

# A Competitive-Cooperative Coevolutionary Paradigm for Dynamic Multiobjective Optimization

Chi-Keong Goh and Kay Chen Tan

**Abstract**—In addition to the need for satisfying several competing objectives, many real-world applications are also dynamic and require the optimization algorithm to track the changing optimum over time. This paper proposes a new coevolutionary paradigm that hybridizes competitive and cooperative mechanisms observed in nature to solve multiobjective optimization problems and to track the Pareto front in a dynamic environment. The main idea of competitive-cooperative coevolution is to allow the decomposition process of the optimization problem to adapt and emerge rather than being hand designed and fixed at the start of the evolutionary optimization process. In particular, each species subpopulation will compete to represent a particular subcomponent of the multiobjective problem, while the eventual winners will cooperate to evolve for better solutions. Through such an iterative process of competition and cooperation, the various subcomponents are optimized by different species subpopulations based on the optimization requirements of that particular time instant, enabling the coevolutionary algorithm to handle both the static and dynamic multiobjective problems. The effectiveness of the competitive-cooperation coevolutionary algorithm (COEA) in static environments is validated against various multiobjective evolutionary algorithms upon different benchmark problems characterized by various difficulties in local optimality, discontinuity, nonconvexity, and high-dimensionality. In addition, extensive studies are also conducted to examine the capability of dynamic COEA (dCOEA) in tracking the Pareto front as it changes with time in dynamic environments.

**Index Terms**—Coevolution, dynamic multiobjective optimization, evolutionary algorithms.

## I. INTRODUCTION

**M**ULTIOBJECTIVE EVOLUTIONARY ALGORITHMS (MOEAs) are a class of stochastic optimization techniques that simulate biological evolution to solve multiobjective (MO) problems. There have been significant contributions made to the field of evolutionary multiobjective optimization (EMOO) in the past two decades, as a result of intense research examining topics such as fitness assignment [15], [16], diversity preservation [13], [35], and elitism [3], [39], [57]. These studies have great influences on algorithmic development in MO optimization, resulting in design trends such as the emergence of Pareto-based techniques as the dom-

inant approach and the incorporation of elitist mechanisms in the state-of-the-arts.

Although it has been established that MOEAs are powerful optimization tools, most of these works are confined to static optimization problems. However, many real-world problems are also dynamic in nature. In such a situation, the optimal Pareto front is unlikely to remain invariant and past optimal solutions must be adapted to reflect the current requirements. Therefore, the optimization goal is not only to evolve a near-optimal and diverse Pareto front, but also to track the front as it changes with time. While there exist a number of studies on evolutionary optimization in dynamic environments, most of them are restricted to the domain of single-objective (SO) problems. Comprehensive discussions on dynamic SO evolutionary algorithms (SOEAs) can be found in [4] and [45]. On the contrary, the application of MOEAs to dynamic MO problems is explored only recently in [10], [14], [23], [43], and [66].

In a certain sense, the dynamic MO problem can be considered as the consecutive optimization of different time-constrained MO problems with varying complexities. It is imperative that MOEA must be capable of attaining a fast convergence in order to find the optimal solution set before it changes and becomes obsolete. However, a fast convergence also implies a rapid loss of diversity during the optimization process, which inevitably leads to the difficulty of tracking the dynamic Pareto front. It is, thus, necessary to maintain and generate sufficient diversity in order to explore the search space when the MO problem changes in a dynamic environment.

The coevolutionary paradigm, inspired by the reciprocal evolutionary change driven by the cooperative [47], [50] or competitive interaction [51] between different species, has been extended successfully to MO optimization recently [7], [28], [32], [41], [42], [56]. Several studies [49], [64] have shown that the introduction of ecological models and coevolutionary architectures are effective methods to improve the efficacy of canonical evolutionary algorithms. For instance, Tan *et al.* [56] demonstrated that a fast convergence can be achieved by coevolution while maintaining a good diversity of solutions. MO coevolutionary algorithms (MOCAs) are particularly suitable for dynamic MO optimization, where the high speed of convergence can potentially be exploited for adapting quickly to the changing environment. In addition, the works in [2] and [50] also demonstrated that both competitive and cooperative coevolutions have their unique advantage for maintaining diversity in the species subpopulation.

On the other hand, successful implementation of coevolution requires an appropriate method of problem decomposition in the search space, which may not be known *a priori* and may change with time in a dynamic MO problem. In order to exploit the mechanisms of fast convergence and good diversity in

Manuscript received November 02, 2006; revised May 29, 2007, September 12, 2007, and November 19, 2007. First published June 27, 2008; current version published January 30, 2009.

C.-K. Goh is with the Data Storage Institute, Agency for Science, Technology and Research, Singapore 117608, Singapore (e-mail: Goh\_Chi\_Keong@dsi.a-star.edu.sg).

K. C. Tan is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576 (e-mail: letankc@nus.edu.sg).

Digital Object Identifier 10.1109/TEVC.2008.920671

coevolution for dynamic MO optimization, this paper presents a new coevolutionary paradigm that incorporates the competitive and cooperative mechanisms observed in nature to facilitate adaptive problem decomposition in coevolution. In particular, each species subpopulation will compete to represent a particular subcomponent of the MO problem, while the eventual winners will cooperate to evolve for better solutions. Through such an iterative process of competition and cooperation, the various subcomponents are optimized by different species subpopulations based on the optimization requirements of that particular time instant, which allows the MOCA to handle both static and dynamic MO problems. A competitive-cooperation coevolutionary algorithm (COEA) for static environment is designed based on the proposed coevolutionary paradigm, which is subsequently extended as a dynamic COEA (dCOEA) to deal with dynamic MO optimization problems.

The organization of this paper is as follows. Section II provides a brief description of the dynamic MO optimization problem and related works. Section III describes the different coevolutionary paradigms and introduces the general framework of the proposed competitive-cooperative coevolutionary paradigm. The cooperative and competitive mechanisms for the proposed COEA are presented in Section IV. Section V describes the diversity and outdated archived solution handling schemes adopted in dCOEA for dealing with the dynamic environments. Extensive empirical studies are conducted to compare and analyze the dynamics and effects of COEA and dCOEA in static and dynamic environments, respectively, in Section VI and Section VII. