science 3242, X. Yao *et al.*, Eds. Springer, 2004, pp. 253–262.

[48] D. Lim, Y.-S. Ong, Y. Jin, B. Sendhoff, and B.-S. Lee, "Efficient hierarchical parallel genetic algorithms using grid computing," *Future Generation Computer Systems – The International Journal of Grid Computing: Theory, Methods and Applications*, vol. 23, no. 4, pp. 658–670, 2007.

[49] L. Graening, S. Menzel, M. Hasenjäger, T. Bihrer, M. Olhofer, and B. Sendhoff, "Knowledge extraction from aerodynamic design data and its application to 3d turbine blade geometries," *Journal of Mathematical Modelling and Algorithms*, vol. 7, no. 4, pp. 329–350, 2008.

[50] I. Parmee, "The development of a dual-agent strategy for efficient search across whole system engineering design hierachies," in *Parallel Problem Solving from Nature*. Springer, 1996, pp. 523–532.

[51] ——, *Evolutionary and Adaptive Computing in Engineering Design*. London: Springer, 2001.

[52] S. Obayashi and D. Sasaki, "Visualization and data mining of Pareto solutions using self-organizing map," in *Evolutionary Multi-Criterion Optimization*, ser. LNCS 2632, 2003, pp. 796–809.

[53] Y. Jin, Ed., *Knowledge Incorporation in Evolutionary Algorithms*. Berlin Heidelberg: Springer, 2005.

[54] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Trans. Evolutionary Computation*, vol. 7, no. 2, pp. 204–223, 2003.

[55] Y. Jin, M. Olhofer, and B. Sendhoff, "Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how?" in *Genetic and Evolutionary Computation Conference*. Morgan Kaufmann, 2001, pp. 1042–1049.

[56] L. Paquete and T. Stüzle, "A two-phase local search for the biobjective traveling salesman problem," in *Evolutionary Multi-Criterion Optimization*, ser. LNCS 2632. Springer, 2003, pp. 479–493.

[57] P. Larrañaga and J. A. Lozano, Eds., *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2002.

[58] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model based multi-objective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.

[59] Y. Jin, A. Zhou, Q. Zhang, B. Sendhoff, and E. Tsang, "Modeling regularity to improve scalability of model-based multi-objective optimization algorithms," in *Multi-Objective Problem Solving from Nature*, J. Knowles, D. Corne, and K. Deb, Eds. Springer, 2008, pp. 331–356.

[60] D. A. van Veldhuizen and G. B. Lamont, "Evolutionary computation and convergence to a Pareto front," in *Late Breaking Papers at the Genetic Programming Conference*. Madison, Wisconsin, USA: Stanford University Bookstore, July 1998, pp. 221–228.

[61] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Congress on Evolutionary Computation*, 2008, pp. 2419–2426.

[62] S. Harding and J. Miller, "The dead state: A comparison between developmental and direct encodings," in *GECCO Workshop on Complexity through Development and Self-Organizing Representations*, 2006.

[63] T. Steiner, Y. Jin, and B. Sendhoff, "A cellular model for evolutionary development of lightweight materials with an inner structure," in *Genetic and Evolutionary Computation Conference*, 2008, pp. 851–858.

[64] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Parallel Problem Solving from Nature*, vol. VI, 2000, pp. 849–858.

[65] H.-G. Beyer and B. Sendhoff, "Robust optimization - A comprehensive survey," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 33-34, pp. 3190–3218, 2007.

[66] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments - A survey." *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.

[67] S. Tsutsui and A. Ghosh, "Genetic algorithms with a robust solution searching scheme," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 3, pp. 201–208, 1997.

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

[69] H.-G. Beyer and B. Sendhoff, "Evolutionary algorithms in the presence of noise: To sample or not to sample?" in *IEEE Symposium on Foundations of Computational Intelligence*, 2007, pp. 17–24.

[70] Y. Jin and B. Sendhoff, "Trade-off between performance and robustness: An evolutionary multiobjective approach." in *Evolutionary Multi-Criterion Optimization*, ser. LNCS 2632. Springer, 2003, pp. 237–251.

[71] D. Lim, Y.-S. Ong, Y. Jin, B. Sendhoff, and B. S. Lee, "Inverse multi-objective robust evolutionary optimization," *Genetic Programming and Evolvable Machines*, vol. 7, no. 4, pp. 383–404, 2007.

[72] I. Paenke, J. Branke, and Y. Jin, "Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 405–420, 2006.

[73] K. Deb, U. B. Rao, and S. Karthik, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermol power scheduling," in *Evolutionary Multi-Criterion Optimization*, ser. LNCS 4403. Springer, 2007, pp. 803–817.

[74] C. Barrico and C. Antunes, "Robustness analysis in multi-objective optimization using a degree of robustness concept," in *Congress on Evolutionary Computation*, 2006, pp. 6778–6783.

[75] S. Yang, Y.-S. Ong, and Y. Jin, Eds., *Evolutionary Computation in Dynamic and Uncertain Environments*. Springer, 2007.

[76] J. Branke, *Evolutionary Optimization in Dynamic Environments*. Kluwer, 2001.

[77] K. Deb, P. Zope, and A. Jain, "Distributed computing of pareto-optimal solutions with evolutionary algorithms," in *Evolutionary Multi-Criterion Optimization*, ser. LNCS 2632. Springer, 2003, pp. 534–549.

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

[79] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization," in *Evolutionary Multi-Criterion Optimization*, ser. LNCS 4403. Springer, 2007, pp. 832–846.

[80] T. Sonoda, M. Hasenjger, T. Arima, and B. Sendhoff, "Effect of endwall contouring on performance of ultra-low aspect ratio transonic turbine inlet guide vanes," *Journal of Turbomachinery*, vol. 131, no. 1, 2009.

[81] G. Avigad, "multi-multi-objective optimization problem and its solution by a moea," in *Evolutionary Multi-Criterion Optimization*, ser. LNCS 4403. Springer, 2007, pp. 847–861.

[82] G. Whitesides and B. Grzybowski, "Self-assembly at all scales," *Science*, vol. 295, pp. 2418–2421, 2002.

[83] K. Ariga, J. Hill, M. Lee, A. Vinu, R. Charvet, and S. Acharya, "Challenges and breakthroughs in recent research on self-assembly," *Sci. Technol. Adv. Mater.*, vol. 9, no. 014109, 2008.

[84] J. Zhang, "Evolution by gene duplication: an update," *Trends in Ecology and Evolution*, vol. 18, no. 6, pp. 292–298, 2003.

[85] D. Roggen and D. Federici, "Multi-cellular development: Is there scalability and robustness to gain?" in *Parallel Problem Solving from Nature*. Springer, 2004, pp. 391–400.

[86] S. Gilbert, *Developmental Biology*. Sunderland, MA: Sinauer Assocaites Inc., 2003.

[87] R. Doursat, "The self-made puzzle: Integrating self-assembly and pattern formation under non-random genetic regulation," *InterJournal Complex Systems*, vol. 2292, 2008.

[88] S. Kumar and P. Bentley, Eds., *On Growth, Form, and Computers*. Amsterdam: Elsevier, 2003.

[89] U.-M. O'Reilly and M. Hemberg, "Integrating generative growth and evolutionary computation for form exploration," *Genetic Programming and Evolvable Machines*, vol. 8, no. 2, pp. 163–186, 2007.

[90] K. Stanley and R. Miikkulainen, "A taxonomy for artificial embryogeny," *Artificial Life*, vol. 9, no. 2, pp. 93–130, 2003.

[91] S. Zhan, J. F. Miller, and A. M. Tyrrell, "An evolutionary system using development and artificial genetic regulatory networks," in *Congress on Evolutionary Computation*. IEEE Press, 2008, pp. 815–822.

[92] T. Kowaliw, P. Grogono, and N. Kharma, "The evolution of structural design through artificial embryogeny," in *IEEE Symposium on Artificial Life*, 2007, pp. 425–432.

[93] T. Steiner, J. Trommler, M. Brenn, Y. Jin, and B. Sendhoff, "Global shape with morphogen gradients and motile polarized cells," in *Congress on Evolutionary Computation*. IEEE Press, 2009, accepted.

[94] M. Mamei, M. Vasirani, and F. Zambonelli, "Experiments in morphogenesis in swarms of simple mobile robots," *Applied Artificial Intelligence*, vol. 18, no. 9-10, pp. 903–919,, 2004.

[95] H. Guo, Y. Meng, and Y. Jin, "Self-adaptive multi-robot construction using gene regulatory networks," in *IEEE Symposium on Artificial Life*. IEEE Press, 2009, pp. 53–60.

[96] S. Murata and H. Kurokawa, "Self-reconfigurable robots: Shape-changing cellular robots can exceed conventional robot flexibility," *IEEE Robotics & Automation Magazine*, pp. 71–78, 2007.

[97] J. Cheng, W. Cheng, and R. Nagpal, "Robust and self-reparing formation control for swarms of mobile agents," in *Twentieth National Conference on Artificial Intelligence*. AAAI Press, 2005, pp. 59–64.

[98] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. Chirikjian, "Modular self-reconfigurable robot systems:Challenges and opportunities for the future," *IEEE Robotics & Automation Magazine*, pp. 2–11, March 2007.