University of Glasgow

Chen, N., Chen, W.-N., Gong, Y.-J., Zhan, Z.-H., Zhang, J., Li, Y., and Tan, Y.-S. (2014) An evolutionary algorithm with double-level archives for multiobjective optimization. IEEE Transactions on Cybernetics (99). ISSN 2168-2267

http://eprints.gla.ac.uk/100869

Deposited on: 14 January 2015

An Evolutionary Algorithm with Double-Level Archives for Multiobjective Optimization

Ni Chen, *Student Member, IEEE*, Wei-Neng Chen, *Member, IEEE*, Yue-Jiao Gong, *Student Member, IEEE*, Zhi-Hui Zhan, *Member, IEEE*, Jun Zhang, *Senior Member, IEEE,* Yun Li, *Member, IEEE,* Yu-Song Tan, *Member, IEEE*

Abstract—Existing multiobjective evolutionary algorithms (MOEAs) tackle a multiobjective problem either as a whole or as several decomposed single-objective sub-problems. Though the problem decomposition approach generally converges faster through optimizing all the sub-problems simultaneously, there are two issues not fully addressed, i.e., distribution of solutions often depends on *a priori* problem decomposition, and the lack of population diversity among sub-problems. In this paper, a MOEA with double-level archives is developed. The algorithm takes advantages of both the multiobjective-problem-level and the sub-problem-level approaches by introducing two types of archives, i.e., the global archive and the sub-archive. In each generation, self-reproduction with the global archive and cross-reproduction between the global archive and sub-archives both breed new individuals. The global archive and sub-archives communicate through cross-reproduction, and are updated using the reproduced individuals. Such a framework thus retains fast convergence, and at the same time handles solution distribution along Pareto front (PF) with scalability. To test the performance of the proposed algorithm, experiments are conducted on both the widely used benchmarks and a set of truly disconnected problems. The results verify that, compared with state-of-the-art MOEAs, the proposed algorithm offers competitive advantages in distance to the PF, solution coverage, and search speed.

Index Terms—Evolutionary algorithm (EA), global optimization, multiobjective optimization.

I. INTRODUCTION

O PTIMIZATION problems with multiple objectives often have a set of optimal solutions. These solutions define the inherent Pareto set (PS) [1]. In *a-posteriori* preference articulation methods [1, part II, ch. 3], there are primarily two goals in multiobjective optimization. The first goal is to converge as close to the Pareto front (PF) as possible in the search space with a high speed, and the second one is to distribute solutions as evenly as possible on the entire PF with a high density.

As an evolutionary algorithm (EA) [2], [3] works with a population of individuals, it has the potential to achieve both goals of multiobjective optimization. A number of multiobjective EAs (MOEAs) have been reported in the literature, which can deliver a set of solutions in a single run [4]–[9].

Most MOEAs work at the multiobjective problem level, which means that individuals, archives and operators are associated with multiple objectives, and special operators are designed to fulfill the two goals of multiobjective optimization. Represented by the Non-dominated sorting genetic algorithm II (NSGA-II) [9], which selects individuals according to a nondominated rank and crowding distance, one group of algorithms in this category uses a multiobjective selection strategy to achieve solutions close to the PS with a relative even distribution of objective vectors. Examples are improved versions of differential evolution (DE) [10]–[12], the "vector evaluation genetic algorithm" (VEGA) based on alternating objectives [13], the domination-based "Pareto archived evolutionary strategy" (PAES) [7], "strength-Pareto EA" (SPEA) [8] and the "strength Pareto EA II" (SPEA-II) [14]. Another group in this category designs population distribution strategies to achieve a good distribution of solutions. Strategies include, using multiple populations [15] and dynamic multiple populations [16], [17], clustering the candidate individuals [18], [19], and devising new strategies to estimate the density of objective space. In most algorithms that work at the problem level, the distribution of solutions on the PF is controlled implicitly at the multiobjective problem level with the optimization operators, and adapts to different shapes of PF.

More recently, another class of MOEAs based on problem decomposition has gained much attention. These algorithms work at a single-objective sub-problem level, where all evolutionary operators handle only single-objective sub-problems. The distribution of solutions on PF explicitly depends on the

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

2   IEEE TRANSACTIONS ON CYBERNETICS

decomposition method, i.e., the definition of sub-problems. Early research efforts on problem decomposition include the research work in [20]–[23]. Later, Zhang and Li [24] proposed the MOEA/D which decomposes the problem into single-objective scalar optimization problems and optimizes all the scalar problems simultaneously. More recently, based on the fact that the framework of MOEA/D is compatible with all existing single-objective reproduction operators, an enhanced version of MOEA/D [25] which adopts the reproduction operator of DE [26] has been proposed. Since then, MOEAs based on decomposition have become the state-of-the-art, featured with a higher convergence speed brought about by simultaneous optimization of multiple single-objective sub-problems, a high compatibility with single-objective evolutionary operators, and better coverage of the PF on a set of widely used benchmarks like ZDT [27] problems.

Despite the encouraging performance of MOEAs based on decomposition, there are two unaddressed issues with most algorithms in this class, i.e., the explicit dependence of the solution distribution on the *a priori* definition of sub-problems and the lack of population diversity for each sub-problem. When the shape of PF is not known *a priori*, a certain decomposition method cannot guarantee good scattering of solutions on the PF [28]. Since the algorithm only works at sub-problem level, it is difficult to fix improper definition of sub-problems during runtime. According to the research in [29], solving the problem with adaptation has corresponding costs. Besides, most of these algorithms adopt an elitist strategy in their selection for each sub-problem, which means that one new solution replaces an old one immediately if it has a better aggregation function value. Such replacement can result in a lack of population diversity at the sub-problem level. Empirical experiments have indicated that algorithms in this class handle cases with truly disconnected PF and extreme shapes of slope on PF with difficulty [30], which could lead to unfavorable performance in solving complicated real problems.

In this paper, an MOEA with double-level archives (MOEA-DLA) is proposed to address the above issues. The proposed algorithm maintains promising individuals, preserves population diversity, and controls the distribution of solutions along PF at the multiobjective problem-level and the single-objective sub-problem-level simultaneously. The sub-problem-level sub-archives are adopted to obtain faster convergence introduced by problem decomposition, and to maintain population diversity for each sub-problem. At the same time, to preserve the population diversity at the problem-level and to handle complicated shapes of PF, the problem-level global archive is adopted. The global archive and the sub-archive communicate indirectly with the cross-reproduction.

The proposed method has the following features.

1) The MOEA-DLA optimizes all of the single-objective sub-problems and the multiobjective problem simultaneously to gain a higher search speed.

2) The MOEA-DLA provides a mechanism of diversity preservation and distribution control at both the sub-problem level and the problem level, and a mechanism for the communication of the two levels. This way,

the explicit dependency of solution distributions on sub-problem definition in existing decomposition-based MOEAs is relieved, which means that the proposed algorithm would handle different shapes of PF with increased scalability.

3) The MOEA-DLA provides a framework for multiobjective optimization with no other parameters than the number and capacity of the archives. Existing selection strategies and density estimation strategies for multiobjective optimization and reproduction operators for single-objective optimization can fit in the framework conveniently. Besides, existing problem decomposition method, e.g., decomposition method adopted in MOEA/D and its enhanced version MOEA/D-M2M [31], and existing local search strategies can be adapted to the proposed framework with minor modifications.

To validate the effectiveness of the proposed MOEA-DLA, experiments will be conducted on 12 multiobjective test problems. The widely used ZDT [27] problems, the WFG [32] problems and the truly disconnected TDY [30] problems will be employed to test the algorithm performance on conventional as well as highly disconnected shapes of PS and PF. Specifically, the TDY problems are a set of problems with true PS and PF in the form of multiple disconnected segments, and are thus challenging for MOEAs.

The rest of this paper is organized as follows. Section II presents the background, including a brief review on MOEAs and the definition of sub-problems, which works as a preliminary. Section III describes the MOEA-DLA in detail, followed by experiments on test problems in Section IV. Section V concludes the paper.

II. BACKGROUND

A. Brief Review of MOEAs

In the early 1990s, a number of MOEAs were developed [2], [3]. Among them, multi-objective genetic algorithm (MOGA) [33], Non-dominated sorting genetic algorithm (NSGA) [34], and niched Pareto genetic algorithm (NPGA) [4] have attracted most attention. To achieve the goals of multiobjective optimization, these early MOEAs focus on two issues: 1) designing methods to evaluate and preserve individuals close to the true PF, e.g., using nondominated sorting and 2) devising mechanisms of maintaining diversity of achieved solutions on the PF. The two issues are so significant in MOEAs that they have attracted constant research attention.

Among various approaches to enhancing MOEA performance on the above two aspects, archives with elitism, which preserve historically best solutions, has drawn much research interest. Solutions preserved in elitism are determined through association with both domination relationship between solutions and the solution distribution. Representatives of these MOEAs with elitism include the SPEA [8], PEAS [7], and NSGA-II [9]. The SPEA is an elitist multicriterion EA with the concept of nondomination. In SPEA, an external population, i.e., archive, is suggested to be maintained at every generation to store all nondominated solutions found so far.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

CHEN *et al.*: EA WITH DOUBLE-LEVEL ARCHIVES FOR MULTIOBJECTIVE OPTIMIZATION 3

The archive participates in the whole evolutionary process and the updating strategy for the archive significantly influences the performance of algorithm.

Considering the issues that: 1) a selection process preferring nondominated solutions would be adopted to preserve individuals closer to the PS and 2) a technique is required to achieve diversified distribution, the PAES and the NSGA-II are proposed. Addressing issue 1), the PAES uses only one parent and one offspring in each generation and the NSGA-II proposed the nondominated sorting. Addressing issue 2), in PAES positions of historically best solutions in an archive are referred to, and in NSGA-II the crowding distance is introduced for the selection of individuals residing in a less crowded region.

There are other research efforts concerning solution distribution strategies, which work on distributing the population with diversity with certain mechanism. These efforts include clustering the candidate individuals [18], [19] to achieve highly spreading population, using multiple populations [15], quantizing the solution space for selection [46], using dynamic multiple populations [16], [17], devising new strategies to estimate the density of objective space, territory definition around each individual [35], and estimating the density of solution space [30].

More recently, one class of MOEAs have been developed based on problem decomposition and have gained much attention [20]–[23], [36], [37]. These algorithms achieve the two MOEA goals in a different way from optimizing the multiobjectives as a whole collectively. By decomposing the multiobjective problem into different single-objective problems, the task of finding solutions close to the true PF for the multiobjective problem is first replaced by that of optimizing a set of single-objective sub-problems. In this case, the distribution of solutions on PF merely depends on the methods of decomposition at the beginning and of recombination at the end. Among the MOEAs based on decomposition, the MOEA/D proposed by Zhang and Li [24] is a representative. This algorithm, after decomposing an MOEA problem into multiple single-objective scalar optimization problems, optimizes all the scalar objectives simultaneously using only single-objective evolutionary operators for simplicity and speed. Since the framework of MOEA/D is compatible with existing single-objective reproduction operators, an enhanced version of MOEA/D [25] which adopts the reproduction operator of DE [26] has been proposed. Algorithms in this class are competitive for a high convergence speed, high compatibility with single-objective evolutionary operators, and good coverage to the PF, as validated by a set of benchmark tests. Based on problem decomposition, another decomposition method has been proposed by Liu *et al.* [31]. The algorithm decomposes the multiobjective problem into a number of simple multiobjective problems and assign each sub-problem one sub-population to conquer the lack of sub-population diversity resulted from the elitism in MOEA/D.

To enhance the local exploitation ability of MOEAs, MOEAs are hybridized with local search strategies. In [38], a synchronous particle local search (SPLS) is adopted in multiobjective particle swarm optimization (MOPSO).

Ke *et al.* [39] proposed a memetic algorithm based on decomposition (MOMAD), which hybridize the Pareto local search with problem decomposition.

For many-objective optimization, research efforts have been paid on strategies for working with large number of objectives efficiently. Wang *et al.* [40] proposed a preference-inspired coevolutionary algorithm (PICEA) for many-objective optimization, which coevolved a population of solutions together with a set of decision-maker preferences. Deb and Jain [41] designed a many-objective particle swarm optimization algorithm based on reference point. For set quality measurement in many-objective MOEAs, Bader and Zitzler [42] designed a hypervolume estimation algorithm HypE. In [43], a shift-based density estimation (SDE) strategy is proposed in order to make Pareto-based MOEAs suitable for many-objective EAs.

B. Definition of Sub-Problems

The MOEA-DLA adopts both the problem-level archive and sub-problem-level sub-archives, with each sub-archive serves one corresponding sub-problem. As the proposed method is closely related to the definition of sub-problems, this section introduces the problem decomposition approach and the definition of sub-problems.

1) Problem Decomposition: Several decomposition methods have been proposed to decompose a multiobjective problem into a series of single-objective sub-problems. In this paper, two most commonly used decomposition approaches are briefly introduced as follows. For more information on decomposition methods, one can refer to [44] and [45].

a) Weighted sum approach: The weighted sum approach is an intuitive approach to problem reduction. The method considers a convex combination of all the objectives. To reduce a multiobjective problem to a single-objective one, define a weight vector $\mathbf{w} = (w_1, \ldots, w_m)^T$, where $w_i \geq 0$ for all $i = 1, 2, \ldots, m$ and $\sum_{i=1}^{m} w_i = 1$. One weight vector $\mathbf{w}$ yields one sub-problem of the multiobjective problem

$$\text{minimize} \quad g_w(x) = \sum_{i=1}^{m} w_i f_i(x) \tag{1}$$

where $f_i(x)$ is the objective function of the *i*th objective and whereby altering $\mathbf{w}$ can control the positions of optimal solutions on the PF explicitly.

b) Chebyshev approach: In the Chebyshev (also known as Tchebycheff) approach [1], the weight vector $\mathbf{w}$ is similar to the weighted sum approach, but the sub-problems are defined as

$$\text{minimize} \quad g_w(x) = \max_{1 \leq i \leq M} \{w_i \cdot |f_i(x) - z_i|\} \tag{2}$$

where $z_i = (z_1, z_2, \ldots, z_M)$ is the reference point defined as $z = \inf(f_j(x))$ for each $j = 1, 2, \ldots, M$.

The Chebyshev approach to optimal solutions for a sub-problem is controlled by the weight vector, which is similar to the weighted sum approach. Here the reference point is associated with the offset of PF. Problems with shifted PF in the objective space can be handled by defining suitable reference
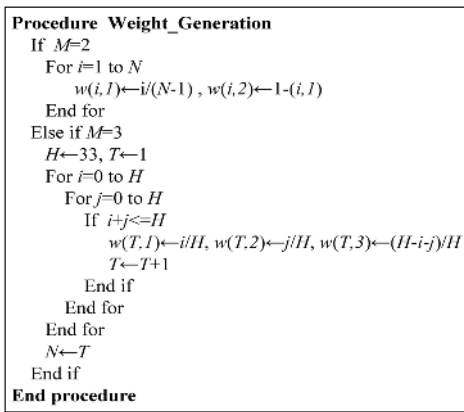
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

4

IEEE TRANSACTIONS ON CYBERNETICS

```
Procedure Weight_Generation
  If M=2
    For i=1 to N
      w(i,1)←i/(N-1) , w(i,2)←1-(i,1)
    End for
  Else if M=3
    H←33, T←1
    For i=0 to H
      For j=0 to H
        If i+j<=H
          w(T,1)←i/H, w(T,2)←j/H, w(T,3)←(H-i-j)/H
          T←T+1
        End if
      End for
    End for
    N←T
  End if
End procedure
```

Fig. 1.   Procedure for the generation of weight vectors.

Fig. 2.   Basic flow of the MOEA-DLA.

point. In practice, the value of $\inf(f_j(x))$ is usually not known *a priori*; thus the algorithm uses z_j^*, i.e., dynamically obtained best-so-far value of $f_j(x)$, as the jth dimension of the reference point.

2) Generation of Weight Vectors: Since the scalarizing functions are continuous functions of the weighting vectors, an infinite number of sub-problems can be defined. Only are a few numbers of these sub-problems used in the actual process of optimization. In the proposed algorithm, the number of sub-problems optimized simultaneously equals the population size N.

The procedure for the generation of weight vectors for bi-objective and tri-objective problems, which is illustrated in Fig. 1, is derived from the same procedure in [24] (also used in [25] and [47]). Here N is the population size, M is the number of objectives, and H is a parameter defining N in tri-objective cases. In this paper, two and three objectives are considered for multiobjective optimization. Cases with more objectives can be handled with an extension of the presented method.

3) Normalization of Fitness for Sub-Problems: In this paper, the Chebyshev approach is used with objective normalization. Much research effort has been made on the objective normalization in evolutionary computation [48]–[50]. A basic normalization method is to redefine the objective f_i as

$$\bar{f}_i(x) = \frac{f_i(x) - z_i'}{z_i - z_i'} \tag{3}$$

where $z_i' = \inf(f_i(x))$ and $z_i = \sup(f_i(x))$. Since the value of $\sup(f_j(x))$ is usually not known *a priori*, the algorithm uses z_i^* and $z_i'^*$, i.e., dynamically achieved maximum and minimum values of $f_i(x)$ during optimization. In such a way, the value of every objective is normalized within the range [0, 1].

III. MOEA-DLA

A. Evolutionary Process

The proposed MOEA-DLA adopts two levels of archives to curate and maintain promising solutions. One level is the problem-level, where one global archive for the multiobjective problem is updated with elitism. Through the elitism strategy, promising individuals are maintained globally at the problem level. The other level is the sub-problem level, where one
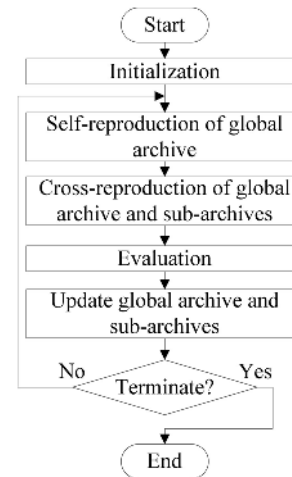
sub-archive is assigned for each single-objective sub-problem. During the evolutionary process, all sub-problems are optimized simultaneously at the sub-problem level to accelerate the convergence.

The two levels of archives work with their own updating strategies and at the same time communicate with each other. The global archive reproduces individuals independently through self-reproduction. The self-reproduction procedure allows the promising individuals in the global archive to breed offspring, which contributes to evolving the population at the problem level. On the other hand, the two levels of archives communicate through cross-reproduction, where the reproduced individuals are used to update all the archives. Through the cross-reproduction, the individuals in the sub-archives are able to breed offspring with the promising individuals in the global archive. Both the global archive and the sub-archives learn from each other in the process of cross-reproduction.

This way, the problem-level global archive benefits from the fast convergence at the sub-problem level, and diversifies the population through learning from different sub-problems. Besides, the solution distribution in the global archive would not explicitly depend on the definition of sub-problems for the global archive works with self-reproduction and its own updating strategy at the problem level.

A flowchart of the MOEA-DLA is shown in Fig. 2. In each generation, MOEA-DLA contains the basic steps of an ordinary MOEA, i.e., reproduction of individuals, evaluation of individuals, and archive update. Specifically, the reproduction procedure consists of self-reproduction of a global archive and cross-reproduction of the global archive and sub-archives. Detailed descriptions of the flowchart are as follows.

Step 1: *Initialization:* N individuals are generated stochastically within the upper and lower bounds to form the initial population $\mathbf{P}_0$, where N is the size of population. A number of N sub-archives are initialized with weight vectors generated according to the previous section. Each sub-archive is randomly assigned one individual from $\mathbf{P}_0$.

Step 2: *Self-Reproduction of the Global Archive:* The algorithm reproduces N offspring from the current
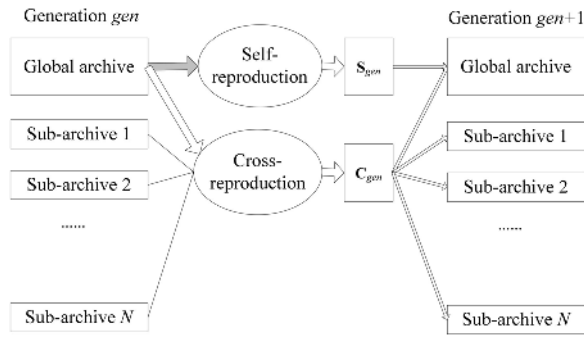
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

CHEN *et al.*: EA WITH DOUBLE-LEVEL ARCHIVES FOR MULTIOBJECTIVE OPTIMIZATION 5

Fig. 3. Flow of individuals in the MOEA-DLA.

global archive to form $\mathbf{S}_{\text{gen}}$. Detailed description of self-reproduction is presented in part *B*.

Step 3: *Cross-Reproduction of the Global Archive and Sub-Archives:* The algorithm reproduces N offspring to form $\mathbf{C}_{\text{gen}}$. Both the global archive and sub-archives contribute to the N offspring reproduced in this step. Detailed description of cross-reproduction is presented in part *B*.

Step 4: *Evaluation:* Evaluate all of the individuals in $\mathbf{S}_{\text{gen}}$ and $\mathbf{C}_{\text{gen}}$.

Step 5: *Update the Sub-Archives and the Global Archive:* Use $\mathbf{S}_{\text{gen}}$ and $\mathbf{C}_{\text{gen}}$ generated in previous steps to update the global archive and the sub-archive.

For sub-archives, all of the individuals in $\mathbf{C}_{\text{gen}}$ are used to update all sub-archives. Each sub-archive is updated according to the fitness values of its corresponding single-objective sub-problem. To update the global archive, all of the individuals in $\mathbf{S}_{\text{gen}}$ and $\mathbf{C}_{\text{gen}}$ are added to the global archive. If the number of individuals in the global archive exceeds the maximum number $\mathbf{G}_{\text{max}}$, one selection procedure with elitism is performed. The detailed strategy is described in the following part.

Step 6: *Termination check:* If the number of function evaluation (FEs) exceeds the predefined maximum number, the algorithm terminates. Otherwise, increase *gen* by 1, go back to Step 2 and start a new generation.

Fig. 3 presents the flow of individuals in two successive generations. As shown in the figure, the self-reproduction reproduces N individuals, i.e., $\mathbf{S}_{\text{gen}}$, and the cross-reproduction reproduces another N individuals, i.e., $\mathbf{C}_{\text{gen}}$. All of the $2N$ individuals reproduced are used to update the global archive to maintain a "best-so-far" nondominated set, whereas only the individuals reproduced in cross-reproduction are used to update the sub-archives.

B. Self-Reproduction and Cross-Reproduction

1) Self-Reproduction: In this procedure, the global archive reproduces within itself, which is independent from the sub-archives. Promising individuals from the global archive undergo crossover and mutation to produce new offspring. The strategy for self-reproduction is as follows.

1) Two or three individuals are selected according to crowding distance which is proposed in NSGA-II [9]

and utilized in many MOEAs. Roulette wheel selection is adopted to select individuals with large crowding distances with a relative high probability. The number of individuals selected is determined by the crossover operator in the following step.

2) Two offspring are reproduced through crossover and mutation. To reproduce new individuals, one crossover operator, e.g., blend crossover (BLX) [51], simulated binary crossover (SBX) [52], simplex crossover (SPX) [53], or DE [26], is adopted, followed by the mutation operator performed with a probability of p_m. In this paper, the DE crossover and polynomial mutation are used.

3) Terminate if N offspring are reproduced, otherwise go back to 1).

2) Cross-Reproduction: Through cross-reproduction, the global archive communicates with the sub-archives to learn from the fast-converging optimization of sub-problems. At the same time, all the sub-archives learn from the global archive simultaneously to converge globally on each sub-problem. Individuals from both global archive and sub-archives contribute to the N offspring reproduced here. The strategy for cross-reproduction is as follows.

1) For a randomly picked sub-archive, select one individual from the nondominated set of global archive according to crowding distance, and one individual from the sub-archive based on fitness. Roulette wheel selection is used in the proposed algorithm.

2) Reproduce two offspring through crossover and mutation. For simplicity, the crossover and mutation operator are the same as those in Step 2.

3) Terminate if N offspring are reproduced, otherwise go back to 1).

C. Update of Sub-Archives and Global Archive

1) Sub-Archives: The sub-archives maintain limited number of solutions for single-objective sub-problems, which provides a mechanism for diversity preservation at the sub-problem level.

Since the sub-problems each has only one fitness value, all the solutions are comparable according to the definition of sub-problem. The definition of fitness for the ith sub-archive is

$$g_i(x) = \sum_{j=1}^{m} w_i^j f_j(x) \qquad (4)$$

where f_j denotes the objective function for the jth objective, and w^j_i denotes the jth dimension for the weight vector of the ith sub-problem.

The individuals in sub-archives are maintained considering their corresponding fitness and the crowding distance [9]. The crowding distance, which is used in the NSGA-II [9], provides an estimation of solution density in objective space. To preserve the diversity, solutions with larger crowding distance are preferred.

In the proposed work, the maximum number of individuals in one sub-archive is $\mathbf{S}_{\text{max}}$. All the individuals reproduced in the cross-reproduction are used to update every sub-archive.

The update strategy for individual *ind* in $\mathbf{C}_{\text{gen}}$ for the *i*th sub-archive $\mathbf{Sub}i$ is described as follows.

Step 1: Add the individual *ind* to the sub-archive.

Step 2: If the number of individuals in sub-archive does not exceed the maximum number $\mathbf{S}_{\text{max}}$, go to Step 4) directly. Otherwise, go to Step 3).

Step 3: If $g_i(ind)$ is better than the best fitness value in $\mathbf{Sub}i$, remove a randomly picked individual except the one with best crowding distance from $\mathbf{Sub}i$ and add *ind* to $\mathbf{Sub}i$. If $g_i(ind)$ and the crowding distance for *ind* are both better than the fitness value of any individual in the $\mathbf{Sub}i$, replace a randomly picked individual that has both worse fitness and worse crowding distance with *ind*. Otherwise remove *ind*.

Step 4: Terminate the procedure.

As shown above, when the number of individuals exceeds the maximum number, the sub-archive eliminates one individual, with the best one always kept. Similar to the global archive update strategy, the update strategy for sub-archive adopts elitism inherently.

In the evolutionary process, all of the sub-archives are updated simultaneously and all the sub-problems are thus optimized simultaneously. The strategy accelerates the convergence, which has been largely verified by the MOEA/D. However, the distribution of individuals in all sub-archives depends on the decomposition method, since the optimum of each sub-problem is defined by the definition of sub-problems.

2) Global Archive: Different from the sub-archives, the global archive maintains a limited number of solutions to cover the PF of the problem. To update the global archive, selection strategy for multiobjective optimization is used. The update of global archive is independent from the definition of sub-problems.

In the proposed work, both the domination of solutions and crowding distances are considered for the selection of individuals. The selection of individuals with larger crowding distance gives the individuals with fewer neighbors in the objective space more chances of surviving and breeding.

The strategy for global archive update is as follows.

Step 1: Add all individuals in $\mathbf{S}_{\text{gen}}$ and $\mathbf{C}_{\text{gen}}$ to global archive.

Step 2: Remove all the individuals that are dominated by other individuals from the global archive. Only nondominated individuals are preserved in the global archive. If the number of individuals in global archive does not exceed the maximum number $\mathbf{G}_{\text{max}}$, go to Step 5) directly. Otherwise, go to Step 3).

Step 3: Sort the individuals in global archive with crowding distance.

Step 4: Preserve the first $\mathbf{G}_{\text{max}}$ individuals with largest crowding distance and remove the other individuals.

Step 5: Terminate the procedure.

According to the above procedure, the global archive is limited to $\mathbf{G}_{\text{max}}$ individuals. All the individuals dominated by other individuals are eliminated from the archive.

When the number of nondominated individuals exceeds the limit $\mathbf{G}_{\text{max}}$, individuals with largest crowding distance are preserved. This update strategy naturally adopts elitism.

D. Time Complexity of the Evolutionary Process of MOEA-DLA

The time consumption of the operators in MOEA-DLA is composed of three parts. The first part is finding the nondominated set and the crowding distances for individuals in the global archive. This is the basis for individual selection and the updating of global archive. For finding a nondominated set, the time complexity is $O(M(N + G_{\text{max}})^2)$ to compare $(N + G_{\text{max}})$ individuals, each costing M comparisons. For computing the crowding distance, the time complexity is $O(MN\log(N))$ to sort the individuals for M times. Thus, the overall complexity for this part is $O(MN^2)$. The second part is the time consumption of crossover and mutation. In our proposed method, the time complexity of the DE reproduction is $O(DN)$. The third part of time consumption is the cost of updating all of the sub-archives. Totally, N individuals are used to update all of the S sub-archives. For each sub-archive, a maximum of S_{max} comparisons are required to decide whether to accept the individual. Overall, the time complexity of the evolutionary operators in MOEA-DLA is $(M(N + G_{\text{max}})^2 + DN + S_{\text{max}}SN)$.

Here the S, G_{max}, and S_{max} are user-defined parameters. Usually, the S_{max} does not exceed 5, and the value of G_{max} and S, i.e., capacity of global archive and the number of sub-problems, are usually linear to the population size N. Thus, the time complexity of the operators in MOEA-DLA can be viewed as $O(MN^2 + DN)$.

IV. EXPERIMENTS AND DISCUSSION

A. Benchmark Test Problems

Experiments are conducted on 12 multiobjective problems to study the performance of the proposed MOEA-DLA. The test problems can be categorized into two groups.

The first group includes the ZDT [27], WFG [32], and DTLZ [54] problems, which are widely used to test the performance of MOEAs. The selected test problems are ZDT3, ZDT4, ZDT6, WFG1, WFG2, and WFG3. Besides, one of the DTLZ problems is selected to experiment the algorithms on problems with three-objectives.

The other group consists of the TDY problems introduced in [30], which are truly disconnected multiobjective problems (MOPs). The PF of these test problems are segments separated far away, and the segments involve some extreme shapes, e.g., near-horizontal line. Similar to the PF, the shapes of PS for TDY problems in the solution space are also truly disconnected. Experimental results in [30] have shown that these problems are difficult for most of the MOEAs which suppose that the neighborhood of one good solution always contain other good solutions. In this paper, the TDY problems are included in the test cases to study the algorithm performance and behavior of MOEA-DLA on truly disconnected PF and PS.
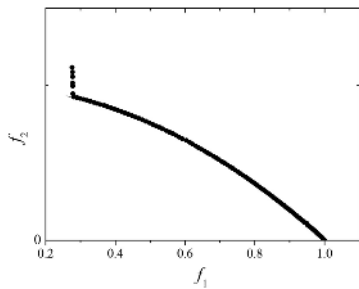
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

CHEN *et al.*: EA WITH DOUBLE-LEVEL ARCHIVES FOR MULTIOBJECTIVE OPTIMIZATION 7

Fig. 4. Example of nondominated set away from the PF with a low IGD value.

B. Performance Measures

To measure the algorithm performance of multiobjective optimization algorithms, performance measures have been suggested in [1] and [8].

In this paper, the inverted generational distance (IGD) [55] metric and the hyper volume (HV) metric [56] are considered. These two metrics are explained briefly as follows.

1) IGD Metric: Suppose that P_T is a series of uniformly distributed points along the true PF and P_A is the set of points achieve by an algorithm. Then the IGD is defined as

$$\frac{1}{|P_T|} \sum_{a \in P_T} \text{dist}(a, P_A) \tag{5}$$

where dist(a, P_A) is the Euclidean distance between the point a and P_A. A low IGD value requires the set P_A to be close to P_T, and to cover all parts of P_T.

The IGD value measures the performance of an algorithm considering both the convergence to PF and the coverage of PF. However, the IGD value can be low when an algorithm obtains a number of undesirable solutions far from the true PF besides the solutions close to the true PF. This situation is depicted in the solution plot illustrated in Fig. 4.

2) HV Metric: The HV metric computes the volume covered by the nondominated set P_A achieved by the algorithm. For each solution $a \in P_A$, a hypercube v_a is formed with the solution a and a reference point as the diagonal corners. Here the reference point can be defined as a vector of worst objective function values. The value of HV is obtained by computing the union of all hypercubes, as shown in

$$HV = volume \left(\bigcup_{a \in P_A} v_a \right). \tag{6}$$

The HV value is associated with both convergence to the PF and the diversity of the obtained solutions. Larger values of the HV usually imply more favorable solutions. In our experiments, the HV value would be presented in percentage, with a value of 100% representing a perfect coverage of the PF.

To investigate the algorithm performance comprehensively, the IGD value, HG value and the solution plots are all used in this paper to report the algorithm performance.

C. Experimental Configurations

In the experiments, five representative MOEAs are tested for comparison, including the NSGA-II [9], MOEA/D [24],

SPEA2 [14], GDE3 [11], and multiobjective density driven evolutionary algorithm (MODdEA) [30]. Specifically, NSGA-II and MOEA/D have received much attention since being proposed, and the MODdEA has performed competitively on both ZDT problems and the truly disconnected TDY problems. Each test is run 100 independent trials and is limited to a maximum of 2.5×10^4 FEs. For simplicity, both MOEA/D and the proposed MOEA-DLA use the weighted sum decomposition method. The parameter configurations for all of the six algorithms can be referred to the supplementary material.

For the MOEA-DLA, the DE crossover operator and polynomial mutation operator are selected to implement the algorithm, and the parameters in these operators take typical values. In the following experiments, the population size N is set as 100. The maximum number of individuals in global archive $\mathbf{G}_{\max}$ is set to 100, and the limit of one sub-archive $S_{\max}$ is set to 10.

All of the algorithms for comparison are implemented with configurations as suggested in their original literature, except that for the NSGA-II and the MOEA/D the enhanced versions with DE crossover operator are selected. Since MOEA-DLA consumes $2N$ FEs per generation compared to N FEs for the other five algorithms, a population size of $N = 200$ is adopted on bi-objective cases for all the other algorithms except the MODdEA, which is designed as an algorithm with small population.

D. Results and Discussion

The experimental results for the tested algorithms are presented in Fig. 5 and Table I. In Table I, the mean results are reported for each algorithm. In the table, for each algorithm the IGD values are averaged over all trials. On each problem, the averaged value of the IGD, the standard deviation and the rank according to the average IGD are presented, with the best value among six algorithms marked in **bold**. The Wilcoxon rank sum test [57] with significance level $a = 0.05$ is used to study the significance of difference between the results achieved by the MOEA-DLA and the algorithms for comparison. Values of z that are larger than 1.645 imply that the proposed MOEA-DLA is significantly better, whereas $z < -1.645$ means significantly worse results. In Table I, the columns where the proposed MOEA-DLA achieved significantly better results are marked with gray background. In the figure, both the true PF and the solutions achieved by the NSGA-II, MOEA/D, MODdEA, and MOEA-DLA are plotted. For page limit, only the most competitive algorithms according to Table I are presented in the figure. For the NSGA-II and MOEA/D, the final population is reported, whereas for the MODdEA and MOEA-DLA the nondominated global archive is reported. All of the algorithms report the trial with the best IGD value.

1) Comparison on the ZDT, DTLZ, and WFG Problems: It is observed from Fig. 5 and Table I that on the ZDT, WFG, and DTLZ problems the proposed MOEA-DLA generally obtained solutions closest to PF among all of the algorithms. The MOEA-DLA ranks first on three ZDT problems and two WFG problems, ranks second on one ZDT problem and one WFG

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

8

IEEE TRANSACTIONS ON CYBERNETICS
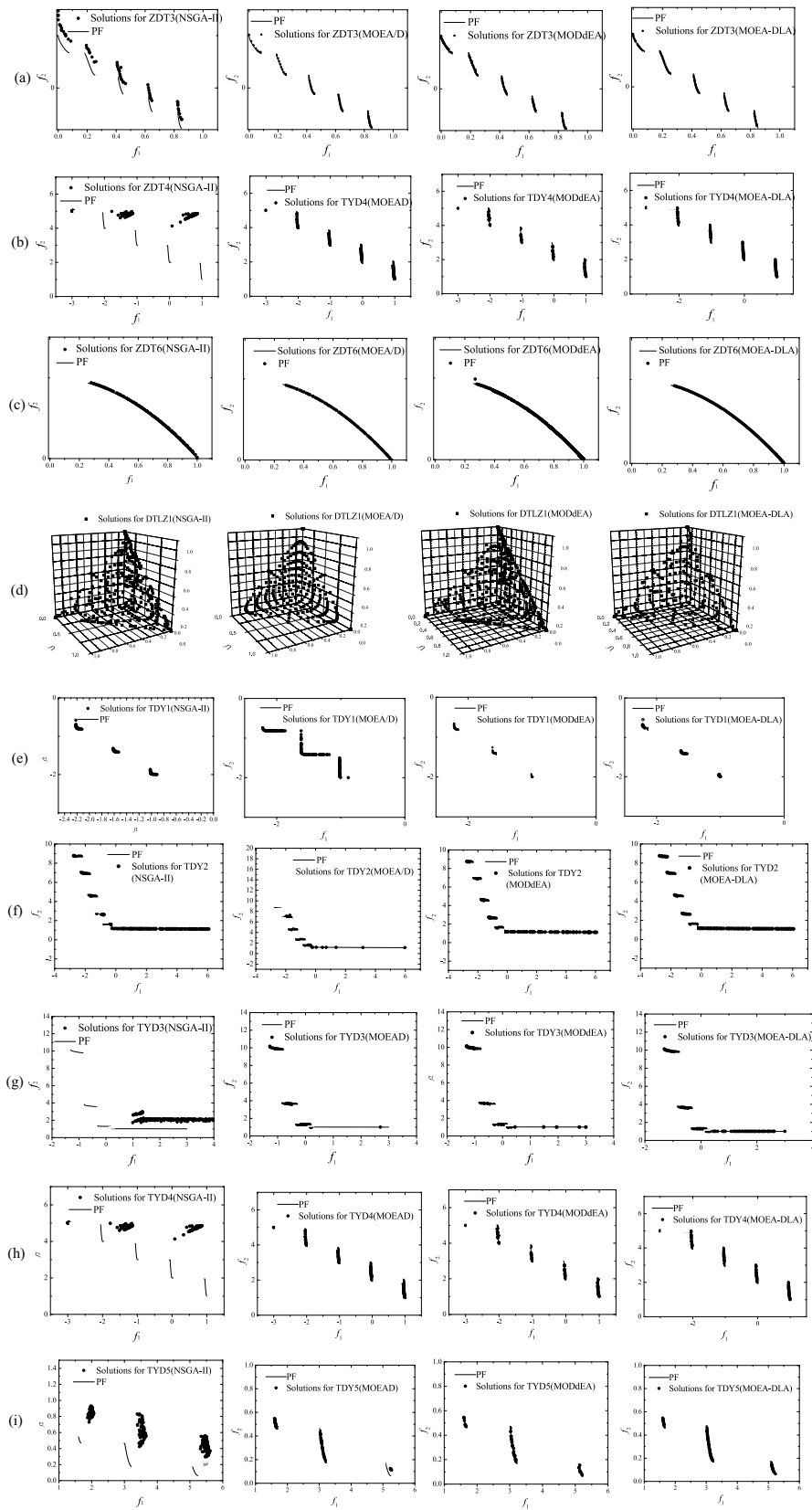
Fig. 5. All solutions obtained by the NSGA-II, MOEA/D, MODdEA, and MOEA-DLA on (a) ZDT3, (b) ZDT4, (c) ZDT6, (d) DTLZ1, (e) TDY1, (f) TDY2, (g) TDY3, (h) TDY4, and (i) TDY5.

problem, and ranks third on one problem. The z value of two-sample Wilcoxon test indicated that on five of these problems the proposed MOEA-DLA has significant advantage over

at least four of the algorithms for comparison. Especially for the problem ZDT4, the MOEA-DLA converged to the PF and achieved favorable IGD and HV values; while all the other

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

CHEN *et al.*: EA WITH DOUBLE-LEVEL ARCHIVES FOR MULTIOBJECTIVE OPTIMIZATION 9

TABLE I
AVERAGE IGD VALUES AND HV VALUES FOR THE SIX ALGORITHMS ON 12 TEST PROBLEMS

Func.			NSGA-II	MOEA/D	SPEA2	GDE3	MODdEA	MOEA-DLA
ZDT3	IGD	Avg.	0.01300	0.01154	0.03253	0.3452	0.00709	**0.005699**
		Z-value	2.194	2.032	5.356	10.425	1.932	-
	HV	Avg.(%)	96.42	96.73	95.25	78.39	98.88	**99.14**
		Z-value	2.932	2.982	3.123	11.335	1.945	-
ZDT4	IGD	Avg.	0.9605	0.006443	11.21	1.3563	0.03299	**0.005710**
		Z-value	5.323	1.834	11.343	8.235	3.527	-
	HV	Avg. (%)	60.26	98.32	19.23	50.29	96.17	**98.93**
		Z-value	12.356	2.423	11.115	10.845	3.654	-
ZDT6	IGD	Avg.	0.01305	0.005412	2.554	0.00143	**0.000713**	0.003868
		Z-value	3.684	2.859	9.235	*-2.134*	*-3.845*	-
	HV	Avg. (%)	96.67	98.79	31.32	99.06	**99.43**	98.95
		Z-value	4.684	1.859	11.235	*-1.834*	*-2.455*	-
WFG1	IGD	Avg.	1.58	2.23	11.93	2.31	1.342	**1.13**
		Z-value	3.235	2.353	11.434	6.453	1.734	-
	HV	Avg. (%)	58.32	49.25	12.45	49.34	60.39	**73.65**
		Z-value	4.245	8.352	12.216	9.354	1.893	-
WFG2	IGD	Avg.	0.62	0.95	9.03	1.12	**0.33**	0.53
		Z-value	2.235	2.983	11.354	4.532	*-2.564*	-
	HV	Avg. (%)	68.06	58.94	14.26	69.22	**75.35**	71.65
		Z-value	1.935	2.563	12.216	6.453	*-3.243*	-
WFG3	IGD	Avg.	0.48	0.72	12.24	1.59	0.44	**0.21**
		Z-value	3.235	3.451	12.216	10.733	2.532	-
	HV	Avg. (%)	69.74	62.37	8.24	56.37	71.94	**85.13**
		Z-value	3.956	4.472	12.216	9.365	3.268	-
DTLZ1	IGD	Avg.	0.086721	**0.055831**	20.234	1.76212	0.070123	0.06868
		Z-value	2.875	*-2.563*	12.216	10.252	2.437	-
	HV	Avg. (%)	90.87	**93.43**	6.83	24.49	91.58	92.16
		Z-value	3.263	*-2.532*	12.216	12.216	2.124	-
TDY1	IGD	Avg.	0.004900	0.005158	0.0405	0.2457	**0.002892**	0.004886
		Z-value	1.214	1.754	5.325	6.732	*-2.638*	-
	HV	Avg. (%)	97.52	97.14	92.85	83.27	**98.96**	98.03
		Z-value	1.484	2.121	4.743	6.356	*-2.539*	-
TDY2	IGD	Avg.	0.04053	0.3836	0.7382	4.3213	0.05611	**0.03198**
		Z-value	2.864	4.948	6.362	12.216	3.142	-
	HV	Avg. (%)	94.35	82.73	64.93	12.32	94.02	**95.68**
		Z-value	2.535	6.742	7.983	12.216	2.643	-
TDY3	IGD	Avg.	2.2301	0.2122	1.0423	4.5323	0.2598	**0.01593**
		Z-value	11.424	9.538	11.015	12.216	9.264	-
	HV	Avg. (%)	24.89	86.58	74.95	13.23	87.74	**96.94**
		Z-value	12.216	8.357	11.422	12.216	8.475	-
TDY4	IGD	Avg.	1.641	0.01893	0.7643	4.0123	0.06043	**0.01384**
		Z-value	12.216	1.856	11.532	12.216	6.945	-
	HV	Avg. (%)	45.76	96.76	75.45	5.39	93.34	**97.05**
		Z-value	12.216	1.527	11.943	12.216	7.395	-
TDY5	IGD	Avg.	0.3637	0.07518	0.8222	5.2314	0.01177	**0.004232**
		Z-value	12.216	8.437	12.216	12.216	3.869	-
	HV	Avg. (%)	82.85	92.18	60.83	4.86	97.39	**98.94**
		Z-value	12.216	9.583	12.216	12.216	3.268	-

algorithms except the MOEA/D failed to obtain a value of IGD lower than 0.01 and a value of HV larger than 97%. When compared with the MOEA/D, MOEA-DLA achieves similar solution quality in terms of distance to the PF and coverage over PF of ZDT4. The IGD values for MOEA-DLA shows the slight advantage of MOEA-DLA over the MOEA/D.
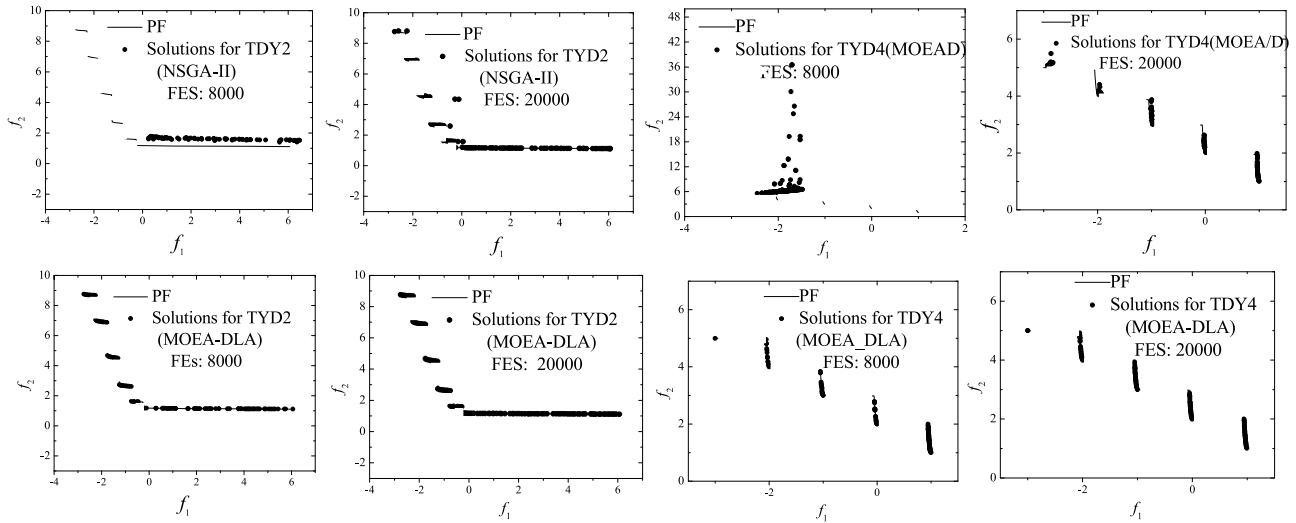
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

10   IEEE TRANSACTIONS ON CYBERNETICS

Fig. 6.   Solutions achieved by the MOEA-DLA and NSGA-II in 8000 and 20 000 FEs on TDY2 and TDY4.

TABLE II
AVERAGE IGD AND HV VALUE FOR MOEA/D AND MOEA-DLA WITH DIFFERENT DECOMPOSITION METHODS

Func.		MOEA-DLA (noSub)	MOEA-DLA (noGlobal)	Func.		MOEA-DLA (noSub)	MOEA-DLA (noGlobal)
TDY1	Avg.	**0.05148**	0.1523	TDY4	Avg.	0.3352	**0.01945**
	Std.	**0.03241**	0.04953		Std.	0.2353	**0.003522**
TDY2	Avg.	**0.2234**	0.5192	TDY5	Avg.	0.7346	**0.06312**
	Std.	**0.0934**	0.1344		Std.	0.4353	**0.01332**
TDY3	Avg.	1.2643	**0.5625**				
	Std.	0.6443	**0.05335**				

2) Comparison on TDY Problems: For the TDY problems, the MOEA-DLA shows obvious advantage over most algorithms for comparison, especially on ZDT3-ZDT6 where the advantage is significant according to the Wilcoxon test. Generally, these problems are difficult for most of the tested algorithms, especially for the NSGA-II, SPEA2, and GDE3 whose results have shown large values of IGD and small values of HV. Nevertheless, the MOEA-DLA ranks first on four problems and ranks second on one problem, and achieved IGD values lower than 0.1 on all the TDY problems. It can be observed that the MODdEA, whose original literature introduced the TDY problem, performed second to the MODdEA on TDY2-TDY5, and only achieved slight advantage on TDY1.

It is interesting that for TDY1 the MOEA/D obtains a considerable number of solutions away from the PF, while the algorithm achieved a small IGD value on the problem. Besides, the coverage for MOEA/D on TDY2 and TDY3 are unfavorable since one part of the PF is always uncovered. This situation is associated with the problem decomposition method in the MOEA/D and the shape of the PF. On the other hand, the proposed MOEA-DL overcomes the problem of MOEA/D on these TDY problems.

E. Search Speed of MOEA-DLA on TDY Problems

It is observed that on TDY2 both the MOEA-DLA and the NSGA-II achieved solutions close to the PF, and on TDY4 the MOEA-DLA and MOEA/D achieved

similar nondominated sets. In this case, the search speed is considered. The solutions for TDY2 and TDY4 achieved by the NSGA-II, MOEA/D, and MOEA-DLA within 8000 and 20 000 FEs are plotted in Fig. 6. According to the figure, the MOEA-DLA, which optimizes all the sub-problems simultaneously with the multiobjective problem, converges close to the PF within 8000 FEs, while the NSGA-II and MOEA/D require more than 20 000 FEs to converge on TDY2 and TDY4, respectively.

F. Effects of the Double-Level Strategy

To study the effect of the proposed double-level archives, experiments are conducted to analyze MOEA-DLA without one level of archive. For convenience, we denote the MOEA-DLA without global archive MOEA-DLA-noGlobal and the MOEA-DLA without sub-archives MOEA-DLA-noSub. Cross-reproduction is deleted from both of the two versions, whereas self-reproduction is added to MOEA-DLA-noGlobal for fair comparison. Table II presents the average IGD values achieved by the two versions of MOEA-DLA. Specifically, in Fig. 7 the results on TDY2 and TDY3 are presented.

The results in Table II indicate that the MOEA-DLA (noSub) achieved better results on problems where the NSGA-II with problem-level nondominated sorting perform well (TDY1-TDY2), while MOEA-DLA (noGlobal) achieved better results on the problems where the MOEA/D based on decomposition are more competitive (TDY3-TDY5).

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

CHEN *et al.*: EA WITH DOUBLE-LEVEL ARCHIVES FOR MULTIOBJECTIVE OPTIMIZATION

11

TABLE III
AVERAGE IGD VALUE AND STANDARD DEVIATION FOR SBX OPERATORS IN THE MOEA-DLA, MOEA/D, AND NSGA-II

| Func.		NSGA-II(SBX)	MOEA/D(SBX)	MOEA-DLA(SBX)	Func.		NSGA-II(SBX)	MOEA/D(SBX)	MOEA-DLA(SBX)
ZDT3	Avg.	0.01538	0.01343	**0.008353**	TDY2	Avg.	0.1653	0.5324	**0.05064**
	Std.	0.00633	0.003254	**0.005633**		Std.	0.1325	0.1344	**0.02442**
ZDT4	Avg.	0.6735	0.007454	**0.006450**	TDY3	Avg.	1.0793	0.4532	**0.03567**
	Std.	0.3257	0.003432	**0.001354**		Std.	0.6443	0.05335	**0.01435**
ZDT6	Avg.	0.3536	**0.005246**	0.006648	TDY4	Avg.	0.3532	**0.01453**	0.03535
	Std.	0.05389	**0.001424**	0.002145		Std.	0.2456	**0.005432**	0.007755
TDY1	Avg.	0.04645	0.1212	**0.008745**	TDY5	Avg.	0.7346	0.06312	**0.01122**
	Std.	0.03245	0.04327	**0.005353**		Std.	0.4353	0.01332	**0.005432**
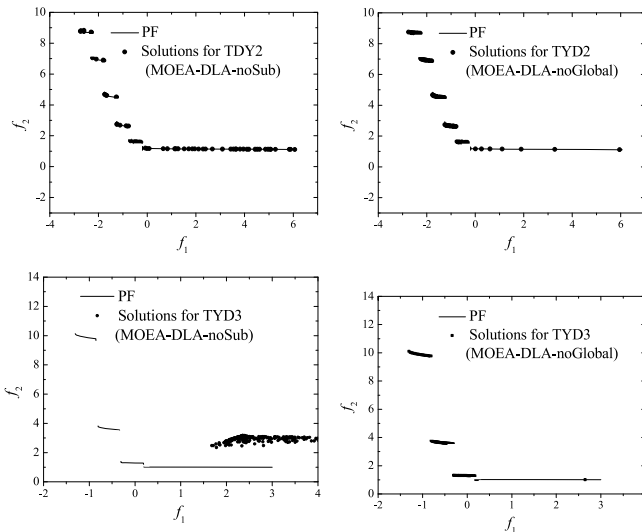
Fig. 7. Solutions achieved by MOEA-DLA-noGlobal and MOEA-DLA-noSub on TDY2 and TDY3.

It can be observed from Table II and Figs. 5 and 7 that the MOEA-DLA-noSub generally converges worse than MOEA-DLA-noGlobal and MOEA-DLA. The result is reasonable since MOEA-DLA-noSub only works at the multiobjective problem level, where the selection of individuals itself is a multiobjective problem. On the other hand, the MOEAs that works at sub-problem level optimize single-objective sub-problems with elitism simultaneously, which accelerates the convergence.

Interestingly, the advantage of MOEA-DLA-noSub over MOEA-DLA-noGlobal lies in the coverage of PF on TDY2, where MOEA-DLA-noSub converged to the PF. In Fig. 7, the MOEA-DLA-noGlobal distributes little solutions at the bottom of PF, which can be explained by the decomposition method. On the other hand, with multiobjective selection operator at the problem level, the MOEA-DLA-noSub and MOEA-DLA distribute the solutions evenly on the bottom of PF.

Comparing Table II and Fig. 7 with Table I and Fig. 5, it is observed that the MOEA-DLA outperforms MOEA-DLA-noGlobal in terms of solution distribution and MOEA-DLA-noSub in terms of convergence. It can be concluded that the MOEA-DLA combines the advantage of working at both problem level and sub-problem levels.

G. Effects of the MOEA-DLA Framework

The proposed MOEA-DLA is implemented with a DE crossover operator, and performed competitively compared

to the DE version of the NSGA-II and MOEA/D. To further investigate the effectiveness of MOEA-DLA framework, in the following experiments the SBX version of NSGA-II, MOEA/D, and MOEA-DLA are simulated.

Table III reports the simulation results of SBX version of the NSGA-II, MOEA/D, and MOEA-DLA. As shown in the table, the MOEA-DLA achieved best IGD values on seven out of the ten algorithms, and rank second on the other three algorithms. The results are highly consistent with those of the three algorithms in the DE version. It can be concluded that the framework of the MOEA-DLA algorithm contributes good performance in the case of both DE and SBX crossover operators.

V. CONCLUSION

A multiobjective optimization algorithm that works at both the problem level and the sub-problem level has been developed. Two different levels of archives are successfully adopted in the algorithm, i.e., the global archive at the problem level and sub-archives at a sub-problem level. In each generation, individuals are reproduced from self-reproduction with the global archive and cross-reproduction between the global archive and the sub-archives. All the archives are updated using the reproduced individuals. Experiments are conducted on 12 benchmark test problems to validate the effectiveness of the proposed MOEA-DLA algorithm. The MOEA-DLA has achieved competitive results on the widely used ZDT problems, and has shown advantage over some state-of-the-art algorithms on various disconnected problems in terms of distance to the PF, solution coverage and search speed. Besides, compared to the MOEA/D and NSGA-II frameworks, the MOEA-DLA has shown competitive advantage with alternative crossover operators.

REFERENCES

[1] K. Miettinen, *Nonlinear Multiobjective Optimization*. Norwell, MA, USA: Kluwer, 1999, ch. 1.
[2] J. Zhang *et al.*, "Evolutionary computation meets machine learning: A survey," *IEEE Comput. Intell. Mag.*, vol. 6, no. 4, pp. 68–75, Nov. 2011.
[3] T. Back, M. Emmerich, and O. M. Shir, "Evolutionary algorithms for real world applications," *IEEE Comput. Intell. Mag.*, vol. 3, no. 1, pp. 64–67, Jan. 2008.
[4] J. Horn, N. Nafploitis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proc. 1st IEEE Conf. Evol. Comput.*, Piscataway, NJ, USA, 1994, pp. 82–87.
[5] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1995.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

12

IEEE TRANSACTIONS ON CYBERNETICS

[6] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—A comparative case study," in *Parallel Problem Solving From Nature—PPSN V*, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds. Berlin, Germany: Springer-Verlag, 1998, pp. 292–301.

[7] J. Knowles and D. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization," in *Proc. 1999 Congr. Evol. Comput.*, Piscataway, NJ, USA, pp. 98–105.

[8] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: Methods and applications," Doctoral dissertation, Dept. Inf. Technol. Elect. Eng., Swiss Federal Inst. Technol., Zurich, Switzerland, 1999.

[9] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[10] S. Kukkonen and J. Lampinen, "An extension of generalized differential evolution for multiobjective optimization with constraints," in *Parallel Problem Solving from Nature—PPSN VIII* (Lecture Notes in Computer Science), vol. 3242. Berlin, Germany: Springer, 2001, pp. 752–761.

[11] S. Kukkonen and J. Lampinen, "GDE3: The third evolution step of generalized differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, vol. 1. Edinburgh, U.K., Jul. 2005, pp. 443–450.

[12] B. V. Babu, P. G. Chakole, and J. H. Syed Mubeen, "Multiobjective differential evolution (MODE) for optimization of adiabatic styrene reactor," *Chem. Eng. Sci.*, vol. 60, no. 17, pp. 4822–4837, 2005.

[13] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. Genet. Algorithms*, 1985, pp. 93–100.

[14] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Methods Des. Optim. Control Appl. Ind. Probl.*, Athens, Greece, pp. 95–100.

[15] Z. H. Zhan *et al.*, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, Apr. 2013.

[16] W. F. Leong and G. G. Yen, "PSO-based multiobjective optimization with dynamic population size and adaptive local archives," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 5, pp. 1270–1293, Oct. 2008.

[17] G. G. Yen and W. F. Leong, "Dynamic multiple swarms in multiobjective particle swarm optimization," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 4, pp. 890–911, Jul. 2009.

[18] S. Gao *et al.*, "An orthogonal multiobjective evolutionary algorithm with lower-dimensional crossover," in *Proc. IEEE Congr. Evol. Comput.*, Trondheim, Norway, May 2009, pp. 1959–1964.

[19] Y. Wang, C. Dang, H. Li, L. Han, and J. Wei, "A clustering multiobjective evolutionary algorithm based on orthogonal and uniform design," in *Proc. IEEE Congr. Evol. Comput.*, Trondheim, Norway, Jul. 2009, pp. 2927–2933.

[20] E. J. Hughes, "Multiple single objective Pareto sampling," in *Proc. Congr. Evol. Comput.*, vol. 4. 2003, pp. 2678–2684.

[21] Y. Jin, M. Olhofer, and B. Sendhoff, "Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how?" in *Proc. Genet. Evol. Comput. Conf.*, San Francisco, CA, USA, 2001, pp. 1042–1049.

[22] R. Jin, W. Chen, and A. Sudjianto, "On sequential sampling for global metamodeling in engineering design," in *Proc. Des. Eng. Tech. Conf. (DETC)*, vol. 2. Montreal, QC, Canada, 2002, pp. 539–548.

