Swarm and Evolutionary Computation 1 (2011) 32–49

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Survey paper

Multiobjective evolutionary algorithms: A survey of the state of the art

Aimin Zhou [a,*], Bo-Yang Qu [b], Hui Li [c], Shi-Zheng Zhao [b], Ponnuthurai Nagaratnam Suganthan [b], Qingfu Zhang [d]

[a] Computer Science and Technology Department, East China Normal University, Shanghai, 200241, China
[b] School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore
[c] School of Science, Xian Jiaotong University, Xian, 710049, China
[d] School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, UK

ARTICLE INFO

Article history:
Received 1 February 2011
Received in revised form
2 March 2011
Accepted 5 March 2011
Available online 16 March 2011

Keywords:
Multiobjective optimization
Evolutionary multiobjective optimization
Multiobjective evolutionary algorithms
Multicriteria decision making

ABSTRACT

A multiobjective optimization problem involves several conflicting objectives and has a set of Pareto optimal solutions. By evolving a population of solutions, multiobjective evolutionary algorithms (MOEAs) are able to approximate the Pareto optimal set in a single run. MOEAs have attracted a lot of research effort during the last 20 years, and they are still one of the hottest research areas in the field of evolutionary computation. This paper surveys the development of MOEAs primarily during the last eight years. It covers algorithmic frameworks such as decomposition-based MOEAs (MOEA/Ds), memetic MOEAs, coevolutionary MOEAs, selection and offspring reproduction operators, MOEAs with specific search methods, MOEAs for multimodal problems, constraint handling and MOEAs, computationally expensive multiobjective optimization problems (MOPs), dynamic MOPs, noisy MOPs, combinatorial and discrete MOPs, benchmark problems, performance indicators, and applications. In addition, some future research issues are also presented.

1. Introduction

Many real-world optimization problems involve multiple objectives. A *multiobjective optimization problem* (MOP) can be mathematically formulated as

$$\begin{aligned} \text{minimize} \quad & F(x) = (f_1(x), \ldots, f_m(x))^T \\ \text{s.t.} \quad & x \in \Omega, \end{aligned} \tag{1}$$

where Ω is the decision space and $x \in \Omega$ is a decision vector. $F(x)$ consists of m objective functions $f_i : \Omega \to R,\ i = 1, \ldots, m$, where R^m is the objective space.

The objectives in (1) often conflict with each other. Improvement of one objective may lead to deterioration of another. Thus, a single solution, which can optimize all objectives simultaneously, does not exist. Instead, the best trade-off solutions, called the *Pareto optimal solutions*, are important to a *decision maker* (DM). The Pareto optimality concept, which was first proposed by Edgeworth and Pareto [1], is formally defined as follows [2,3].

Definition 1. A vector $u = (u_1, \ldots, u_m)^T$ is said to *dominate* another vector $v = (v_1, \ldots, v_m)^T$, denoted as $u \prec v$, iff $\forall i \in \{1, \ldots, m\}$, $u_i \leq v_i$ and $u \neq v$.

Definition 2. A feasible solution $x^* \in \Omega$ of problem (1) is called a *Pareto optimal solution*, iff $\nexists y \in \Omega$ such that $F(y) \prec F(x^*)$. The set of all the Pareto optimal solutions is called the *Pareto set* (PS), denoted as

$$PS = \{x \in \Omega | \nexists y \in \Omega, F(y) \prec F(x)\}.$$

The image of the PS in the objective space is called the *Pareto front* (PF)

$$PF = \{F(x) | x \in PS\}.$$

Due to their population-based nature, *evolutionary algorithms* (EAs) are able to approximate the whole PS (PF) of an MOP in a single run. There has been a growing interest in applying EAs to deal with MOPs since Schaffer's seminal work [4], and these EAs are called *multiobjective evolutionary algorithms* (MOEAs). By January 2011, more than 5600[1] publications have been published

* Corresponding author.
E-mail addresses: amzhou@cs.ecnu.edu.cn (A. Zhou), e070088@ntu.edu.sg (B.-Y. Qu), lihui10@mail.xjtu.edu.cn (H. Li), zh0047ng@ntu.edu.sg (S.-Z. Zhao), epnsugan@ntu.edu.sg (P.N. Suganthan), qzhang@essex.ac.uk (Q. Zhang).

[1] The statistical data is based on the paper repository in the EMOO web site, http://delta.cs.cinvestav.mx/~ccoello/EMOO/, which is maintained by Professor Coello Coello.

2210-6502/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.swevo.2011.03.001

on evolutionary multiobjective optimization. Among these papers, 66.8% have been published in the last eight years, 38.4% are journal papers and 42.2% are conference papers. The research work on MOEAs has been surveyed from different aspects. Among these surveys, some are mainly on generic methodologies [5–12]; some are on theoretical developments and applications [13,14]; some work focus on special methods for MOPs, for example *simulated annealing* (SA) [15], *particle swarm optimization* (PSO) [16], and memetic algorithms [17]; some are on combinational problems [18,19]; and others are on special applications, such as engineering problems [14,20,21], scheduling problems [22], economic and finance problems [23], automatic cell planning problems [24], traveling salesman problems [25], and preferences in MOPs [26]. However, no comprehensive survey has been conducted on MOEA development in recent years [6].

In this paper, we focus on recent developments on MOEAs. Our major concern is on continuous MOPs, while the works on combinational MOPs are covered in [19]. The remainder of this paper is organized as follows. Section 2 summarizes the advances in generic MOEA designs. Algorithm frameworks, selection strategies, and offspring reproduction operators are surveyed in this section. In Section 3, MOEAs for some complicated problems, such as constrained MOPs, multimodal problems, many-objective problems, expensive MOPs, and dynamic and noisy MOPs, are outlined. The benchmark problems and algorithm performance measures are surveyed in Section 4. Section 5 briefly discusses the applications of MOEAs. Finally, the paper is concluded in Section 6 with some potential directions for future research.

2. Advances in MOEA design

In this section, recent developments, including algorithm frameworks, selection and population updating strategies, offspring reproduction schemes, and other related issues, are surveyed.

2.1. Algorithm frameworks

The algorithm framework is a key issue to design an MOEA. A majority of MOEAs in both the research and the application areas share more or less the same framework as that of non-dominated sorting genetic algorithm II (NSGA-II) [27]: a selection operator based on Pareto domination and a reproduction operator are used iteratively. In this section, we introduce some frameworks which are different from that of NSGA-II.

2.1.1. An MOEA based on decomposition: MOEA/D

A multiobjective evolutionary algorithm based on decomposition (MOEA/D) [28] is a recent multiobjective evolutionary algorithmic framework. It is based on conventional aggregation approaches in which an MOP is decomposed into a number of *scalar objective optimization problems* (SOPs). The objective of each SOP, also called a subproblem, is a (linearly or nonlinearly) weighted aggregation of the individual objectives. Neighborhood relations among these subproblems are defined based on the distances between their aggregation weight vectors. Subproblem i is a neighbor of subproblem j if the weight vector of subproblem i is close to that of subproblem j. Each subproblem is optimized in the MOEA/D by using information mainly from its neighboring subproblems.

In a simple version of the MOEA/D, each individual subproblem keeps one solution in its memory, which could be the best solution found so far for the subproblem. For each subproblem, the algorithm generates a new solution by performing genetic operators on several solutions from its neighboring subproblems, and updates its memory if the new solution is better than old one for the subproblem. A subproblem also passes its newly generated solution on to some (or all) of its neighboring subproblems, which will update their current solutions if the received solution is better. A major advantage of MOEA/Ds is that a scalar objective local search can be used in each subproblem in a natural way since its task is for optimizing a scalar objective subproblem.

Several improvements on MOEA/Ds have been made recently. Li and Zhang [29] suggested using two different neighborhood structures for balancing exploitation and exploration. Zhang et al. [30] proposed a scheme for dynamically allocating computational efforts to different subproblems in an MOEA/D in order to reduce the overall cost and improve the algorithm performance. This implementation of MOEA/D is efficient and effective and has won the *Congress on Evolutionary Computation (CEC)* 2009 MOEA competition [31]. Nebro and Durillo [32] developed a thread-based parallel version of MOEA/D, which can be executed on multicore computers. Palmers et al. [33] proposed an implementation of MOEA/D in which each subproblem records more than one solution. Ishibuchi et al. [34] proposed using different aggregation functions at different search stages. MOEA/Ds have been successfully applied to a number of application areas [33,35–42].

2.1.2. MOEAs based on preference

Due to the conflicts among the objectives in MOPs, the total number of Pareto optimal solutions might be very large or even infinite. However, the DM may be only interested in preferred solutions instead of all Pareto optimal solutions. To find the preferred solutions, the preference information is needed to guide the search towards the region of the PF of interest to the DM. Based on the role of the DM in the solution process, multiobjective optimization methods can be classified into priori methods, posteriori methods, and interactive methods [2].

In a priori method, preference information is given by the DM before the solution process. An MOP can be converted into an SOP. Then, a scalar objective solver is applied to find the desired Pareto optimal solution. A posteriori method uses the DM's preference information after the search process. A well-distributed approximation of the PF is first obtained. Then, the DM selects the most preferred solutions based on the preferences. In an interactive method, the intermediate search results are presented to the DM to investigate; then the DM can understand the problem better and provide more preference information for guiding the search.

The earliest attempts on MOEAs based on the DM's preference were made by Fonseca and Fleming [43] and Tanino et al. [44] in 1993. In these algorithms, the rank of the members of a population is determined by both the Pareto dominance and the preference information from the DM. In [45], Greenwood et al. used value functions to rank the population, and preference information was also used in the survival criteria.

Sakawa and Kato [46] used a fuzzy approach to represent preference in the form of reference points. The DM is asked to specify a new reference point until satisfactory results are reached. Phelps and Köksalan [47] compared a pair of individuals in terms of their fitness values based on the DM's preferences at each iteration. A single substitute objective defined by weighted sum of objectives is used for some generations.

In [48], Branke and Deb incorporated the preference information into NSGA-II by modifying the definition of dominance and using a biased crowding distance based on weights. Deb et al. [49] further considered the use of reference points to determine preference information. A guided dominance scheme and a biased crowding scheme are also suggested. In [50], Deb et al. suggested an interactive MOEA based on reference directions. The DM provides one or more reference directions to guide the search towards the region of preferred solution.

Deb and Chaudhuri [51] proposed an interactive decision support system called I-MODE, in which a number of existing multiobjective optimization and classical decision-making methods can be appropriately adopted for generating solutions in the regions of interest in the PS. For example, the weighted sum approach and weighted Tchebycheff approach may be used to deal with nonconvexity of the PF.

Li and Silva [52] developed an improved version of an MOEA/D combined with SA. In this version, the weights can be adaptively changed by the DM according to the location of solutions in the current population. The fitness functions with modified weights can guide the search towards different parts of the PF during the search. It can be viewed as an interactive MOEA.

Sanchis et al. [53] proposed an MOEA integrated with priori preferences, which were generated by applying the principle of physical programming. In this algorithm, the preferences are expressed by partitioning the objective space into several levels. The preference functions are built to reflect the DM's interests and to use meaningful parameters for each objective. The designer's expert knowledge can be translated into preferences for design objectives. A scalar objective is automatically built and no weight selection is performed.

In [54], Deb et al. proposed a progressively interactive MOEA. In this method, an approximate value function is progressively generated after every few generations. Periodically, several non-dominated points found so far are provided to the DM. Based on the DM's preference information, all these points are ranked from the worst to the best. Then, a suitable polynomial value function is constructed by solving an SOP.

In [55], Rachmawati and Srinivasan proposed a preference-based MOEA to find the knee region in the PF, which is visually a convex bulge in the front. The preference-based focus is achieved by optimizing a set of linear weighted sums of the original objectives, and control of the extent of the focus is attained by careful selection of the weight set based on a user-specified parameter. The fitness scheme could be easily adopted in any Pareto-based MOEA with little additional computational cost.

Thiele et al. [56] used the DM's preferences expressed interactively in the form of reference points. The information is used in an EA to generate a new population by combining the fitness function and an achievement scalarization function. The selection based on the utility functions with the modified parameters is expected to lead the search to focus on the most interesting parts of the PS. In multiobjective optimization, achievement scalarization functions are widely used to project a given reference point on to the PS.

2.1.3. Indicator-based MOEAs

The quality of an approximated PF could be measured by a scalar indicator such as generational distance and hypervolume. Indicator-based MOEAs use an indicator to guide the search, particularly to perform solution selection.

Zitzler and Künzli [57] first suggested a general *indicator-based evolutionary algorithm* (IBEA). This approach uses an arbitrary indicator to compare a pair of candidate solutions. In the IBEA, any additional diversity preservation mechanism such as fitness sharing, is no longer required. In comparison to other MOEAs, the IBEA only compares pairs of individuals instead of entire approximation sets.

In [58], Basseur and Zitzler proposed an indicator-based model for handling uncertainty, in which each solution is assigned a probability in the objective space. In an uncertain environment, some methods for computing expected indicator values are discussed, and several variants of their ϵ-indicator-based model are suggested and empirically investigated.

Brockhoff and Zitzler [59] proposed a general approach to incorporate objective reduction techniques into hypervolume-based algorithms. Different objective reduction strategies are studied for improving the performance of hypervolume-based MOEAs.

In [60], Bader and Zitzler suggested a fast hypervolume-based MOEA for many-objective optimization. To reduce the computational overhead in hypervolume computation, a fast method based on Monte Carlo simulations is proposed to estimate the hypervolume value of an approximation set. Therefore, the proposed hypervolume-based MOEA may be applied to problems with many objectives.

Very recently, Bader and Zitzler [61] further investigated the robustness of hypervolume-based multiobjective search methods. Three existing approaches for handling robustness in the area of evolutionary computing, modifying the objective functions, additional objectives, and additional robustness constraints, are integrated into a multiobjective hypervolume-based search. An extension of the hypervolume indicator is also proposed for robust multiobjective optimization.

2.1.4. Hybrid MOEAs

In MOEAs, there are many techniques which have different characteristics and advantages. Hybridizing these techniques is thus a natural choice to utilize their advantages for dealing with complicated MOPs. What techniques to use and how to hybridize them are two major problems to solve when designing a hybrid MOEA. Some recent work could thus be categorized as follows.

Hybridizing different search methods: A general idea is to combine global search and local search methods, known as the memetic approach, elaborated in Section 2.1.5. Another widely used idea is to combine the search operators of different algorithms. PSO and EA are hybridized in [62]. In each generation, the solutions generated by a PSO (EA) operator are then improved by an EA (PSO) operator. In [63], quantum operators are applied to solutions in binary representation and a genetic operator is then applied to the good solutions in permutation representation.

Hybridizing search and updating methods: This strategy hybridizes different components from different algorithms. For example, in [62], the PSO's operator is inserted into an EA's main loop.

Hybridizing different methods in different search phases: In the above two strategies, the hybrid methods are used in each generation. It is also natural to partition a search process into different phases and to use different search strategies in these phases. For example, in [64], the search is partitioned into three phases to emphasize dominated solutions, to balance dominated and non-dominated solutions, and to focus on non-dominated solutions, respectively. NSGA-II and a local incremental search algorithm are used to achieve the goals.

2.1.5. Memetic MOEAs

As a special case of hybrid MOEAs, MOEAs incorporating local search methods have also been investigated recently [65–73]. These algorithms are known as memetic MOEAs. Memetic algorithms are able to offer not only better speed of convergence to the evolutionary approach, but also better accuracy for the final solutions [65]. Ishibuchi and Murata proposed one of the first memetic MOEAs [66]. The algorithm uses a local search method after classical variation operators are applied, and a randomly drawn scalar function to assign fitness is used for parent selection.

In [74], the best solutions found in each generation are improved by a local search method in the objective space, and the improved solutions are then mapped back to the decision space to predict the corresponding decision variables. A local search operator is used to generate offspring solutions [75]. Similar ideas are also mentioned in [76,77].

In [78], Knowles and Corne proposed a memetic Pareto archived evolution strategy to solve MOPs. The algorithm introduces a Pareto ranking-based selection method and couples it with a partition scheme in objective space. It uses two different archives to save non-dominated solutions.

Jaszkiewicz [79] proposed a *multiobjective genetic local search* (MOGLS) algorithm for the multiobjective 0/1 knapsack problem. At each iteration, a weighted scalarization function is used as the fitness function during selection. The weights are generated in a random way. The mating population in the MOGLS consists of a few individuals selected from the current population in terms of the current scalarization function. An offspring solution is then produced by recombining members in the mating population. A local search procedure is followed to improve the quality of the offspring solution. The current population and an external population including only non-dominated solutions are updated by the improved solutions obtained in the local search.

In [80], Caponio and Neri proposed the cross dominant multiobjective memetic algorithm, making use of two local search engines to balance the global search and the local search. The choice of local search engines is decided by using the parameter of mutual dominance between non-dominated solutions belonging to consecutive generations. A memetic version of coevolutionary multiobjective *differential evolution* (DE) is presented in [81]. In this approach, the population of solutions and promising search directions are evolved synchronously. A local search method is applied to a portion of the population after each iteration.

In [68], a *memetic algorithm based on differential evolution* (MADE) was proposed by Qian et al. to handle multiobjective no-wait flow-shop scheduling problems (MNFSSPs). This algorithm uses several local searchers developed according to the landscape of an MNFSSP to enhance the local exploitation.

Wanner et al. [72] employed a local search optimizer as an additional operator in multiobjective evolutionary techniques. The local search technique is able to find more precise estimation of the Pareto optimal surface with a reduced number of function evaluations. In [73], Ishibuchi et al. studied the use of biased neighborhood structures for a local search in multiobjective memetic algorithms. The methods assign higher probabilities to more promising neighbors in order to improve the search ability of multiobjective memetic algorithms. More recently, Lara et al. [65] investigated a new local search strategy called the hill climber with sidestep (HCS) for multiobjective memetic algorithms. The new point-wise local search procedure is able to move both toward (using hill climber techniques) and along (sidestep) the PS.

MOEA/D [28] also belongs to the class of multiobjective memetic algorithms. It optimizes multiple subproblems. Each solution is associated with one weighted scalarization function. A local search procedure can be called for improving a solution. Since MOEA/D is a general framework, different heuristic search methods can serve as the local search component. In [52], SA is used to improve the current solution of each subproblem. In [82], each subproblem is optimized by the *greedy randomized adaptive search procedure* (GRASP).

2.1.6. MOEAs based on coevolution

Coevolution can be regarded as evolving multiple subpopulations simultaneously to tackle a complicated problem. Algorithms using an archive strategy, such as [83], thus fall into this category because they evolve a population and an archive at the same time to approximate the PF of an MOP.

However, there is another explanation of coevolution by using the idea of divide and conquer. Following this idea, a coevolutionary algorithm breaks down a problem into a set of subproblems in the level of individual coding and evolves multiple subpopulations. A variety of papers adopt this idea [84–86]. Among them, the subpopulations are competitive and/or cooperative with each other and the components from different subpopulations are combined to form a complete solution.

2.2. Selection and population updating

The selection of solutions for the next generation plays a key role in an MOEA. The main difference between EAs for SOPs and MOPs in algorithm components is the selection procedure. An EA for SOPs can be directly applied to MOPs by replacing the selection component. In scalar objective optimization, there naturally exists a complete order to differentiate all feasible solutions, i.e., for any two feasible solutions x and y, either $f(x) \leq f(y)$ or $f(y) \leq f(x)$. However, in multiobjective optimization, the Pareto dominance, $\prec$, only defines a partial order in the objective space, and not all the feasible solutions can be compared to each other.

Since the Pareto dominance cannot be naturally used to select solutions, additional strategies need to be considered. The design of selection operators has been gaining significant attention in evolutionary multiobjective optimization. The previous major works on selection follow the idea of defining complete orders over individuals, and recently some works follow the idea of defining complete orders over populations.

2.2.1. Complete orders over individuals

Since Pareto domination only defines a partial order, extending the partial order to a complete order becomes a natural way to differentiate solutions. To this end, a two-stage strategy is usually employed. In the first stage, a population is partitioned into several clusters by Pareto dominance. Each individual x will be assigned an integer value, called the *rank*, and denoted as x^{rnk}. Those with the same rank value are equal to each other, and smaller rank is preferred. In the second stage, individuals with the same rank are further differentiated by assigning each individual a real value, called the *density*, and denoted as x^{den}. Those with lower density values are preferred. A complete order, denoted as $\prec_i$, can thus be defined as follows:

$$x \prec_i y, \quad \text{iff } (x^{\mathrm{rnk}} < y^{\mathrm{rnk}}), \text{ or } (x^{\mathrm{rnk}} = y^{\mathrm{rnk}} \text{ and } x^{\mathrm{den}} < y^{\mathrm{den}}).$$

Domination rank [87], domination count [43], and domination strength [88] are usually used to assign rank values. A variety of density estimation methods have been proposed. The widely used methods include the niching and fitness sharing strategy [43], crowding distance [27], K-nearest-neighbor method [89], fast sorting [90], and gridding and ϵ-domination method [91–95].

In recent years, a variety of methods [77,83,96–98] and the extension of Pareto domination to fuzzy domination [99,100] have been proposed to improve the algorithmic performance. However, the basic idea is still the same as presented here.

It should be noted that there are many redundant comparisons between individuals in the rank assignment procedure if the definition of Pareto domination is to be followed. Thus, some new data structures have been proposed to improve the sorting performance [101,102].

2.2.2. Complete orders over populations

Recall that, in an MOEA, populations are actually updated from one generation to another. Selection mechanisms based on performance indicators define a complete order over populations. Let $I(P)$ be a quality indicator which assigns a real value to a non-dominated population P. A full order, $\prec_p$ is defined as follows:

$$P \prec_p Q \quad \text{iff } I(P) < I(Q),$$

where a smaller value of indicator $I(P)$ is preferred.

The idea of using performance to guide the selection was first proposed by Fleischer in [103]. Huband et al. [104] proposed the first MOEA with a hypervolume guided selection procedure. Indicator-based selection has since then been widely applied in MOEAs [105,106]. Zitzler and Künzli generalized the idea and proposed an indicator-based MOEA [57,58]. These methods are called *indicator-based MOEAs*, and they are discussed in Section 2.1.3. A major disadvantage with this kind of selection is that it might be time-consuming. More work is needed to improve the efficiency.

2.3. Reproduction

Conventional reproduction operators designed for scalar objective EAs could be directly used in MOEAs. However, the optima structures of scalar optimization and multiobjective optimization are quite different, i.e., an isolated point or several points with the same objective value in scalar optimization and a solution set in multiobjective optimization. The operators designed for scalar optimization might not be suitable for multiobjective optimization. For example, in [107], it is observed that some widely used reproduction operators did not work well for rotated problems. We argue that this difference should be emphasized in evolutionary multiobjective optimization. The characteristics and/or problem-specific knowledge should be considered in designing reproduction operators for multiobjective optimization [108–111].

Recent advances in reproduction are summarized as follows.

2.3.1. DE-based approaches

The *differential evolution* (DE) algorithm [112,113], which was introduced by Storn and Price in 1995, uses weighted difference between solutions to perturb the population and to create candidate solutions. The new trial solutions are partly from the candidate solutions and partly from the old population. The DE algorithm was originally designed for scalar objective optimization. However, it has since attracted much attention in multiobjective optimization because of its simplicity to implement and efficiency for solving problems.

A Pareto-frontier differential evolution (PDE) algorithm was proposed in [114]. The major modifications are (1) the step length parameter F is randomly sampled from a Gaussian distribution $N(0, 1)$, and (2) the parents are from the non-dominated set. To find a uniformly distributed, near-complete, and near-optimal PF, a multiobjective DE based on Pareto-adaptive ϵ-dominance and orthogonal design was proposed in [115]. In this approach, the DE/rand/1/bin strategy is used to produce new trial solutions. It also adopts some previous strategies, such as (1) population initialization based on orthogonal design, (2) archive updating by the Pareto-adaptive ϵ-dominance and saving extreme solutions, and (3) alternatively selecting parents by a random scheme and an elitist selection scheme, to improve its performance. A multiobjective DE algorithm with diversity enhancement strategies was proposed in [116].

The DE algorithm has also been extended to tackle discrete or mixed continuous and discrete MOPs. A multiobjective DE algorithm was proposed in [117] for mining numeric association rules. A solution contains both integer and real values. To solve this problem, a chromosome is treated as a real vector and a rounding operator is applied to repair a real component to an integer component. In [68], a memetic algorithm based on DE was proposed to deal with MNFSSPs. It is a discrete problem, but the chromosome is a real vector which enables DE to work on it. A largest-order-value rule based on random key representation is used to convert a real vector to a job permutation. A problem-dependent local search is applied to a job permutation to improve its quality.

Since the DE algorithm has two control parameters which are not easy to set properly, self-adaption has also attracted much attention recently. In [98], the two control parameters are randomly picked up from predefined ranges. In this approach, a crowding entropy-based diversity measure is applied to maintain an elitist archive.

2.3.2. Immune-based approaches

Due to the clonal selection and affinity maturation by hypermutation, the immune system is able to adapt B-cells to new types of antigens. By simulating this phenomenon, artificial immune systems were proposed to deal with optimization problems [118]. Recently, immune systems have been extended from scalar objective optimization to multiobjective optimization. In multiobjective immune systems, clonal selections based on Pareto dominance are usually used to select promising solutions while crossover and mutation operators are widely used to generate new trial solutions.

Most of the multiobjective immune systems focus on static problems. In [119], two mutation operators are used to mutate antibodies with different qualities. An archive is used to store elitist solutions to approximate the PF. In [120], a hybrid multiobjective algorithm based on an immune system and bacterial optimization was proposed to deal with bi-objective no-wait flow-shop scheduling problems. In this approach, a linear combination method is applied to generate antibodies which are improved by using bacterial optimization operations. In [96], a non-dominated neighbor immune algorithm was proposed for multiobjective optimization. The selection strategy emphasizes more on less crowded solutions. In [69], a hybrid immune multiobjective optimization algorithm based on a clonal selection principle was proposed. In this approach, Gaussian and polynomial mutations are adaptively applied to mutate the new trial solutions after crossover. The selection procedure proposed in [96] is used to update the population directly. In [121], a multiobjective immune system based on a multiple-affinity model was proposed.

Some immune algorithms have been applied to dynamic and uncertain optimization problems. In [122,123], a multiobjective immune system was proposed to deal with dynamic multiobjective problems with constraints. In [124], a multiobjective immune system was presented to find Pareto optimal robust solutions for bi-objective scheduling problems.

2.3.3. PSO-based approaches

Particle swarm optimization (PSO) is a population-based stochastic optimization technique developed by Eberhart and Kennedy in 1995 [125,126], inspired by the social behavior of bird flocking or fish schooling. Moore and Chapman extended this idea to multiobjective optimization in 1999 [127]. Since PSO cannot be directly applied to multiobjective optimization, there are two issues to be considered when extending PSO to multiobjective optimization. The first one is how to select the global and local best particles (leaders) to guide the search of a particle. The second is how to maintain good points found so far. For the latter, a secondary population is usually used to maintain the non-dominated solutions. Here, we focus on the first issue.

In [128], the particles are clustered into groups, the global best particle of a particle is from its group and a weighted-sum of the objectives is used to maintain its local best particle. In [129], a tournament niche method is introduced to select the global best particle, and the local best particle is updated by the Pareto dominance. In [130], the global best particle is selected from the non-dominated solutions with a roulette wheel selection in which the density values are used as fitness. The self-adaptive control parameter is also considered. In [131], a preference order, a generalization of Pareto dominance, is introduced to rank all the particles and thus to identify the global best particle.

Three EA–PSO hybrid algorithms were proposed in [62]. The fitness assignment strategy is based on that of strength Pareto evolutionary algorithm 2 (SPEA2) [89]. The global best particle is selected from the external archive by a tournament selection, and the neighborhood best particle is selected as the one with lowest strength Pareto fitness.

In [132], a multiobjective PSO was designed to tackle multiobjective mixed-model assembly line sequencing problems. To this end, a coding strategy and a local search are introduced. The global best particle is the non-dominated solution in the archive with the highest crowding distance in the archive.

A multiple swam algorithm was proposed in [70]. Several components, such as cell-based rank density estimation, population growing and declining strategies, and adaptive local search, are designed to improve the algorithmic performance. A leader selection was proposed to assign a leader for each group.

In [94], an archive is applied to maintain the non-dominated solutions found so far, and a mutation operator is used to keep the population diversity. To choose a global best particle, the non-dominated ones in sparse areas are emphasized.

In [133], a fuzzy clustering-based PSO was proposed to tackle electrical power dispatch problems. A fuzzy clustering technique is applied to maintain the external archive. A self-adaptive mutation operator is also used to generate new trial solutions. A niching mechanism is designed to find the global best particle for each particle and thus to emphasize less explored areas. Finally, a fuzzy decision rule is used to assist decision making.

In [134], a multiobjective comprehensive learning particle swarm optimizer (MOCLPSO) was presented. MOCLPSO uses a learning strategy whereby all other particles' historical best information is used to update a particle's velocity. This strategy enables the diversity of the swarm to be preserved to discourage premature convergence.

In [135], a two-local-best (lbest)-based multiobjective PSO (2LB-MOPSO) technique was proposed. Different from canonical multiobjective PSO, 2LB-MOPSO uses two local bests instead of one personal best and one global best to lead each particle. The two local bests are selected to be close to each other in order to enhance the local search ability of the algorithm. Compared to the canonical multiobjective PSO, 2LB-MOPSO shows great advantages in convergence speed and fine-searching ability.

In [136], PSO is used in the MOEA/D framework. Each particle is responsible for solving one subproblem.

More works on multiobjective PSO are presented in [16].

2.3.4. Probabilistic model-based approaches

Probabilistic model-based EA is a new computing paradigm in evolutionary computation. The main feature of these algorithms is that they do not use traditional crossover or mutation operators to generate new solutions. Instead, they explicitly extract global statistical information from their previous search and build a probability distribution model of promising solutions. Based on the extracted information, new solutions are sampled from the model thus built. Compared to traditional EA methods, they emphasize the population distribution information rather than the individual location information. The key issues in these methods include model selection before executing the algorithm and model building and sampling in the running process. The following methods share the above basic ideas and they differ from each other on origins.

Ant colony optimization (ACO) [137] was introduced by Dorigo in 1992. ACO takes inspiration from the behavior of real ant colonies and is used to solve optimization problems. Ants deposit pheromone on the ground in order to mark some favorable paths followed by other members of the colony with higher probability. ACO exploits a similar mechanism by constructing a probability matrix, named the pheromone model, to denote the probability to choose an edge in a graph and thus sampling new solutions. The structure of ACO probability model makes it a natural choice for discrete optimization. In the case of multiobjective optimization, ACO has been applied to traveling salesman problems [138,139], vehicle routing problems [140], flow-shop scheduling problems [141], portfolio selection [142,143] and others.

The *cross entropy* (CE) method [144] was proposed by Rubinstein in 1997. CE originated from the field of rare-event simulation involving the estimation of parameters for a number of probability distributions associated with some rare events. CE methods iteratively generate sample points from the probability model and update the model parameters on the basis of the data. Currently, however, there are not many reports on applying CE for multiobjective optimization. In [145], a CE-based approach was proposed for MOPs. In the approach, a population is partitioned into several clusters, and a CE method with a Gaussian model is utilized in each cluster.

The *quantum-inspired genetic algorithm* (QGA) [146,147] was first proposed by Han and Kim in 2000. The QGA simulates the quantum mechanism and uses a Q-bit vector to represent a solution. The Q-bit vector actually denotes probability distributions of all Q-bits to be 0 or 1. A quantum gate is used to generate new individuals. In [148], a multiobjective QGA was proposed to deal with hardware–software co-synthesis problems in embedded systems. Another version of the multiobjective QGA was proposed to deal with flow-shop scheduling problems [63].

The *estimation of distribution algorithm* (EDA) [149] was first introduced by Mühlenbein and Paaß in 1996. Most EDAs aim to discover the variable linkage information from the population to benefit offspring generation. To this end, different models with univariate, bivariate, and/or multivariate variable linkages have been widely studied [150]. Depending on the models used, EDAs are suitable for both combinatorial and continuous optimization. In the case of continuous multiobjective optimization, Okabe et al. [151] proposed a Voronoi model-based method. Bosman and Thierens [152] proposed an EDA method based on a mixture univariate Gaussian model. Dong and Yao [153] proposed a multivariate Gaussian model-based method. Igel et al. [106] extended the *covariance matrix adaptation evolution strategy* (CMA-ES) for dealing with MOPs. In the case of combinatorial multiobjective optimization, Laumanns and Očenášek [154] proposed a Bayesian network-based method for knapsack problems. Pelikan et al. [155] designed a method with hierarchical Bayesian networks to study building boxes for binary coding problems.

The PF and PS of a continuous MOP are piecewise continuous $(m-1)$-dimensional manifolds under mild conditions [2]. Based on this regularity property, Zhang et al. [110] proposed a *regularity model-based multiobjective estimation of distribution algorithm* (RM-MEDA) for continuous MOPs with variable linkages. In some cases, a good approximation to both the PF and the PS is required by a DM. To this end, the RM-MEDA has been extended in [111] to tackle a class of MOPs in which the dimensionalities of the PF and the PS manifolds are different. RM-MEDAs have been applied to static MOPs [110,111], dynamic [156] MOPs, MOPs with local PFs [157], MOPs with high search dimensions [158]. Recently, the RM-MEDA has been improved by combing it with some other techniques [159]. A basic idea behind RM-MEDAs is to use statistical and machine-learning techniques to guide the search of EDAs. Dimension-reduction techniques are thus used in RM-MEDAs. Some other ways to use this regularity property are referred to in [160,161]. The research work on RM-MEDAs is among very few efforts to design MOEAs based on mathematical programming theory.

2.3.5. SA-based approaches

Simulated annealing (SA) is a single-point-based global optimization technique which is inspired by annealing in metallurgy [162]. Due to its simplicity, SA has been incorporated into multiobjective frameworks for dealing with MOPs.

Like some other MOEAs, multiobjective SAs also need to maintain an archive to store current non-dominated solutions and to use reproduction operators to generate new solutions. The main difference between multiobjective SAs and other MOEAs is on how to update a solution when the offspring individual is dominated by the parent. The SA updating rule is usually used in such case.

In [163], the SA updating rule is used to choose the next individual when an offspring individual is dominated by the parent. A similar method was proposed in [164], in which a domination-based energy function is used to calculate the probability to accept a dominated new trial solution. In [165], the domination relationship between an offspring point and its parent as well as archive points is systematically studied. A multiobjective SA with a single point was introduced in [166]. In this approach, each objective is assigned a different cooling schedule, taking into account the prioritization of that objective. The probability to accept a new solution which is worse than the parent is controlled by using SA rules.

2.3.6. Other approaches

In addition to the above-mentioned methods, there are also many other heuristics which are originally designed for scalar objective optimization. By incorporating with the Pareto domination and/or population (archive) updating strategies, these heuristics could also be extended to tackle MOPs. These meta-heuristics include *tabu search* [71,167], *scatter search* [168], and the GRASP approach [169].

2.4. Other issues

2.4.1. Theoretical studies of MOEAs

Theoretical analysis of algorithms is important both for explaining the algorithmic performance and for guiding the algorithmic design. Some recent analyses of MOEAs are as follows. In [170], a rigorous running time analysis of an algorithm on pseudo-problems was presented. In [171,172], the convergence of DE-based MOEAs are discussed. In [173], online and offline measurements were introduced to detect convergence. The measurements are based on performance indicators and a convergence threshold which is predefined offline or detected from the data online. In [174], the convergence property of a PSO-based MOEA was studied. Two fitness inheritance methods were proposed in [175], based on a statistical evaluation of the performance of an NSGA-II-like algorithm. The experimental results shown that the performance of the approaches is quite similar to that of those general MOEAs.

2.4.2. Adaptation

MOEAs usually have several control parameters, and their performance is highly related to the parameters. Usually, the parameters are predefined based on algorithmic knowledge or an empirical study. In real-world applications, this strategy might not be applicable. Thus, adaptively tuning control parameters has attracted much attention recently.

In [176], the crossover probability and mutation probability are varying with both population diversity and progress of the search. A similar idea was used in [177], where the mutation probability is adapted along the evolution. In some cases, the upper and lower boundaries of the parameters are easy to estimate. The approach in [98] is based on this strategy, and the parameters are randomly selected from given boundaries. Huang et al. [178] employed an adaptive DE algorithm capable of learning the crossover rate CR and suitable mutation strategies in their algorithm.

3. MOEAs for complicated problems

3.1. Constraint handling in MOEAs

Although MOEAs have been more extensively investigated within the context of unconstrained and bound constrained MOPs, various general constraints are involved when solving real-world problems. Typically, the search space Ω of a constrained MOP can be formulated as follows:

$$\Omega = \begin{cases} g_j(x) = g_j(x_1, x_2, \ldots, x_n) \le 0 & j = 1, 2, \ldots, J \\ h_k(x) = h_k(x_1, x_2, \ldots, x_n) = 0 & k = 1, 2, \ldots, K \\ x_i^L \le x_i \le x_i^U & i = 1, 2, \ldots, n, \end{cases} \quad (2)$$

where $g_j(x)$ and $h_k(x)$ are inequality and equality constraint functions, respectively. Generally, equality constraints are transformed into inequality forms, and then combined with inequality constraints using

$$G_j(x) = \begin{cases} \max\{g_j(x), 0\} & j = 1, 2, \ldots, J \\ \max\{|h_{j-J}(x)| - \delta, 0\} & j = J+1, J+2, \ldots, J+K, \end{cases} \quad (3)$$

where δ is a tolerance parameter for the equality constraints.

Due to the presence of the constraints, the search space is partitioned into feasible and infeasible regions. Many constraint-handling methods have been proposed to solve constrained SOPs [179]. According to the characteristics of different constraint-handling methods, Coello Coello [180] grouped them into five categories: (1) penalty functions; (2) special representations and operators; (3) repair algorithms; (4) separate objective and constraints; and (5) hybrid methods. Although not all constraint-handling methods developed for scalar objective optimization are suitable for *constrained multiobjective problems* (CMOPs), some of them have been successfully extended to constrained multiobjective areas [75,108,122,181–183]. This section introduces some of the constraint-handling methods used for constrained multiobjective optimization in recent years.

The penalty function is known as the most popular constraint-handling method. It was first introduced by Courant [184] and extended to solve CMOPs by many researchers [181,183]. The basic idea is to transform a constrained optimization problem into an unconstrained one by introducing a penalty term into the original fitness function to penalize constraint violations [182]. There are several schemes [180] to impose suitable penalties when solving CMOPs, including the death penalty, static penalty, dynamic penalty, and adaptive penalty. Woldesenbet [183] introduced a very promising self-adaptive penalty function recently for constrained multiobjective evolutionary optimization. The method tracks the percentage of the feasible solutions in the current population to determine the amount of penalty to be added. A small percentage of feasible individuals results in a larger penalty while a larger percentage generates a small penalty factor. This technique is able to balance information extraction from feasible and infeasible solutions.

In [3], a constrained dominance concept was introduced by Deb et al. to handle CMOPs. A solution x is said to constrain dominate a solution y if (1) x is feasible, while y is infeasible; (2) both are infeasible and x has less constraint violation than y; or (3) both are feasible and x dominates y. The solutions are ranked using the non-constrain-dominated method while the superior ones are selected to evolve. This method is also known as superiority of the feasible.

An immune algorithm is also commonly used to handle constraints. In [122], Zhang proposed a *constrained nonlinear multi-objective optimization immune algorithm* (CNMOIA) based on the humoral immune response principle and ideas of T-cell regulation. The algorithm adopted and modified a uniform design scheme [185] to provide an alternative feasible solution set for dealing with constraints and infeasible solutions created during the evolutionary process.

In recent years, emphasis has been increasingly placed on hybrid methods. In [186], an ensemble of constraint-handling methods was introduced to tackle the difficulty of selecting different constraint-handling methods. Liu [187] integrated fuzzy membership functions into the selection-based constraint-handling strategy and constructed a new constraint-handling method. In [188], a constraint-handling strategy based on infeasible individual stochastic binary modification was proposed. The method randomly modifies infeasible individuals into feasible solutions according to a predefined modification rate (R_m) during evolutionary optimization.

3.2. MOEAs as constraint-handling methods

Constraint-handling methods based on multiobjective concepts have been increasingly investigated in the past few years [182,189,190]. This constraint-handling technique treats the constraints as additional objectives to be optimized. In [182], Cai and Wang classified the methods based on multiobjective concepts into two categories:

1. Methods based on biasing feasible over infeasible solutions;
2. Methods based on multiobjective optimization techniques.

The first category generally treats the constrained SOP as a bi-objective optimization problem. The original objective and the constraint violation are considered as two separated objectives to be optimized synchronously. On the other hand, the second category modifies the constrained SOP into an MOP with $m + 1$ objectives, where m is the number of constraints.

Simplicity is the major merit of an MOEA as a constraint-handling method, as no fitness modification is required. However, treating constraint violation as an extra objective increases the computational complexity of the algorithm, and thereby may slow down the algorithm.

3.3. MOEAs and multimodal problems

MOEAs can also be used to solve multimodal problems. Multimodal optimization algorithms aim to find numerous global/local optima in one single run.

In [191], Yao et al. described a *bi-objective multipopulation genetic algorithm* (BMPGA) aiming to solve multimodal optimization problems on a real-valued differentiable landscape. The algorithm uses two objectives to enhance the diversity of the population. The first objective is known as the original fitness function, while the second objective is the gradient of the function. By using the two objective functions, Yao et al. showed that the BMPGA is able to generate stable niching behavior over some benchmark functions.

In [192,193], Deb and Saha converted a scalar objective multimodal optimization problem into a bi-objective optimization problem so that all global/local optima became members of the resulting weak Pareto-optimal set. This approach also treats the gradient of the function as the second objective. The experimental results showed the superior performance of the modified NSGA-II on the tested constrained and unconstrained multimodal problems.

Compared with other scalar objective multimodal optimization methods, MOEAs are able to maintain a population with higher diversity. However, the extra objective and non-domination sorting procedure are computationally costly.

There are also multimodal MOPs. In [157], Zhou et al. suggested using a two-phase search strategy to dealing with MOPs with many local PFs. Some good points near the global PF are found in the first phase; and the whole PF is then approximated in the second phase.

3.4. MOEAs for many-objective problems

In many real-world applications, there usually exist problems with more than three objectives. These many-objective optimization problems are challenges to MOEAs, and the most widely used selection operators do not work well in these cases. To deal with many-objective problems, new ideas and techniques are required to improve the performances of current MOEAs.

Much work has been done to address the importance of solving many-objective optimization problems [94,194,195]. A large number of objectives introduce extra difficulties with respect to computation, visualization, and decision making for the conventional MOEAs. The state-of-the-art MOEAs such as NSGA-II are not effective in solving optimization problems with more than three objectives. In [196,197], the Pareto dominance relation and rank definition are modified to increase the selection pressure toward the PF and to solve many objective optimization problems. In [194], a dynamical MOEA is proposed based on the principle of thermodynamics. The algorithm defines a new concept known as *L*-optimality, which not only takes into account the number of improved objective values but also considers the values of improved objective functions if all objectives have the same importance. Based on the new definition, the MOEA uses a new selection scheme to deal with many-objective optimization problems. The main advantage is that this method can generate several different test points simultaneously at each iteration.

3.5. Computationally expensive multiobjective optimization

In some multiobjective engineering optimization problems [198,199], the process of locating the PF could be extremely computationally or financially expensive. These problems generally demand huge numbers of physical experiments or time-consuming simulations. In order to solve these problems, a method that can produce reasonably good solutions within a given (limited) computational cost is desired [200]. In [201], Knowles classified the computationally expensive MOPs into the following classes.

1. The problem has multiple, possibly incommensurable, objectives.
2. The time taken to perform one evaluation is of the order of minutes or hours.
3. Only one evaluation can be performed at one time (no parallelism is possible).
4. The total number of evaluations to be performed is limited by financial, time, or resource constraints.
5. No realistic simulator or other method of approximating the full evaluation is readily available.
6. Noise is low (repeated evaluations yield very similar results).
7. The overall gains in quality (or reductions in cost) that can be achieved are high.
8. The search landscape is locally smooth but multimodal.
9. The dimensionality of the search space is low to medium.

These computationally expensive MOPs have attracted the attention of many researchers in recent decades [200–207]. The Gaussian stochastic process model is known to be one of the most popular and efficient methods for dealing with expensive SOPs [200]. It also has been extended to the multiobjective case. In [207], Emmerich et al. generalized the probability of improvement as well as the expected improvement to multiobjective optimization. The method selects multiple test points with high metric values during the search process. This idea was extended in [194], where the algorithm optimizes a hypervolume-based metric using CMA-ES. The advantage of this method is the use of Gaussian random-field metamodels to predict the objective function values for new candidate solutions, which is able to speed up the convergence.

Knowles [201] proposed ParEGO, which applies the EGO algorithm to a randomly selected aggregation function to find which point to evaluate in the next step. Although the results are very encouraging, the method considers only one aggregation function at each iteration, which can be a major drawback for MOPs. Note that ParEGO is able to offer a more effective search on problems like the instrument setup optimization problem, where only one function evaluation can be performed at a time.

More recently, Zhang et al. [200] proposed a *multiobjective evolutionary algorithm based on decomposition with the Gaussian stochastic process model* (MOEA/D-EGO) for expensive multiobjective optimization. In this algorithm, a Gaussian stochastic process model for each subproblem is built, and the expected improvements for the subproblems are optimized at the same time. It uses parallel computing as a basic tool for solving an expensive MOP, which makes the algorithm very efficient and effective.

3.6. Dynamic multiobjective optimization

In many real-world applications such as investment optimization, robot navigation, and control system design, the fitness function, parameter space, and constraints as well as the location of the optimal front may change dynamically over time [123,208]. These kinds of problems are known as the dynamic MOPs. The dynamic environment generates great challenges for classical MOEAs.

Although dynamic scalar objective optimization has been studied for decades, the *dynamic multiobjective optimization problem* (DMOP) has just started to attract attention among many researchers [123,208–212]. The main challenge in DMOPs is that the PFs may change over time. To solve a DMOP, diversity and adaptive exploring ability are the key issues, as the changed PF has to be rapidly discovered.

To overcome the difficulties caused by the dynamic environment, some researchers [208,213–215] transform a DMOP into a dynamic or fuzzy dynamic SOP. In [212], a direction-based method is adopted to handle DMOPs. This method is a neighborhood search algorithm. It constructs an environmental recognition rule using the difference of environmental evaluations. This method gives promising performance for problems that have slowly changing environments and dimensions of the parameter space. In [156], Zhou et al. proposed to predict the PS when the environment changes. In [216], Hatzakis et al. used a queuing multiobjective optimizer and multivariate autoregressive forecasting model to solve DMOPs. This method uses time series analysis to predict the location of the new PF. More recently, Goh and Tan [217] proposed a new coevolutionary paradigm which incorporates the competitive and cooperative mechanisms observed in nature to facilitate adaptive problem decomposition in coevolution. During the process of competition and cooperation between different subpopulations, the whole population is optimized according to the requirement of different time instants to handle both static and dynamic MOPs. The advantage of this algorithm is that it is able to adapt quickly to a changing environment.

3.7. Noisy multiobjective optimization

In real-world multiobjective optimization, noise is one of the main issues faced by researchers. When noise is present, the optimization process becomes relatively unstable, as the algorithms have to cope not only with multiple objectives, but also with the stochastic noise. Various approaches have been proposed to deal with noise in multiobjective optimization [218–223].

There are a few methods to deal with noise in MOEAs. The most commonly used methods are known as resampling and the probabilistic ranking process. With the resampling method, the noise is reduced by a factor. Although resampling is effective in solving noisy problem, it is computationally costly. In [224], Hughes introduced the probabilistic ranking process, which takes into account the standard deviation in the evaluation of each solution to deal with the effect of noise.

In [220], Deb and Gupta proposed a technique for finding robust solutions for MOPs in engineering designs. This method can be considered as a resampling method in the design space. The objective values of each solution are the averaged values of a number of solutions within its neighborhood. This algorithm proposes a user-friendly procedure which allows a user to find robust solutions with a user-defined limit to the extent of change in objective values with respect to local perturbations.

In [222], Goh and Tan proposed an experiential learning directed perturbation operator that adapts the magnitude and direction of variation using the past experiences for faster convergence. The proposal also has a gene adaptation selection method to help the search process in escaping from local optima or premature convergence. A possibility archiving model based on the concept of possibility and necessity measures is used to handle problems with uncertainties or noise. The experimental results showed that with the proposed method the algorithm performs well in terms of proximity, diversity, and distribution for both noiseless and noisy problems.

Bui et al. [221] used a framework of local models to deal with noise in multiobjective optimization. The method divides the search space into a number of non-overlapping hyperspheres. The average performance of the sphere is used to move solutions within each sphere and to filter the noise. The algorithm is able to generate a good performance in terms of both convergence and diversity. The drawback of this method is the extra complexity that is introduced by dividing the decision space, which may slow down the speed of the algorithm.

In [223], Syberfeldt et al. proposed a noise-handling method by using an iterative resampling procedure that reduces the noise until the likelihood of selecting the correct solution reaches a given confidence level. This technique is able to prevent the propagation of inferior solutions in the selection process due to noisy objective values. Different from other methods, the proposed technique varies the number of samples used per solution based on the amount of noise in the local area of the search space. In this way, the algorithm avoids wasteful samplings when the benefit of additional samplings is insignificant.

3.8. MOEAs for combinatorial and discrete problems

Combinatorial and discrete optimization problems such as routing, task allocation, and scheduling are important optimization applications in the real world. For conventional methods, the time required to solve a combinatorial problem may increase exponentially in the worst case, thereby making them computationally too costly. Moreover, if the optimization involves multiple objectives, the process becomes more complex and difficult to solve.

Recently, researchers have shifted their focus from conventional methods to more efficient MOEAs [225–227]. Various MOEAs have also been proposed to solve multiobjective combinatorial problems [228–230]. In [228], Ishibuchi et al. carried out an empirical study on similarity-based mating strategies for evolutionary multiobjective combinatorial optimization. In this work, the performance of recombining similar or dissimilar parents is examined. The effect of biasing selection probabilities toward extreme solutions that are dissimilar from other solutions in each population is also studied. It concludes that the performance of MOEAs for combinatorial can be improved by the similarity-based mating scheme.

In [229], Xing et al. presented a simulation model to solve a multiobjective flexible job-shop scheduling problem (FJSSP). The FJSSP is very important in the fields of combinatorial optimization and production management. Throughout the experiments, Xing et al. showed that MOEAs are very effective for solving the FJSSP.

In [230], Chang and Chen proposed a new algorithm, called the subpopulation genetic algorithm II, to solve multiobjective combinatorial problems. The algorithm develops a mechanism to exchange information among subpopulations. Once a subpopulation reaches a better non-dominated solution, other subpopulations will apply them directly in their search space. In this way, all individuals in the same population will be guided to search toward the true PF.

4. Benchmark problems and performance measures

4.1. Benchmark problems

Benchmark problems are important for both assessing the qualities of MOEAs and designing MOEAs. Quite a few test problems have been designed in the early stages of MOEA research. Since they are relatively simple, they are not widely used nowadays. In recent years, several test suites have been designed, and some widely used ones are as follows.

In the report of the CEC 2007 Special Session and Competition [231], 19 multiobjective minimization problems are described, including bi-objective, tri-objective, and five-objective problems. The classical benchmark functions may have the same parameter values for different variables/dimensions at the global optimum that may be located at center or bounds of the search range. To overcome these shortcomings, shifting and rotation have been applied in the parameter space of the CEC 2007 test problems.

Apart from demonstrating adequately the usefulness of MOEAs in finding multiple Pareto solutions for static MOPs, there is a growing need for solving DMOPs in a similar manner. The paper [212] addresses this issue by developing a number of test problems and by suggesting a baseline algorithm. Since, in a DMOP, the PS may change with time, a suite of five test problems offering different patterns of such changes and different difficulties in tracking the dynamic PF are presented.

Many of the multiobjective test problems employed in the EA literature have not been rigorously analyzed. Thus it is difficult to draw accurate conclusions about the strengths and weaknesses of the algorithms tested on them. In [232], some problems from the EA literature are systematically reviewed and analyzed, each belonging to the important class of real-valued, unconstrained, multiobjective test problems. To support this, a set of test problem criteria are introduced. The analysis of test problems highlights a number of areas requiring attention. Not only are many test problems poorly constructed, but the important class of non-separable problems, particularly non-separable multimodal problems, are also poorly represented. Motivated by these findings, a flexible toolkit is proposed for constructing well-designed test problems.

Li and Zhang [29] introduced a general class of continuous multiobjective optimization test instances with arbitrarily prescribed PS shapes. There problems with complicated PS shapes have taken challenges to many MOEAs. The decomposition-based MOEAs are suggested for dealing with complicated PS shapes. Moreover, in the CEC 2009 Special Session and Competition, unconstrained and constrained test functions are designed with complicated PS shapes [31].

4.2. Performance measures

As the outcome MOEAs is usually an approximation of the PS, the quantitative comparison of the performance of different algorithms becomes an important issue. There are two goals in measuring a multiobjective algorithm: (1) convergence to the true PF, and (2) distribution of approximated solutions. Generally, methods that assign each approximation set a vector of real numbers that reflect different aspects of quality are well accepted among researchers. The elements of the vector to represent the performance of MOEAs are called the unary quality indicators. Over the past few decades, many unary indicators have been introduced (Table 1).

Table 1
Unary quality indicators.

Indicator	Description	Reference
I_{R2}	R indicator	[233]
$I_{\overline{H}}$	Hypervolume indicator	[233]
I_{HC}	Enclosing hypercube indicator	[234]
I_O	Objective vector indicator	[234]
I_I	Unary ε-indicator	[234]
I_P	Number of Pareto points contained	[234]
I_W	Average best weight combination	[235]
I_D	Distance from reference set	[236]
I_{PF}	Fraction of PF covered	[237]
I_{ER}	Error ratio	[238]
I_{ONVG}	Overall non-dominated vector generated	[238]
I_{GD}	Generational distance	[238]
I_{ME}	Maximum PF error	[238]
I_{CD}	Chi-square-like deviation indicator	[87]
I_S	Spacing	[239]
I_{MS}	Maximum spread	[240]
I_{MD}	Minimum distance between two solutions	[241]
I_{CE}	Coverage error	[241]
I_{DU}	Deviation from uniform distribution	[3]
I_{OS}	Pareto spread	[242]
I_A	Accuracy	[242]
I_{NDC}	Number of distinct choices	[242]
I_{CL}	Cluster	[242]

5. Applications

Due to the rapidly growing popularity of MOEAs as effective and robust multiobjective optimizers, researchers from several domains of science and engineering have been applying MOEAs to solve optimization problems arising in their own fields. The literature on MOEA applications is huge and multifaceted. Therefore, we summarize only the major applications of MOEAs in Table 2.

6. Conclusions and future directions

In this paper, research work on MOEAs has been surveyed. The advances in MOEA designs, MOEAs for complicated MOPs, benchmark problems, performance measures, and some applications, have been covered. Evolutionary multiobjective optimization is still in its early stage, although there have been a huge number of publications. The following issues, along with others, should define the future research trends of MOEAs.

- New algorithmic frameworks: The current popular frameworks are Pareto-dominance based, decomposition based and indicator based. The strengths and weaknesses of these frameworks should be investigated thoroughly. Some new frameworks or combinations of these frameworks may emerge in the near future. These frameworks will also raise other new research issues and opportunities.
- Offspring generators and description of approximation: The purpose of MOEAs is to approximate a set of Pareto optimal solutions instead of a single one. The distribution of the Pareto optimal solutions can exhibit some regularities. How to make use of these properties and to design efficient offspring generators would be worth investigating. Although most current MOEAs use a finite number of solutions to approximate the PF, it would be interesting to investigate how to use other approaches to describe an approximate PF. In the case of continuous problems, one may consider the first-order or higher-order approximations. However, in the case of combinatorial MOPs, it could be a very challenging issue to use the first-order or higher-order approximations.
- Interactive MOEAs: Interaction with the DMs has been identified as a very important research avenue in MOEAs. It should call for joint research efforts from evolutionary computation, decision science, software engineering, and psychology. Such work will be needed for widening MOEA applications.
- Dynamic and noisy multiobjective optimization: Although some work has been done on this, some fundamental issues have not been studied well yet. To develop an efficient algorithm, one may need to study first how to model or classify noises and dynamic environments. Future works on dynamic and noisy scalar objective optimization will be definitely useful for this research agenda.

Table 2
Summary of applications of MOEA to real world problems.

Areas and details	Types of MOEA applied and references
Scheduling heuristics	
Planning	An ε-constrained method for constraint handling, an NSGA-II, and a multiobjective constrained algorithm (MCA) [243]. An NSGA-II with the geographical information system (GIS) [244]. A multiobjective particle swarm (MOPS) [132]
Scheduling	Two derivatives of bi-objective genetic algorithms (GAs) with adjustable crossover and mutation rates [245]. An ACO with local search [229]. Planning inspections and other operations within a software development (SD) project with an MOEA [246]. A multiobjective genetic algorithm (MOGA) [247]. A form of utility theory with specific subsets of the PF by merely ranking a small set of initial solutions [248]. A hybrid multiobjective algorithm based on the features of a biological immune system (IS) and bacterial optimization (BO) (HMOIA) [120]. A multiobjective immune algorithm [124]. A Pareto-based multiobjective DE [68]. A hybrid quantum-inspired genetic algorithm (HQGA) with two trimming techniques for population diversity maintenance [63]. An MOEA/D [35]
Data mining and rule extraction	
Data mining	A Pareto-based multiobjective DE algorithm [117]. A Pareto-based multiobjective GA [249]. A multiobjective GA [250]
Rule extraction	An MOGA approach for fuzzy association rule mining in terms of three important criteria: strangeness, interestingness, and comprehensibility [250]. Six different MOEAs [251]. A multiobjective genetic cooperative competitive learning (GCCL) [252]. A multiobjective genetic programming [253]. An MOEA/D [41]
Assignment and management	
Placement	A multiobjective variable-length GA based on NSGA-II [254]
Management	An MOEA with multiple neighboring regions [255]
Resource allocation	Tabu search and multiobjective concepts with combination of a dominance rule and a multicriteria filtering method [167]
Assignment	A VEGA and a SPEA [256]. A multiobjective staff-to-job assignment model (MUST) incorporated with PSO [257]
Routing	Multiobjective combinatorial optimization [71]. An MOEA with two VRPSD-specific heuristics for local exploitation and a route simulation method [258]
Packing	A multiobjective evolutionary particle swarm optimization algorithm (MOEPSO) [129]. The University of Sheffield's Genetic Algorithm Toolbox for Matlab [259]
Circuits and communications	
Antenna array design	A real coded NSGA-II [260]. A decomposition-based multiobjective evolutionary algorithm with differential evolution (MOEA/D-DE) [261,262]. A multiobjective DE [36]
Wireless sensor network	The normal boundary intersection (NBI) method and an NSGA-II [263]. An MOEA/D [40]
Circuit design	A multiobjective evaluation mechanism of fitness with weight-vector adaptation and circuit simulation [176]. An MOEA/D [33]
DS-CDMA design	Two multiobjective artificial immune systems based on the clonal selection principle [181]
Bioinformatics	
Molecular docking	A multiobjective particle swarm optimization (ClustMPSO) [128]
DNA sequence design	A constrained controlled elitist NSGA-II [264]
Oligonucleotide probe design	An ε-MOEA [265]. Cluster-oriented genetic algorithms (COGAs), a DE, and an NSGA-II [266]. An NSGA-II and a SPEA2 [267]
Gene network	Genetic and hybrid approaches for multiobjective optimization with fuzzy dominance [77]
Control systems and robotics	
Greenhouse control	A novel multiobjective optimization immune algorithm in dynamic environments [123]
Robot motion planning	An MOGA, an NSGA-II, and an MODE with normalized weighting objective functions method [268]. An MOGA [269]. An MOEA/D [39]
Control scheme design	A prioritized multiobjective stochastic algorithm based on SA (PMOSA) [166]
Controllers design	A multiobjective GA with an H_∞ loop-shaping technique [270]. An NSGA-II [271]. A novel two-lbest multiobjective particle swarm optimizer [272]
Pattern recognition and image processing	
Image processing	A bi-objective EA [273]. A hybrid algorithm combining evolutionary multiobjective optimization and gradient-based learning [274]. A bi-objective GA [275]. A multiobjective real coded genetic fuzzy clustering scheme [276]
Pattern classification	Multiobjective genetic programming [277]. An elitist multiobjective genetic algorithm (EMOGA) [278]. An MOEA based on a greedy randomized adaptive search procedure (GRASP) [169]. An NSGA-II, a SPEA2, and a PESA-II [279]. An evolutionary tri-objective optimization algorithm with the ROC convex hull method [280]. An MOEA for the unsupervised learning and data clustering problems [281]. A fuzzy clustering-based particle swarm (FCPSO) algorithm involved an external repository, niching mechanism, a self-adaptive mutation operator and a fuzzy-based feedback mechanism [133]. A novel clustering methodology termed evolutionary multiobjective conceptual clustering (EMO-CC) [282]. An MOEA approach with majority voting method followed by k-nearest-neighbor classification [283]
Artificial neural networks (ANNs) and fuzzy systems	
Neural network training	A GA-based multiobjective optimization technique [218]. A multiobjective hybrid procedures based on the SPEA2 and NSGA-II using the Baldwinian hybridization strategy [76]. A time-variant multiobjective PSO [284]
Fuzzy	An approach based on a multiobjective GA with fuzzy set theory [285]. Several MOEAs incorporating some expert evolutionary operators [286]. A variant of $(2+2)$ PAES with one-point crossover and two appropriately defined mutation operators [287]. An evolutionary tri-objective optimization algorithm integrated with the ROC convex hull method [280]. A fuzzy clustering-based particle swarm (FCPSO) algorithm [133]. A multiobjective GA [288]
Manufacturing	
Plant design	A vector evaluated artificial bee colony [289]
Production engineering	A hybrid algorithm including efficient problem-specific algorithms [109]. A fuzzy multicriteria evaluation method [290]

Table 2 (*continued*)

Areas and details	Types of MOEA applied and references
Composite components design	A multiobjective GA combined with fuzzy set theory [285]
Process plant	An NSGA and a NSGA-II for the optimal design of complex Williams–Otto model process plant [291,292]
Traffic engineering and transportation	
Traffic engineering	An MOEA for the problem of engineering the distribution of the interdomain traffic in the internet [293]
Transportation	An NSGA-II with a knee identification procedure and a multiobjective decision aid method [294]. A diversity-maintaining evolutionary multiobjective optimizer [295]
Others	
Life sciences	An NSGA-II with SBX crossover [296]
Fault diagnosis	A multiobjective artificial immune algorithm [297]
Embedded system	MOEAs with fuzzy approach [298]
Robust design	An inverse multiobjective robust evolutionary (IMORE) [219]
Chaotic system	An NSGA-II with SA-based optimization [163]
STCA system	A multiobjective $(1+1)$ evolution strategy [299]
Chanel coding	An MOEA/D for optimizing degree distributions in LT codes [38]
Phylogenetic inference	Multiobjective optimization [300]
DC motor drive	A three-step design process with a multiobjective GA, a multi-attribute decision-making process and fine tuning [301]
Multicriteria minimum spanning tree	A non-generational multiobjective GA with an efficient crossover operator by using dislocation a crossover technique and a niche evolution procedure [302]
Reservoir system operation	A self-learning genetic algorithm (SLGA) integrated with a self-organizing map (SOM), and a variable neighborhood search (VNS) [303]
Temporal process	A dynamic predictive-optimization framework that integrates data-mining algorithms and evolutionary strategy algorithms [304]
Algorithm design	A competitive and cooperative coevolutionary approach for multiobjective PSO algorithm design [86]. A viable and hybrid evolutionary-cum-local-search-based algorithm with self-adaptive population sizing and termination criteria [305]
Electric power dispatch	An NSGA, a niched Pareto GA, and a SPEA with a quality measure procedure [264]. A multiobjective fuzzy dominance-based bacterial foraging algorithm [306]
Web-site	A multicriteria GA [307]. A multiobjective grammar-based genetic programming algorithm [308]
Financial optimization	A Pareto ant colony optimization (P-ACO) [143]. A bi-objective PSO [309]. A multiobjective optimization approach based on prototype optimization with evolved improvement steps [310]

- Many objectives: The complexity of handling numerous objectives has attracted growing attention. It may not be feasible to use a finite set of solutions to approximate the whole PF of a generic many-objective problem due to the curse of dimensionality. One should develop new approaches for dealing with many objectives. Interaction with the DMs will definitely be an appropriate way. One may also need to study what problems are doable in many-objective optimization, which will be crucial for any real progress in this area.

Acknowledgements

This work is partly supported by National Basic Research Program of China (No. 2011CB707104) and National Science Foundation of China (No. 61005050).

References

[1] W. Stadler, A survey of multicriteria optimization or the vector maximum problem, part I: 1776–1960, Journal of Optimization Theory and Applications 29 (1) (1979) 1–52.
[2] K. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, 1999.
[3] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons Ltd., 2001.
[4] J.D. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, in: 1st International Conference on Genetic Algorithms, 1985, pp. 93–100.
[5] C.M. Fonseca, P.J. Fleming, An overview of evolutionary algorithms in multiobjective optimization, Evolutionary Computation 3 (1) (1995) 1–16.
[6] C.A. Coello Coello, An updated survey of GA-based multiobjective optimization techniques, ACM Computing Surveys 32 (2) (2000) 109–143.
[7] D.A. Van Veldhuizen, G.B. Lamont, Multiobjective evolutionary algorithms: analyzing the state-of-the-art, Evolutionary Computation 8 (2) (2000) 125–147.
[8] C.A. Coello Coello, A short tutorial on evolutionary multiobjective optimization, in: 1st International Conference on Evolutionary Multi-Criterion Optimization, EMO 2001, in: LNCS, vol. 1993, 2001, pp. 21–40.
[9] A. Ghosh, S. Dehuri, Evolutionary algorithms for multi-criterion optimization: a survey, International Journal of Computing & Information Sciences 2 (1) (2004) 38–57.
[10] E. Zitzler, M. Laumanns, S. Bleuler, A tutorial on evolutionary multiobjective optimization, in: Workshop on Multiple Objective Metaheuristics, MOMH 2002, in: LNEMS, vol. 535, 2004, pp. 3–37.
[11] A. Chinchuluun, P. Pardalos, A survey of recent developments in multiobjective optimization, Annals of Operations Research 154 (1) (2007) 29–50.
[12] M. Gong, L. Jiao, D. Yang, W. Ma, Research on evolutionary multi-objective optimization algorithms, Journal Software 20 (2) (2009) 271–289.
[13] C.A. Coello Coello, Recent trends in evolutionary multiobjective optimization, in: Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, Springer-Verlag, 2005, pp. 7–32.
[14] C.A. Coello Coello, G.T. Pulido, E.M. Montes, Current and future research trends in evolutionary multiobjective optimization, in: Information Processing with Evolutionary Algorithms: From Industrial Applications to Academic Speculations, Springer-Verlag, 2005, pp. 213–231.
[15] B. Suman, P. Kumar, A survey of simulated annealing as a tool for single and multiobjective optimization, Journal of the Operational Research Society 57 (10) (2006) 1143–1160.
[16] M. Reyes-Sierra, C.A. Coello Coello, Multi-objective particle swarm optimizers: a survey of the state-of-the-art, International Journal of Computational Intelligence Research 2 (3) (2006) 287–308.
[17] J.D. Knowles, D.W. Corne, Memetic algorithms for multiobjective optimization: issues, methods and prospects, in: Recent Advances in Memetic Algorithms, Springer, 2004, pp. 313–352.
[18] M. Ehrgott, X. Gandibleux, Approximative solution methods for multiobjective combinatorial optimization, Top 12 (1) (2004) 1–90.
[19] A. Jaszkiewicz, H. Ishibuchi, Q. Zhang, Multiobjective memetic algorithms, Tech. Rep., The School of Computer Science and Electronic Engineering, University of Essex, 2011.
[20] J. Andersson, A survey of multiobjective optimization in engineering design, Tech. Rep. LiTH-IKP-R-1097, Department of Mechanical Engineering, Linköping University, 2000.
[21] R.T. Marler, J.S. Arora, Survey of multi-objective optimization methods for engineering, Structural and Multidisciplinary Optimization 26 (6) (2004) 369–395.
[22] D. Lei, Multi-objective production scheduling: a survey, International Journal of Advanced Manufacturing Technology 43 (9–10) (2009) 926–938.
[23] M.G.C. Tapia, C.A. Coello Coello, Applications of multi-objective evolutionary algorithms in economics and finance: a survey, in: IEEE Congress on Evolutionary Computation, CEC 2007, 2007, pp. 532–539.

[24] F. Luna, J.J. Durillo, A.J. Nebro, E. Alba, Evolutionary algorithms for solving the automatic cell planning problem: *S* survey, Engineering Optimization 42 (7) (2010) 671–690.
[25] T. Lust, J. Teghem, The multiobjective traveling salesman problem: a survey and a new approach, in: Advances in Multi-Objective Nature Inspired Computing, in: SCI, vol. 272, 2010, pp. 119–141.
[26] C.A. Coello Coello, Handling preferences in evolutionary multiobjective optimization: a survey, in: IEEE Congress on Evolutionary Computation, CEC 2000, 2000, pp. 30–37.
[27] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.
[28] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731.
[29] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Transactions on Evolutionary Computation 13 (2) (2009) 284–302.
[30] Q. Zhang, W. Liu, H. Li, The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, Tech. Rep. CES-491, The School of Computer Science and Electronic Engineering, University of Essex, 2009.
[31] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, S. Tiwari, Multiobjective optimization test instances for the CEC 2009 special session and competition, Tech. Rep. CES-487, The School of Computer Science and Electronic Engineering, University of Essex, 2009.
[32] A.J. Nebro, J.J. Durillo, A study of the parallelization of the multi-objective metaheuristic MOEA/D, in: 4th International Conference on Learning and Intelligent Optimization, LION 4, 2010, pp. 303–317.
[33] P. Palmers, T. McConaghy, M. Steyaert, G.G.E. Gielen, Massively multi-topology sizing of analog integrated circuits, in: Conference on Design, Automation and Test in Europe, DATE 2009, 2009, pp. 706–711.
[34] H. Ishibuchi, Y. Sakane, N. Tsukamoto, Y. Nojima, Simultaneous use of different scalarizing functions in MOEA/D, in: Conference on Genetic and Evolutionary Computation, GECCO 2010, 2010, pp. 519–526.
[35] P.C. Chang, S.H. Chen, Q. Zhang, J.L. Lin, MOEA/D for flowshop scheduling problems, in: IEEE Congress on Evolutionary Computation, CEC 2008, 2008, pp. 1433–1438.
[36] S. Pal, B. Qu, S. Das, P.N. Suganthan, Optimal synthesis of linear antenna arrays with multi-objective differential evolution, Progress in Electromagnetics Research B 21 (2010) 87–111.
[37] T.J. Yuen, R. Raml, Comparison of computational efficiency of MOEA/D and NSGA-II for passive vehicle suspension optimization, in: 24th European Conference on Modelling and Simulation, ECMS 2010, 2010, pp. 219–225.
[38] C.-M. Chen, Y.-P. Chen, T.-C. Shen, J.K. Zao, Optimizing degree distributions in LT codes by using the multiobjective evolutionary algorithm based on decomposition, in: IEEE Congress on Evolutionary Computation, CEC 2010, 2010, pp. 1–8.
[39] A. Waldock, D. Corne, Multiple objective optimisation applied to route planning, in: 5th SEAS DTC Technical Conference, 2010.
[40] A. Konstantinidis, C. Charalambous, A. Zhou, Q. Zhang, Multi-objective mobile agent-based sensor network routing using MOEA/D, in: IEEE Congress on Evolutionary Computation, CEC 2010, 2010, pp. 1–8.
[41] Y.-H. Chan, T.-C. Chiang, L.-C. Fu, A two-phase evolutionary algorithm for multiobjective mining of classification rules, in: IEEE Congress on Evolutionary Computation, CEC 2010, 2010, pp. 1–7.
[42] Y. Mei, K. Tang, X. Yao, Decomposition-based memetic algorithm for multi-objective capacitated arc routing problem, IEEE Transactions on Evolutionary Computation (2010), in press (doi:10.1109/TEVC.2010.2051446).
[43] C.M. Fonseca, P.J. Fleming, Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, in: 5th International Conference Genetic Algorithms, 1993, pp. 416–423.
[44] T. Tanino, M. Tanaka, C. Hojo, An interactive multicriteria decision making method by using a genetic algorithm, in: 2nd International Conference on Systems Science and Systens Engineering, 1993, pp. 381–386.
[45] G. Greenwood, X. Hu, J. D'Ambrosio, Fitness functions for multiple objective optimization problems: combining preferences with Pareto rankings, in: Foundations of Genetic Algorithms, vol. 4, 1997, pp. 437–455.
[46] M. Sakawa, K. Kato, An interactive fuzzy satisficing method for general multiobjective 0-1 programming problems through genetic algorithms with double strings based on a reference solution, Fuzzy Sets and Systems 125 (3) (2002) 289–300.
[47] S. Phelps, M. Köksalan, An interactive evolutionary metaheuristic for multiobjective combinatorial optimization, Management Science 49 (2003) 1726–1738.
[48] J. Branke, K. Deb, Integrating user preferences into evolutionary multi-objective optimization, Tech. Rep. KanGAL 2004, Indian Institute of Technology, 2004.
[49] K. Deb, J. Sundar, N.U.B. Rao, S. Chaudhuri, Reference point based multi-objective optimization using evolutionary algorithms, International Journal of Computational Intelligence Research 2 (3) (2006) 273–286.
[50] K. Deb, A. Kumar, Interactive evolutionary multi-objective optimization and decision-making using reference direction method, Tech. Rep., Indian Institute of Technology, KanGAL 2007001, 2007.
[51] K. Deb, S. Chaudhuri, I-EMO: an interactive evolutionary multi-objective optimization tool, in: 1st International Conference on Pattern Recognition and Machine Intelligence, PReMI 2005, 2005, pp. 690–695.
[52] H. Li, D.L. Silva, Evolutionary multi-objective simulated annealing with adaptive and competitive search direction, in: IEEE Congress on Evolutionary Computation, CEC 2008, 2008, pp. 3310–3317.
[53] J. Sanchis, M.A. Martínez, X.B. Ferragud, Integrated multiobjective optimization and a priori preferences using genetic algorithms, Information Sciences 178 (4) (2008) 931–951.
[54] K. Deb, A. Sinha, P. Korhonen, J. Wallenius, An interactive evolutionary multi-objective optimization method based on progressively approximated value functions, Tech. Rep. KanGAL 2009005, Indian Institute of Technology, 2009.
[55] L. Rachmawati, D. Srinivasan, Multiobjective evolutionary algorithm with controllable focus on the knees of the Pareto front, IEEE Transactions on Evolutionary Computation 13 (4) (2009) 810–824.
[56] L. Thiele, K. Miettinen, P.J. Korhonen, J.M. Luque, A preference-based evolutionary algorithm for multi-objective optimization, Evolutionary Computation 17 (3) (2009) 411–436.
[57] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: Parallel Problem Solving from Nature, PPSN VIII, in: LNCS, vol. 3242, 2004, pp. 832–842.
[58] M. Basseur, E. Zitzler, Handling uncertainty in indicator-based multiobjective optimization, International Journal of Computational Intelligence Research 2 (3) (2006) 255–272.
[59] D. Brockhoff, E. Zitzler, Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods, in: IEEE Congress on Evolutionary Computation, CEC 2007, 2007, pp. 2086–2093.
[60] J. Bader, E. Zitzler, HypE: an algorithm for fast hypervolume-based many-objective optimization, Tech. Rep. TIK 286, Computer Engineering and Networks Laboratory, ETH Zurich, 2008.
[61] J. Bader, E. Zitzler, Robustness in hypervolume-based multiobjective search, Tech. Rep. TIK 317, Computer Engineering and Networks Laboratory, ETH Zurich, 2010.
[62] A. Elhossini, S. Areibi, R. Dony, Strength Pareto particle swarm optimization and hybrid EA-PSO for multi-objective optimization, Evolutionary Computation 18 (1) (2010) 127–156.
[63] B.B. Li, L. Wang, A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling, IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 37 (3) (2007) 576–591.
[64] D. Yang, L. Jiao, M. Gong, Adaptive multi-objective optimization based on nondominated solutions, Computational Intelligence 25 (2) (2009) 84–108.
[65] A. Lara, G. Sanchez, C.A. Coello Coello, O. Schutze, HCS: a new local search strategy for memetic multiobjective evolutionary algorithms, IEEE Transactions on Evolutionary Computation 14 (1) (2010) 112–132.
[66] H. Ishibuchi, T. Murata, A multiobjective genetic local search algorithm and its application to flowshop scheduling, IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews 28 (3) (1998) 392–403.
[67] A. Jaszkiewicz, Do multiple-objective metaheuristics deliver on their promises? A computational experiment on the set-covering problem, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 133–143.
[68] B. Qian, L. Wang, D. Huang, X. Wang, Multi-objective no-wait flow-shop scheduling with a memetic algorithm based on differential evolution, Soft Computing 13 (8–9) (2009) 847–869.
[69] J. Chen, Q. Lin, Z. Ji, A hybrid immune multiobjective optimization algorithm, European Journal of Operational Research 204 (2) (2010) 294–302.
[70] W.-F. Leong, G.G. Yen, PSO-based multiobjective optimization with dynamic population size and adaptive local archives, IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 38 (5) (2008) 1270–1293.
[71] R. Caballero, M. González, F.M. Guerrero, J.M. Luque, C. Paralera, Solving a multiobjective location routing problem with a metaheuristic based on Tabu search. application to a real case in Andalusia, European Journal of Operational Research 177 (3) (2007) 1751–1763.
[72] E.F. Wanner, F.G. Guimarães, R.H.C. Takahashi, P.J. Fleming, Local search with quadratic approximations into memetic algorithms for optimization with multiple criteria, Evolutionary Computation 16 (2) (2008) 185–224.
[73] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, Y. Nojima, Use of biased neighborhood structures in multiobjective memetic algorithms, Soft Computing 13 (8–9) (2009) 795–810.
[74] S.F. Adra, T.J. Dodd, I.A. Griffin, P.J. Fleming, Convergence acceleration operator for multiobjective optimization, IEEE Transactions on Evolutionary Computation 13 (4) (2009) 825–847.
[75] Y. Wang, Z. Cai, G. Guo, Z. Zhou, Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems, IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 37 (3) (2007) 560–575.
[76] M. Delgado, M.P. Cuéllar, M.C. Pegalajar, Multiobjective hybrid optimization and training of recurrent neural networks, IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 38 (2) (2008) 381–403.
[77] P. Koduru, Z. Dong, S. Das, S. Welch, J.L. Roe, E. Charbit, A multiobjective evolutionary-simplex hybrid approach for the optimization of differential equation models of gene networks, IEEE Transactions on Evolutionary Computation 12 (5) (2008) 572–590.
[78] J. Knowles, D. Corne, M-PAES: a memetic algorithm for multiobjective optimization, in: IEEE Congress on Evolutionary Computation, CEC 2000, 2000, pp. 325–332.
[79] A. Jaszkiewicz, On the performance of multiple-objective genetic local search on the 0/1 knapsack problem—a comparative experiment, IEEE Transactions on Evolutionary Computation 6 (4) (2002) 402–412.

[80] A. Caponio, F. Neri, Integrating cross-dominance adaptation in multi-objective memetic algorithms, in: Multiobjective Memetic Algorithms, in: SCI, vol. 171, Springer-Verlag, 2009, pp. 325–351.
[81] O. Soliman, L.T. Bui, H. Abbass, A memetic coevolutionary multiobjective differential evolution algorithm, in: Multiobjective Memetic Algorithms, in: SCI, vol. 171, Springer-Verlag, 2009, pp. 369–388.
[82] H. Li, D.L. Silva, An elitist GRASP metaheuristic for the multi-objective quadratic assignment problem, in: 5th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2009, 2009, pp. 481–494.
[83] K. Deb, M. Mohan, S. Mishra, Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions, Evolutionary Computation 13 (4) (2005) 501–525.
[84] K.C. Tan, Y.J. Yang, C.K. Goh, A distributed cooperative coevolutionary algorithm for multiobjective optimization, IEEE Transactions on Evolutionary Computation 10 (5) (2006) 527–549.
[85] C.K. Goh, K.C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization, IEEE Transactions on Evolutionary Computation 13 (1) (2009) 103–127.
[86] C.K. Goh, K.C. Tan, D.S. Liu, S.C. Chiam, A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design, European Journal of Operational Research 202 (1) (2010) 42–54.
[87] N. Srinivas, K. Deb, Multiobjective optimization using nondominated sorting in genetic algorithms, Evolutionary Computation 2 (3) (1994) 221–248.
[88] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Transactions on Evolutionary Computation 3 (4) (1999) 257–271.
[89] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, in: Evolutionary Methods for Design Optimisation and Control, 2002, pp. 95–100.
[90] B.Y. Qu, P.N. Suganthan, Multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection, Information Sciences 180 (17) (2010) 3170–3181.
[91] J.D. Knowles, D.W. Corne, Approximating the nondominated front using the Pareto archived evolution strategy, Evolutionary Computation 8 (2) (2000) 149–172.
[92] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, Combining convergence and diversity in evolutionary multiobjective optimization, Evolutionary Computation 10 (3) (2002) 263–282.
[93] G.G. Yen, H. Lu, Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation, IEEE Transactions on Evolutionary Computation 7 (3) (2003) 253–274.
[94] C.A. Coello Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 256–279.
[95] S.Z. Zhao, P.N. Suganthan, Multi-objective evolutionary algorithm with ensemble of external archives, International Journal of Innovative Computing, Information and Control 6 (1) (2010) 1713–1726.
[96] M. Gong, L. Jiao, H. Du, L. Bo, Multiobjective immune algorithm with nondominated neighbor-based selection, Evolutionary Computation 16 (2) (2008) 225–255.
[97] B. Soylu, M. Koksalan, A favorable weight-based evolutionary algorithm for multiple criteria problems, IEEE Transactions on Evolutionary Computation 14 (2) (2010) 191–205.
[98] Y. Wang, L.-H. Wu, X. Yuan, Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure, Soft Computing 14 (3) (2010) 193–209.
[99] B. Panigrahi, V.R. Pandi, S. Das, S. Das, Multiobjective fuzzy dominance based bacterial foraging algorithm to solve economic emission dispatch problem, Energy 35 (12) (2010) 4761–4770.
[100] D. Kundu, K. Suresh, S. Ghosh, S. Das, B. Panigrahi, S. Das, Multi-objective optimization with artificial weed colonies, Original Research Article Information Sciences 181 (12) (2011) 2441–2454.
[101] H. Fang, Q. Wang, Y.-C. Tu, M.F. Horstemeyer, An efficient non-dominated sorting method for evolutionary algorithms, Evolutionary Computation 16 (3) (2008) 355–384.
[102] C. Shi, Z. Yan, Z. Shi, L. Zhang, A fast multi-objective evolutionary algorithm based on a tree structure, Applied Soft Computing 10 (2) (2010) 468–480.
[103] M. Fleischer, The measure of Pareto optima applications to multi-objective metaheuristics, in: 2nd International Conference on Evolutionary Multi-Criterion Optimization, EMO 2003, in: LNCS, vol. 2632, 2003, pp. 519–533.
[104] S. Huband, P. Hingston, L. White, L. Barone, An evolution strategy with probabilistic mutation for multi-objective optimisation, in: IEEE Congress on Evolutionary Computation, CEC 2003, 2003, pp. 2284–2291.
[105] B. Naujoks, N. Beume, M. Emmerich, Multi-objective optimisation using s-metric selection: application to three-dimensional solution spaces, in: IEEE Congress on Evolutionary Computation, CEC 2005, vol. 2, 2005, pp. 1282–1289.
[106] C. Igel, N. Hansen, S. Roth, Covariance matrix adaptation for multi-objective optimization, Evolutionary Computation 15 (1) (2007) 1–28.
[107] A.W. Iorio, X. Li, Rotated problems and rotationally invariant crossover in evolutionary multi-objective optimization, International Journal of Computational Intelligence and Applications 7 (2) (2008) 149–186.
[108] S. Zeng, L. Kang, L. Ding, An orthogonal multi-objective evolutionary algorithm for multi-objective optimization problems with constraints, Evolutionary Computation 12 (1) (2004) 77–98.
[109] K. Weinert, A. Zabel, P. Kersting, T. Michelitsch, T. Wagner, On the use of problem-specific candidate generators for the hybrid optimization of multi-objective production engineering problems, Evolutionary Computation 17 (4) (2009) 527–544.
[110] Q. Zhang, A. Zhou, Y. Jin, RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm, IEEE Transactions on Evolutionary Computation 12 (1) (2008) 41–63.
[111] A. Zhou, Q. Zhang, Y. Jin, Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm, IEEE Transactions on Evolutionary Computation 13 (5) (2009) 1167–1189.
[112] R. Storn, K. Price, Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces, Tech. Rep. TR-95-012, ICSI, 1995.
[113] K.V. Price, An introduction to differential evolution, in: New Ideas in Optimization, McGraw-Hill, 1999, pp. 79–108.
[114] R. Sarker, H. Abbass, Differential evolution for solving multi-objective optimization problems, Asia Pacific Journal of Operational Research 21 (2) (2004) 225–240.
[115] W. Gong, Z. Cai, An improved multiobjective differential evolution based on Pareto-adaptive epsilon-dominance and orthogonal design, European Journal of Operational Research 198 (2) (2009) 576–601.
[116] B.Y. Qu, P.N. Suganthan, Multi-objective differential evolution with diversity enhancement, Journal of Zhejiang University Science A 11 (2010) 538–543.
[117] B. Alatas, E. Akin, A. Karci, MODENAR: multi-objective differential evolution algorithm for mining numeric association rules, Applied Soft Computing 8 (1) (2008) 646–656.
[118] T. Fukuda, K. Mori, M. Tsukiyama, Immune networks using genetic algorithm for adaptive production scheduling, in: 15th IFAC World Congress, vol. 3, 1993, pp. 57–60.
[119] C.A. Coello Coello, N.C. Cortés, Solving multiobjective optimization problems using an artificial immune system, Genetic Programming and Evolvable Machines 6 (2) (2005) 163–190.
[120] R. Tavakkoli-Moghaddam, A. Rahimi-Vahed, A.H. Mirzaei, A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: weighted mean completion time and weighted mean tardiness, Information Sciences 177 (22) (2007) 5072–5090.
[121] Z.-H. Hu, A multiobjective immune algorithm based on a multiple-affinity model, European Journal of Operational Research 202 (1) (2010) 60–72.
[122] Z. Zhang, Immune optimization algorithm for constrained nonlinear multiobjective optimization problems, Applied Soft Computing 7 (3) (2007) 840–857.
[123] Z. Zhang, Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control, Applied Soft Computing 8 (2) (2008) 959–971.
[124] X. Zuo, H. Mo, J. Wu, A robust scheduling method based on a multi-objective immune algorithm, Information Sciences 179 (19) (2009) 3359–3369.
[125] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: 6th International Symposium on Micromachine and Human Science, 1995, pp. 39–43.
[126] J. Kennedy, R.C. Eberhart, Y. Shi, Swarm Intelligence, Morgan Kaufmann, San Francisco, US, 2001.
[127] J. Moore, R. Chapman, Application of particle swarm to multiobjective optimization, Tech. Rep., Department of Computer Science and Software Engineering, Auburn University, 1999.
[128] S. Janson, D. Merkle, M. Middendorf, Molecular docking with multi-objective particle swarm optimization, Applied Soft Computing 8 (1) (2008) 666–675.
[129] D.S. Liu, K.C. Tan, S.Y. Huang, C.K. Goh, W.K. Ho, On solving multiobjective bin packing problems using evolutionary particle swarm optimization, European Journal of Operational Research 190 (2) (2008) 357–382.
[130] P.K. Tripathi, S. Bandyopadhyay, S.K. Pal, Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients, Information Sciences 177 (22) (2007) 5033–5049.
[131] Y. Wang, Y. Yang, Particle swarm optimization with preference order ranking for multi-objective optimization, Information Sciences 179 (12) (2009) 1944–1959.
[132] A. Rahimi-Vahed, S.M. Mirghorbani, M. Rabbani, A new particle swarm algorithm for a multi-objective mixed-model assembly line sequencing problem, Soft Computing 11 (10) (2007) 997–1012.
[133] S. Agrawal, B.K. Panigrahi, M.K. Tiwari, Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch, IEEE Transactions on Evolutionary Computation 12 (5) (2008) 529–541.
[134] V.L. Huang, P.N. Suganthan, J.J. Liang, Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems, International Journal of Intelligent Systems 21 (2) (2006) 209–226.
[135] S.Z. Zhao, P.N. Suganthan, Two-lbests based multi-objective particle swarm optimizer, Engineering Optimization 43 (1) (2011) 1–17.
[136] N.A. Moubayed, A. Petrovski, J. McCall, A novel smart multi-objective particle swarm optimisation using decomposition, in: Parallel Problem Solving from Nature, PPSN XI, in: LNCS, vol. 6239, 2010, pp. 1–10.
[137] M. Dorigo, T. Stützle, Ant Colony Optimization, MIT Press, 2004.
[138] D. Angus, Crowding population-based ant colony optimisation for the multi-objective travelling salesman problem, in: IEEE Symposium on Computational Intelligence in Multicriteria Decision Making, MCDM 2007, 2007, pp. 333–340.

[139] C. Garcia-Martinez, O. Cordon, F. Herrera, A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP, European Journal of Operational Research 127 (1) (2007) 116–148.
[140] D.M. Chitty, M.L. Hernandez, A hybrid ant colony optimisation technique for dynamic vehicle routing, in: Conference on Genetic and Evolutionary Computation, GECCO 2004, in: LNCS, vol. 3102, 2004, pp. 48–59.
[141] J.M. Pasia, R.F. Hartl, K.F. Doerner, Solving a bi-objective flowshop scheduling problem by Pareto-ant colony optimization, in: 5th International Workshop on Ant Colony Optimization and Swarm Intelligence, ANTS 2006, 2006, pp. 294–305.
[142] K. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, C. Stummer, Pareto ant colony optimization: a metaheuristic approach to multiobjective portfolio selection, Annals of Operations Research 131 (1–4) (2004) 79–99.
[143] K.F. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, C. Stummer, Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection, European Journal of Operational Research 171 (3) (2006) 830–841.
[144] R.Y. Rubinstein, D.P. Kroese, The Cross-Entropy Method: A Unified Approach to Monte Carlo Simulation, Randomized Optimization and Machine Learning, Springer Verlag, 2004.
[145] A. Unveren, A. Acan, Multi-objective optimization with cross entropy method: Stochastic learning with clustered Pareto fronts, in: IEEE Congress on Evolutionary Computation, CEC 2007, 2007, pp. 3065–3071.
[146] K.-H. Han, J.-H. Kim, Genetic quantum algorithm and its application to combinatorial optimization problem, in: IEEE Congress on Evolutionary Computation, CEC 2000, 2000, pp. 1354–1360.
[147] K.-H. Han, J.-H. Kim, Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, IEEE Transactions on Evolutionary Computation 6 (2002) 580–593.
[148] W. Wei, B. Li, Y. Zou, W. Zhang, Z. Zhuang, A multi-objective hw-sw co-synthesis algorithm based on quantum-inspired evolutionary algorithm, International Journal of Computational Intelligence and Applications 7 (2) (2008) 129–148.
[149] H. Mühlenbein, G. Paaß, From recombination of genes to the estimation of distributions I: binary parameters, in: Parallel Problem Solving from Nature, PPSN IV, in: LNCS, vol. 1411, 1996, pp. 178–187.
[150] P. Larrañaga, J.A. Lozano (Eds.), Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Kluwer Academic Publishers, 2002.
[151] T. Okabe, Y. Jin, B. Sendhoff, M. Olhofer, Voronoi-based estimation of distribution algorithm for multi-objective optimization, in: IEEE Congress on Evolutionary Computation, CEC 2004, 2004, pp. 1594–1601.
[152] P.A.N. Bosman, D. Thierens, The naive MIDEA: A baseline multi-objective EA, in: 3rd International Conference on Evolutionary Multi-Criterion Optimization, EMO 2005, in: LNCS, vol. 3410, 2005, pp. 428–442.
[153] W. Dong, X. Yao, Unified eigen analysis on multivariate Gaussian based estimation of distribution algorithms, Information Sciences 178 (15) (2008) 3000–3023.
[154] M. Laumanns, J. Očenášek, Bayesian optimization algorithms for multi-objective optimization, in: Parallel Problem Solving From Nature, PPSN VII, in: LNCS, vol. 2439, 2002, pp. 298–307.
[155] M. Pelikan, K. Sastry, D. Goldberg, Multiobjective hBOA, clustering, and scalability, in: Conference on Genetic and Evolutionary Computation, GECCO 2005, Vol. 2, 2005, pp. 663–670.
[156] A. Zhou, Q. Zhang, Y. Jin, B. Sendhoff, E. Tsang, Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization, in: 4th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2007, in: LNCS, vol. 4403, 2007, pp. 832–846.
[157] A. Zhou, Q. Zhang, Y. Jin, B. Sendhoff, E. Tsang, Global multiobjective optimization via estimation of distribution algorithm with biased initialization and crossover, in: Conference on Genetic and Evolutionary Computation, GECCO 2007, 2007, pp. 617–622.
[158] Y. Jin, A. Zhou, Q. Zhang, B. Sendhoff, E. Tsang, Modeling regularity to improve scalability of model-based multiobjective optimization algorithms, in: Multobjective Problem Solving From Nature, Springer, 2008, pp. 331–355.
[159] L. Mo, G. Dai, J. Zhu, The RM-MEDA based on elitist strategy, in: 5th International Conference on Advances in Computation and Intelligence, ISICA 2010, 2010, pp. 229–239.
[160] A.K.A. Talukder, M. Kirley, R. Buyya, A Pareto following variation operator for fast-converging multiobjective evolutionary algorithms, in: Conference on Genetic and Evolutionary Computation, GECCO 2008, 2008, pp. 721–729.
[161] D. Yang, L. Jiao, M. Gong, H. Feng, Hybrid multiobjective estimation of distribution algorithm by local linear embedding and an immune inspired algorithm, in: IEEE Congress on Evolutionary Computation, CEC 2009, 2009, pp. 463–470.
[162] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[163] L. Sánchez, J.R. Villar, Obtaining transparent models of chaotic systems with multi-objective simulated annealing algorithms, Information Sciences 178 (4) (2008) 952–970.
[164] K.I. Smith, R.M. Everson, J.E. Fieldsend, C. Murphy, R. Misra, Dominance-based multiobjective simulated annealing, IEEE Transactions on Evolutionary Computation 12 (3) (2008) 323–342.
[165] S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb, A simulated annealing-based multiobjective optimization algorithm: AMOSA, IEEE Transactions on Evolutionary Computation 12 (3) (2008) 269–283.
[166] E. Aggelogiannaki, H. Sarimveis, A simulated annealing algorithm for prioritized multiobjective optimization-implementation in an adaptive model predictive control configuration, IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 37 (4) (2007) 902–915.
[167] L. Belfares, W. Klibi, N. Lo, A. Guitouni, Multi-objectives Tabu search based algorithm for progressive resource allocation, European Journal of Operational Research 177 (3) (2007) 1779–1799.
[168] R.P. Beausoleil, MOSS: Multiobjective scatter search applied to non-linear multiple criteria optimization, European Journal of Operational Research 169 (2) (2006) 426–449.
[169] A.P. Reynolds, B. de la Iglesia, A multi-objective GRASP for partial classification, Soft Computing 13 (3) (2009) 227–243.
[170] M. Laumanns, L. Thiele, E. Zitzler, Running time analysis of multiobjective evolutionary algorithms on pseudo-Boolean functions, IEEE Transactions on Evolutionary Computation 8 (2) (2004) 170–182.
[171] F. Xue, A.C. Sanderson, R.J. Graves, Modeling and convergence analysis of a continuous multi-objective differential evolution algorithm, in: IEEE Congress on Evolutionary Computation, CEC 2005, vol. 1, 2005, pp. 228–235.
[172] F. Xue, A.C. Sanderson, R.J. Graves, Multi-objective differential evolution — algorithm, convergence analysis, and applications, in: IEEE Congress on Evolutionary Computation, CEC 2005, Vol. 1, 2005, pp. 228–235.
[173] H. Trautmann, T. Wagner, B. Naujoks, M. Preuß, J. Mehnen, Statistical methods for convergence detection of multi-objective evolutionary algorithms, Evolutionary Computation 17 (4) (2009) 493–509.
[174] P. Chakraborty, S. Das, G.G. Roy, A. Abraham, On convergence of the multi-objective particle swarm optimizers, Information Sciences 181 (8) (2011) 1411–1425.
[175] E.I. Ducheyne, B.D. Baets, R.R.D. Wulf, Fitness inheritance in multiple objective evolutionary algorithms: a test bench and real-world evaluation, Applied Soft Computing 8 (1) (2008) 337–349.
[176] S. Zhao, L. Jiao, Multi-objective evolutionary design and knowledge discovery of logic circuits based on an adaptive genetic algorithm, Genetic Programming and Evolvable Machines 7 (3) (2006) 195–210.
[177] K.C. Tan, C.K. Goh, Y.J. Yang, T.H. Lee, Evolving better population distribution and exploration in evolutionary multi-objective optimization, European Journal of Operational Research 171 (2) (2006) 463–495.
[178] V.L. Huang, S.Z. Zhao, R. Mallipeddi, P.N. Suganthan, Multi-objective optimization using self-adaptive differential evolution algorithm, in: IEEE Congress on Evolutionary Computation, CEC 2009, 2009, pp. 190–194.
[179] Z. Michalewicz, M. Schoenauer, Evolutionary algorithms for constrained parameter optimization problems, Evolutionary Computation 4 (1) (1996) 1–32.
[180] C.A. Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Computer Methods in Applied Mechanics and Engineering 191 (11–12) (2002) 1245–1287.
[181] S. Das, B. Natarajan, D. Stevens, P. Koduru, Multi-objective and constrained optimization for DS-CDMA code design based on the clonal selection principle, Applied Soft Computing 8 (1) (2008) 788–797.
[182] Z. Cai, Y. Wang, A multiobjective optimization-based evolutionary algorithm for constrained optimization, IEEE Transactions on Evolutionary Computation 10 (6) (2006) 658–675.
[183] Y.G. Woldesenbet, G.G. Yen, B.G. Tessema, Constraint handling in multiobjective evolutionary optimization, IEEE Transactions on Evolutionary Computation 13 (3) (2009) 514–525.
[184] R. Courant, Variational methods for the solution of problems of equilibrium and vibrations, Bulletin of the American Mathematical Society 49 (1) (1943) 1–23.
[185] Y.W. Leung, Y.P. Wang, Multi-objective programming using uniform design and genetic algorithm, IEEE Transactions on Systems, Man, and Cybernetics, Part C 30 (3) (2000) 293–304.
[186] B.Y. Qu, P.N. Suganthan, Constrained multi-objective optimization algorithm with ensemble of constraint handling methods, Engineering Optimization 43 (4) (2011) 403–416.
[187] B. Liu, F.V. Fernandez, P. Gao, G. Gielen, A fuzzy selection based constrained handling method for multi-objective optimization of analog cells, in: European Conference on Circuit Theory and Design, 2009, pp. 611–614.
[188] H.T. Geng, Q.X. Song, T.T. Wu, J.F. Liu, A multi-objective constrained optimization algorithm based on infeasible individual stochastic binary-modification, in: IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS 2009, 2009, pp. 89–93.
[189] A.H. Aguirre, S.B. Rionda, C.A. Coello Coello, G.L. Lizáraga, E. Mezura-Montes, Handling constraints using multiobjective optimization concepts, International Journal for Numerical Methods in Engineering 59 (15) (2004) 1989–2017.
[190] E. Mezura-Montes, C.A. Coello Coello, A numerical comparison of some multiobjective-based techniques to handle constraints in genetic algorithms, Tech. Rep. EVOCINV-01-2003, Evolutionary Computation Group, CINVESTAV, Sección de Computación, Dept. de Ingeniería Eléctrica, CINVESTAV-IPN, México DF, México, 2003.
[191] J. Yao, N. Kharma, P. Grogono, Bi-objective multipopulation genetic algorithm for multimodal function optimization, IEEE Transactions on Evolutionary Computation 14 (1) (2010) 80–102.
[192] K. Deb, A. Saha, Multimodal optimization using a bi-objective evolutionary algorithm, Tech. Rep. KanGAL 2009006, Indian Institute of Technology, 2009.

[193] K. Deb, A. Saha, Finding multiple solutions for multimodal optimization problems using a multi-objective evolutionary approach, in: Conference on Genetic and Evolutionary Computation, GECCO 2010, 2010, pp. 447–454.

[194] X. Zou, Y. Chen, M. Liu, L. Kang, A new evolutionary algorithm for solving many-objective optimization problems, IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 38 (5) (2008) 1402–1412.

[195] R.C. Purshouse, P.J. Fleming, On the evolutionary optimization of many conflicting objectives, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 770–784.

[196] H. Sato, H.E. Aguirre, K. Tanaka, Controlling dominance area of solutions and its impact on the performance of MOEAs, in: 4th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2007, in: LNCS, vol. 4403, 2007, pp. 5–20.

[197] D. Corne, J.D. Knowles, Techniques for highly multiobjective optimisation: Some nondominated points are better than others, in: Conference on Genetic and Evolutionary Computation, GECCO 2007, 2007, pp. 773–780.

[198] K.C. Giannakoglou, Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence, Progress in Aerospace Sciences 38 (1) (2002) 43–76.

[199] Z.S. Davies, R.J. Gilbert, R.J. Merry, D.B. Kell, M.K. Theodorou, G.W. Griffith, Efficient improvement of silage additives by using genetic algorithms, Applied and Environmental Microbiology 66 (4) (2000) 1435–1443.

[200] 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.

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

[202] B. Yang, Y.-S. Yeun, W.-S. Ruy, Manageing approximation models in multiobjective optimization, Structural and Multidisciplinary Optimization 24 (2) (2002) 141–156.

[203] H.-S. Chung, J.J. Alonso, Multiobjective optimization using approximation model based genetic algorithms, in: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2004, pp. 2004–4325.

[204] S. Jeong, Efficient global optimization (EGO) for multi-objective problem and data mining, in: IEEE Congress on Evolutionary Computation, CEC 2005, vol. 3, 2005, pp. 2138–2145.

[205] J. Keane, Statistical improvement criteria for use in multi-objective design optimization, AIAA Journal 44 (4) (2006) 879–891.

[206] M.K. Karakasis, K.C. Giannakoglou, On the use of metamodel-assisted, multiobjective evolutionary algorithms, Engineering Optimization 38 (8) (2006) 941–957.

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

[208] Z. Bingul, Adaptive genetic algorithms applied to dynamic multiobjective problems, Applied Soft Computing 7 (3) (2007) 791–799.

[209] M.A. Abo-Sinna, Multiple objective (fuzzy) dynamic programming problems: a survey and some application, Applied Mathematics and Computation 157 (3) (2004) 861–888.

[210] Y. Jin, J. Branke, Evolutionary optimization in uncertain environments—a survey, IEEE Transactions on Evolutionary Computation 9 (3) (2005) 303–317.

[211] L. Bui, H. Abbass, J. Branke, Multiobjective optimization for dynamic environments, in: IEEE Congress on Evolutionary Computation, CEC 2005, vol. 3, 2005, pp. 2349–2356.

[212] M. Farina, K. Deb, P. Amato, Dynamic multiobjective optimization problems: test cases, approximations, and applications, IEEE Transactions on Evolutionary Computation 8 (5) (2004) 425–442.

[213] K. Mitra, S. Majumdar, S. Raha, Multiobjective dynamic optimization of a semi-batch epoxy polymerization process, Computers and Chemical Engineering 28 (12) (2004) 2583–2594.

[214] S. Palaniappan, S. Zein-Sabatto, A. Sekmen, Dynamic multiobjective optimization of war resource allocation using adaptive genetic algorithms, in: IEEE Southeast Conference, 2002, pp. 160–165.

[215] Y. Jin, M. Olhofer, B. Sendhoff, Dynamic weighted aggregation for evolutionary multi-objective optimization: why does it work and how?, in: Conference on Genetic and Evolutionary Computation, GECCO 2001, 2001, pp. 1042–1049.

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

[217] C.K. Goh, K.C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multi-objective optimization, IEEE Transaction on Evolutionary Computation 13 (2009) 103–127.

[218] F. Pettersson, N. Chakraborti, H. Saxén, A genetic algorithms based multiobjective neural net applied to noisy blast furnace data, Applied Soft Computing 7 (1) (2007) 387–397.

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

[220] K. Deb, H. Gupta, Introducing robustness in multi-objective optimization, Evolutionary Computation 14 (4) (2006) 463–494.

[221] L.T. Bui, H.A. Abbass, D. Essam, Localization for solving noisy multi-objective optimization problems, Evolutionary Computation 17 (3) (2009) 379–409.

[222] C.K. Goh, K.C. Tan, An investigation on noisy environments in evolutionary multiobjective optimization, IEEE Transactions on Evolutionary Computation 11 (3) (2007) 354–381.

[223] A. Syberfeldt, A. Ng, R.I. John, P. Moore, Evolutionary optimisation of noisy multi-objective problems using confidence-based dynamic resampling, European Journal of Operational Research 204 (3) (2010) 533–544.

[224] E.J. Hughes, Evolutionary multi-objective ranking with uncertainty and noise, in: 1st International Conference on Evolutionary Multi-Criterion Optimization, EMO 2001, in: LNCS, vol. 1993, 2001, pp. 329–343.

[225] P.C. Chang, S.H. Chen, K.L. Lin, Two-phase sub population genetic algorithm for parallel machine-scheduling problem, Expert Systems with Applications 29 (3) (2005) 705–712.

[226] J.K. Cochran, S.M. Horng, J.W. Fowler, A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines, Computers and Operations Research 30 (7) (2003) 1087–1102.

[227] P.C. Chang, S.H. Chen, C.H. Liu, Sub-population genetic algorithm with mining gene structures for multi-objective flowshop scheduling problems, Expert Systems with Applications 33 (3) (2007) 762–771.

[228] H. Ishibuchi, K. Narukawa, N. Tsukamoto, Y. Nojima, An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization, European Journal of Operational Research 188 (1) (2008) 57–75.

[229] L.-N. Xing, Y.-W. Chen, K.-W. Yang, Multi-objective flexible job shop schedule: design and evaluation by simulation modeling, Applied Soft Computing 9 (1) (2009) 362–376.

[230] P.-C. Chang, S.-H. Chen, The development of a sub-population genetic algorithm II (SPGA II) for multi-objective combinatorial problems, Applied Soft Computing 9 (1) (2009) 173–181.

[231] V.L. Huang, A.K. Qin, K. Deb, E. Zitzler, P.N. Suganthan, J.J. Liang, M. Preuss, S. Huband, Problem definitions for performance assessment on multi-objective optimization algorithms, Tech. Rep., Nanyang Technological University, Singapore, 2007.

[232] S. Huband, P. Hingston, L. Barone, R.L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Transactions on Evolutionary Computation 10 (5) (2006) 477–506.