[23] Y. Jin, T. Okabe, and B. Sendho, "Adapting weighted aggregation for multiobjective evolution strategies," in *Evolutionary Multi-Criterion Optimization*. Berlin, Germany: Springer, 2001, pp. 96–110.

[24] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[25] C. M. Chen, Y. P. Chen, and Q. Zhang, "Enhancing MOEA/D with guided mutation and priority update for multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Trondheim, Norway, May 2009, pp. 209–216.

[26] R. Storn and K. V. Price, "Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces," Inst. Company Secretaries India, Chennai, India, Tech. Rep. TR-95-012, 1995.

[27] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[28] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition," in *Evolutionary Multi-Criterion Optimization,* vol. 7811. Berlin, Germany: Springer, 2013, pp. 428–442.

[29] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Towards understanding the cost of adaptation in decomposition-based optimization algorithms," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Manchester, U.K., 2013, pp. 615–620.

[30] C. K. Chow and S. Y. Yuen, "A multiobjective evolutionary algorithm that diversifies population by its density," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 149–172, Apr. 2012.

[31] H. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2013.

[32] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Evolutionary Multi-Criterion Optimization* (Lecture Notes in Computer Science), vol. 3410. Berlin, Germany: Springer-Verlag, Mar. 2005, pp. 280–295.

[33] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *Proc. 5th Int. Conf. Genet. Algorithms*, San Mateo, CA, USA, 1993, pp. 416–423.

[34] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1995.

[35] I. Karahan and M. Köksalan, "A territory defining multiobjective evolutionary algorithms and preference incorporation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 4, pp. 636–664, Aug. 2010.

[36] E. J. Hughes, "MSOPS-II: A general-purpose many-objective optimiser," in *Proc. IEEE Congr. Evol. Comput.*, Singapore, 2007, pp. 3944–3951.

[37] E. J. Hughes, "Many-objective directed evolutionary line search," in *Proc. 13th Annu. Conf. Genet. Evol. Comput.*, Dublin, Ireland, 2011, pp. 761–768.

[38] D. Liu, K. C. Tan, C. K. Goh, and W. K. Ho, "A multiobjective memetic algorithm based on particle swarm optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 42–50, Feb. 2007.

[39] L. Ke, Q. Zhang, and R. Battiti, "Hybridization of decomposition and local search for multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1808–1820, Oct. 2014.

[40] R. Wang, R. Purshouse, and P. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 474–494, Aug. 2013.

[41] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, 2014, pp. 577–601.

[42] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.

[43] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for Pareto-based algorithms in many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 348–365, Jun. 2011.

[44] K. Miettinen and M. M. Mäkelä, "On scalarizing functions in multiobjective optimization," *OR Spectr.*, vol. 24, no. 2, pp. 193–213, 2002.

[45] K. Miettinen, F. Ruiz, and A. Wierzbicki, "Introduction to multiobjective optimization: Interactive approaches," in *Multiobjective Optimization*, vol. 5252, J. Branke, K. Deb, K. Miettinen, and R. Słowiński, Eds. Berlin, Germany: Springer, 2008, pp. 27–57.

[46] B. Y. Qu and P. N. Suganthan, "Multiobjective evolutionary programming without non-domination sorting is up to twenty times faster," in *Proc. IEEE Congr. Evol. Comput.*, Trondheim, Norway, May 2009, pp. 2934–2939.

[47] I. Das and J.E. Dennis, "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems," SIAM Journal on Optimization, vol. 8, no. 3, 1998, pp. 631–657.

[48] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*. New York, NY, USA: Wiley, 1997, chs. 1 and 10.

[49] P. J. Bentley and J. P. Wakefield, "Finding acceptable solutions in the Pareto-optimal range using multiobjective genetic algorithms," in *Proc. 2nd On-Line World Conf. Soft Comput. Eng. Des. Manuf.*, Jun. 1997, pp. 231–240.

[50] I. C. Parmee and D. Cvetkovic, "Preferences and their application in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 42–57, Feb. 2002.

[51] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," in *Foundations of Genetic Algorithms 2*. San Mateo, CA, USA: Morgan Kaufman, 1993, pp. 187–202.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[52] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.

[53] S. Tsutsui, M. Yamamura, and T. Higuchi, "Multi-parent recombination with simplex crossover in real coded genetic algorithms," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Orlando, FL, USA, 1999, pp. 657–664.

[54] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multiobjective optimization test problems," in *Proc. IEEE Congr. Evol. Comput.*, Honolulu, HI, USA, 2002, pp. 825–830.

[55] Q. Zhang *et al.*, "Multiobjective optimization test instances for the CEC 2009 special session and competition," Univ. Essex, Essex, U.K., Nanyang Technol. Univ., Singapore, Tech. Rep. CES-487, 2008.

[56] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[57] W. J. Connover, *Practical Nonparametric Statistics*, 3rd ed. New York, NY, USA: Wiley, 1999, ch. 5.

Ni Chen (S'11) received the B.S. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2009, where she is currently pursuing the Ph.D. degree.

Her current research interests include global optimization, evolutionary algorithm, multiobjective optimization, and their applications in scheduling.

Wei-Neng Chen (S'07–M'12) received the bachelor's and Ph.D. degrees from the Department of Computer Science, Sun Yat-sen University, Guangzhou, China, in 2006 and 2012, respectively.

He is currently an Associate Professor with the School of Advanced Computing, Sun Yat-sen University. His current research interests include swarm intelligence algorithms and their applications on cloud computing, financial optimization, operations research, and software engineering. He has published over 30 papers in international journals and conferences.

Prof. Chen was the recipient of the China Computer Federation Outstanding Dissertation in 2012.

Yue-Jiao Gong (S'10) received the B.S. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2010, where she is currently pursuing the Ph.D. degree.

Her current research interests include artificial intelligence, evolutionary computation, swarm intelligence, and their applications in design and optimization of intelligent transportation systems, wireless sensor networks, and RFID systems.

Zhi-Hui Zhan (S'09–M'13) received the bachelor's and Ph.D. degrees from the Department of Computer Science, Sun Yat-sen University, Guangzhou, China, in 2007 and 2013, respectively.

He is currently a Lecturer with Sun Yat-sen University. His current research interests include particle swarm optimization, differential evolution, ant colony optimization, genetic algorithms, and their applications in real-world problems.

Dr. Zhan was the recipient of the China Computer Federation Outstanding Dissertation in 2013.

Jun Zhang (M'02–SM'08) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2002.

He is currently a Changjiang Chair Professor with the School of Advanced Computing, Sun Yat-sen University, Guangzhou, China. His current research interests include computational intelligence, cloud computing, high performance computing, data mining, wireless sensor networks, operations research, and power electronic circuits. He has published over 100 technical papers in his research areas.

Dr. Zhang was the recipient of the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011 and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and the IEEE TRANSACTIONS ON CYBERNETICS.

Yun Li (S'87-M'90) received the B.S. degree in radio electronics science from Sichuan University, Chengdu, China, in 1984, the M.Eng. degree in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, in 1987, and the Ph.D. degree in parallel processing for control engineering from the University of Strathclyde, Glasgow, U.K., in 1990.

From 1989 to 1990, he was with U.K. National Engineering Laboratory and Industrial Systems and Control Ltd., Glasgow, U.K. He joined the University of Glasgow, Glasgow, U.K., as a Lecturer in 1991, where he is currently a Professor of Systems Engineering. He served as a Founding Director with the University of Glasgow Singapore, Singapore, from 2011 to 2013 and acted as the Founding Director of the university's international joint program with the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2013. He established the IEEE Computer-Aided Control System Design Evolutionary Computation Working Group and European Network of Excellence in Evolutionary Computing Workgroup on Systems, Control, and Drives in 1998. He was invited to Kumamoto University, Kumamoto, Japan, as a Visiting Professor in 2002 and is currently a Visiting Professor with UESTC and Sun Yat-sen University, Guangzhou, China. He has authored the interactive online courseware EA Demo for *Evolutionary Computation* and has over 180 publications.

Dr. Li is a Chartered Engineer.

Yu-Song Tan (M'07) received the Ph.D. degree from the School of Computer Science, National University of Defense Technology, Changsha, China, in 2004.

He is currently a Professor with the National University of Defense Technology. His current research interests include operating system, cloud computing, and high performance computing. He has published over 50 technical papers in his research areas.

Prof. Tan has participated as a Core Member in the Research of Tianhe Supercomputer Series. He is currently leading a team to develop large-scale cloud platforms and big data processing platforms that is deployed on the Tianhe-2 in the National Supercomputing Center in Guangzhou as well as other systems.