[233] J. Knowles, L. Thiele, E. Zitzler, A tutorial on the performance assessment of stochastic multiobjective optimizers, Tech. Rep. TIK-Report No.214, Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland, 2006.

[234] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G. Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 117–131.

[235] H. Esbensen, E.S. Kuh, Design space exploration using the genetic algorithm, in: IEEE Symposium on Circuits and Systems, ISCAS 1996, vol. 4, 1996, pp. 500–503.

[236] P. Czyzak, A. Jaszkiewicz, Pareto simulated annealing—a metaheuristic for multiobjective combinatorial optimization, Journal of Multi-Criteria Decision Analysis 7 (1) (1998) 34–47.

[237] E.L. Ulungu, J. Teghem, P. Fortemps, D. Tuyttens, MOSA method: a tool for solving multiobjective combinatorial optimization problems, Journal of Multi-Criteria Decision Analysis 8 (4) (1999) 221–236.

[238] D.A. van Veldhuizen, Multiobjective evolutionary algorithms: classifications, analyses, and new innovations, Ph.D. Thesis, Graduate School of Engineering of the Air Force Institute of Technology, Air University, Wright-Patterson AFB, OH, 1999.

[239] J. Schott, Fault tolerant design using single and multicriteria genetic algorithm optimization, Master's Thesis, Department of Aeronautics and Astronautics, MIT, Cambridge, MA, US, 1995.

[240] E. Zitzler, Evolutionary algorithms for multiobjective optimization: methods and applications, Ph.D. Thesis, Swiss Federal Institute of Technology, Zurich, Switzerland, 1999.

[241] S. Sayin, Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming, Mathematical Programming 87 (3) (2000) 543–560.

[242] J. Wu, S. Azarm, Metrics for quality assessment of a multiobjective design optimization solution set, Journal of Mechanical Design 123 (1) (2001) 18–25.

[243] R. Sarker, T. Ray, An improved evolutionary algorithm for solving multiobjective crop planning models, Computers and Electronics in Agriculture 68 (2009) 191–199.

[244] M. Saadatseresht, A. Mansourian, M. Taleai, Evacuation planning using multiobjective evolutionary optimization approach, European Journal of Operational Research 198 (1) (2009) 305–314.

[245] P.-C. Chang, J.-C. Hsieh, C.-Y. Wang, Adaptive multi-objective genetic algorithms for scheduling of drilling operation in printed circuit board industry, Applied Soft Computing 7 (3) (2007) 800–806.

[246] T. Hanne, S. Nickel, A multiobjective evolutionary algorithm for scheduling and inspection planning in software development projects, European Journal of Operational Research 167 (3) (2005) 663–678.

[247] L.H. Lee, C. Lee, Y.P. Tan, A multi-objective genetic algorithm for robust flight scheduling using simulation, European Journal of Operational Research 177 (3) (2007) 1948–1968.

[248] G. Quan, G.W. Greenwood, D. Liu, X.S. Hu, Searching for multiobjective preventive maintenance schedules: combining preferences with evolutionary algorithms, European Journal of Operational Research 177 (3) (2007) 1969–1984.

[249] A. Ghosh, B. Nath, Multi-objective rule mining using genetic algorithms, Information Sciences 163 (1–3) (2004) 123–133.

[250] M. Kaya, Multi-objective genetic algorithm based approaches for mining optimized fuzzy association rules, Soft Computing 10 (7) (2006) 578–586.
[251] M.J. Gacto, R. Alcalá, F. Herrera, Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems, Soft Computing 13 (5) (2009) 419–436.
[252] L. Sánchez, J. Otero, I. Couso, Obtaining linguistic fuzzy rule-based regression models from imprecise data with multiobjective genetic algorithms, Soft Computing 13 (5) (2009) 467–479.
[253] Y. Zhang, P. Rockett, A generic optimising feature extraction method using multiobjective genetic programming, Applied Soft Computing 11 (1) (2011) 1087–1097.
[254] C.-K. Ting, C.-N. Lee, H.-C. Chang, J.-S. Wu, Wireless heterogeneous transmitter placement using multiobjective variable-length genetic algorithm, IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 39 (4) (2009) 945–958.
[255] T. Siegfried, S. Bleuler, M.L.E. Zitzler, W. Kinzelbach, Multiobjective groundwater management using evolutionary algorithms, IEEE Transactions on Evolutionary Computation 13 (2) (2009) 229–242.
[256] I.H. Toroslu, Y. Arslanoglu, Genetic algorithm for the personnel assignment problem with multiple objectives, Information Sciences 177 (3) (2007) 787–803.
[257] I.-T. Yang, J.-S. Chou, Multiobjective optimization for manpower assignment in consulting engineering firms, Applied Soft Computing 11 (1) (2011) 1183–1190.
[258] K.C. Tan, C.Y. Cheong, C.K. Goh, Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation, European Journal of Operational Research 177 (2) (2007) 813–839.
[259] D. Maravall, J. de Lope Asiaín, Multi-objective dynamic optimization with genetic algorithms for automatic parking, Soft Computing 11 (3) (2007) 249–257.
[260] M. Panduro, C.A. Brizuela, D. Covarrubias, C. Lopez, A trade-off curve computation for linear antenna arrays using an evolutionary multi-objective approach, Soft Computing 10 (2) (2006) 125–131.
[261] S. Pal, S. Das, A. Basak, Design of time-modulated linear arrays with a multi-objective optimization approach, Progress in Electromagnetics Research B 23 (2010) 83–107.
[262] S. Pal, S. Das, A. Basak, P.N. Suganthan, Synthesis of difference patterns for monopulse antennas with optimal combination of array-size and number of subarrays—a multi-objective optimization approach, Progress in Electromagnetics Research B 21 (2010) 257–280.
[263] E. Masazade, R. Rajagopalan, P.K. Varshney, C.K. Mohan, G.K. Sendur, M. Keskinoz, A multiobjective optimization approach to obtain decision thresholds for distributed detection in wireless sensor networks, IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 40 (2) (2010) 444–457.
[264] S.-Y. Shin, I.-H. Lee, D. Kim, B.-T. Zhang, Multiobjective evolutionary optimization of DNA sequences for reliable DNA computing, IEEE Transactions on Evolutionary Computation 9 (2) (2005) 143–158.
[265] S.-Y. Shin, I.-H. Lee, Y.-M. Cho, K.-A. Yang, B.-T. Zhang, EvoOligo: oligonucleotide probe design with multiobjective evolutionary algorithms, IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 39 (6) (2009) 1606–1616.
[266] A. Benedetti, M. Farina, M. Gobbi, Evolutionary multiobjective industrial design: the case of a racing car tire-suspension system, IEEE Transactions on Evolutionary Computation 10 (3) (2006) 230–244.
[267] C. Erbas, S. Cerav-Erbas, A.D. Pimentel, Multiobjective optimization and evolutionary algorithms for the application mapping problem in multiprocessor system-on-chip design, IEEE Transactions on Evolutionary Computation 10 (3) (2006) 358–374.
[268] R. Saravanan, S. Ramabalan, N.G.R. Ebenezer, C. Dharmaraja, Evolutionary multi criteria design optimization of robot grippers, Applied Soft Computing 9 (1) (2009) 159–172.
[269] O. Castillo, L. Trujillo, P. Melin, Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots, Soft Computing 11 (3) (2007) 269–279.
[270] V.V.R. Silva, P.J. Fleming, J. Sugimoto, R. Yokoyama, Multiobjective optimization using variable complexity modelling for control system design, Applied Soft Computing 8 (1) (2008) 392–401.
[271] P. Wozniak, Preferences in multi-objective evolutionary optimisation of electric motor speed control with hardware in the loop, Applied Soft Computing 11 (1) (2011) 49–55.
[272] S.Z. Zhao, M.W. Iruthayarajan, S. Baskar, P.N. Suganthan, Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization, Information Sciences, 5th revision submitted.
[273] B. Lazzerini, F. Marcelloni, M. Vecchio, A multi-objective evolutionary approach to image quality/compression trade-off in JPEG baseline algorithm, Applied Soft Computing 10 (2) (2010) 548–561.
[274] S. Wiegand, C. Igel, U. Handmann, Evolutionary multi-objective optimisation of neural networks for face detection, International Journal of Computational Intelligence and Applications 4 (3) (2004) 237–254.
[275] D. Balasubramanian, M.C. Krishna, R. Murugesan, Multi-objective ga-optimized interpolation kernels for reconstruction of high resolution EMR images from low-sampled *k*-space data, International Journal of Computational Intelligence and Applications 8 (2) (2009) 127–140.
[276] A. Mukhopadhyay, U. Maulik, A multiobjective approach to MR brain image segmentation, Applied Soft Computing 11 (1) (2011) 872–880.
[277] Y. Zhang, P. Rockett, A generic multi-dimensional feature extraction method using multiobjective genetic programming, Evolutionary Computation 17 (1) (2009) 89–115.
[278] S. Dehuri, S. Patnaik, A. Ghosh, R. Mall, Application of elitist multi-objective genetic algorithm for classification rule generation, Applied Soft Computing 8 (1) (2008) 477–487.
[279] G.N. Demir, S. Uyar, S.G. Ögüdücü, Multiobjective evolutionary clustering of web user sessions: a case study in web page recommendation, Soft Computing 14 (6) (2010) 579–597.
[280] P. Ducange, B. Lazzerini, F. Marcelloni, Multi-objective genetic fuzzy classifiers for imbalanced and cost-sensitive datasets, Soft Computing 14 (7) (2010) 713–728.
[281] J. Handl, J.D. Knowles, An evolutionary approach to multiobjective clustering, IEEE Transactions on Evolutionary Computation 11 (1) (2007) 56–76.
[282] R. Romero-Záliz, C. Rubio-Escudero, J.P. Cobb, F. Herrera, O. Cordón, I. Zwir, A multiobjective evolutionary conceptual clustering methodology for gene annotation within structural databases: a case of study on the gene ontology database, IEEE Transactions on Evolutionary Computation 12 (6) (2008) 679–701.
[283] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, Multiobjective genetic algorithm-based fuzzy clustering of categorical attributes, IEEE Transactions on Evolutionary Computation 13 (5) (2009) 991–1005.
[284] S.N. Qasem, S.M.H. Shamsuddin, Radial basis function network based on time variant multi-objective particle swarm optimization for medical diseases diagnosis, Applied Soft Computing 11 (1) (2011) 1427–1438.
[285] A.A. Aguilar-Lasserre, L. Pibouleau, C. Azzaro-Pantel, S. Domenech, Enhanced genetic algorithm-based fuzzy multiobjective strategy to multiproduct batch plant design, Applied Soft Computing 9 (4) (2009) 1321–1330.
[286] J. González, I. Rojas, H. Pomares, F. Rojas, J.M. Palomares, Multi-objective evolution of fuzzy systems, Soft Computing 10 (9) (2006) 735–748.
[287] M. Cococcioni, P. Ducange, B. Lazzerini, F. Marcelloni, A Pareto-based multi-objective evolutionary approach to the identification of Mamdani fuzzy systems, Soft Computing 11 (11) (2007) 1013–1031.
[288] M. Cococcioni, B. Lazzerini, F. Marcelloni, On reducing computational overhead in multi-objective genetic Takagi–Sugeno fuzzy systems, Applied Soft Computing 11 (1) (2011) 675–688.
[289] S.N. Omkar, J. Senthilnath, R. Khandelwal, G.N. Naik, S. Gopalakrishnan, Artificial bee colony (ABC) for multi-objective design optimization of composite structures, Applied Soft Computing 11 (1) (2011) 489–499.
[290] X. Zeng, Y. Zhu, L. Koehl, M. Camargo, C. Fonteix, F. Delmotte, A fuzzy multi-criteria evaluation method for designing fashion oriented industrial products, Soft Computing 14 (12) (2010) 1277–1285.
[291] L. Govindarajan, T. Karunanithi, Multiobjective optimization of process plant using genetic algorithm, International Journal of Computational Intelligence and Applications 5 (4) (2005) 425–438.
[292] L. Govindarajan, T. Karunanithi, Multiobjective optimization of process plant using genetic algorithm, International Journal of Computational Intelligence and Applications 6 (3) (2006) 315–327.
[293] S. Uhlig, A multiple-objectives evolutionary perspective to interdomain traffic engineering, International Journal of Computational Intelligence and Applications 5 (2) (2005) 215–230.
[294] J.G. Iniestra, J.G. Gutiérrez, Multicriteria decisions on interdependent infrastructure transportation projects using an evolutionary-based framework, Applied Soft Computing 9 (2) (2009) 512–526.
[295] C. Horoba, Exploring the runtime of an evolutionary algorithm for the multi-objective shortest path problem, Evolutionary Computation 18 (3) (2010) 357–381.
[296] O. Steuernagel, D. Polani, Multiobjective optimization applied to the eradication of persistent pathogens, IEEE Transactions on Evolutionary Computation 14 (5) (2010) 759–765.
[297] I. Aydin, M. Karaköse, E. Akin, A multi-objective artificial immune algorithm for parameter optimization in support vector machine, Applied Soft Computing 11 (1) (2011) 120–129.
[298] G. Ascia, V. Catania, A.G.D. Nuovo, M. Palesi, D. Patti, Performance evaluation of efficient multi-objective evolutionary algorithms for design space exploration of embedded computer systems, Applied Soft Computing 11 (1) (2011) 382–398.
[299] R.M. Everson, J.E. Fieldsend, Multiobjective optimization of safety related systems: an application to short-term conflict alert, IEEE Transactions on Evolutionary Computation 10 (2) (2006) 187–198.
[300] L. Poladian, L.S. Jermiin, Multi-objective evolutionary algorithms and phylogenetic inference with multiple data sets, Soft Computing 10 (4) (2006) 359–368.
[301] P. Kumar, P. Bauer, Progressive design methodology for complex engineering systems based on multiobjective genetic algorithms and linguistic decision making, Soft Computing 13 (7) (2009) 649–679.
[302] G. Chen, S. Chen, W. Guo, H. Chen, The multi-criteria minimum spanning tree problem based genetic algorithm, Information Sciences 177 (22) (2007) 5050–5063.
[303] M. Hakimi-Asiabar, S.H. Ghodsypour, R. Kerachian, Deriving operating policies for multi-objective reservoir systems: application of self-learning genetic algorithm, Applied Soft Computing 10 (4) (2010) 1151–1163.
[304] Z. Song, A. Kusiak, Multiobjective optimization of temporal processes, IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 40 (3) (2010) 845–856.
[305] K. Deb, A. Sinha, An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary-local-search algorithm, Evolutionary Computation 18 (3) (2010) 403–449.
[306] J. Bader, E. Zitzler, HypE: an algorithm for fast hypervolume-based many-objective optimization, Evolutionary Computation 19 (1) (2011) 45–76.

[307] A. Asllani, A. Lari, Using genetic algorithm for dynamic and multiple criteria web-site optimizations, European Journal of Operational Research 176 (3) (2007) 1767–1777.
[308] A. Zafra, E.L.G. Galindo, S. Ventura, Multiple instance learning with multiple objective genetic programming for web mining, Applied Soft Computing 11 (1) (2011) 93–102.
[309] A.C. Briza, P.C. Naval Jr., Stock trading system based on the multi-objective particle swarm optimization of technical indicators on end-of-day market data, Applied Soft Computing 11 (1) (2011) 1191–1201.
[310] T. Kremmel, J. Kubalík, S. Biffl, Software project portfolio optimization with advanced multiobjective evolutionary algorithms, Applied Soft Computing 11 (1) (2011) 1416–1426.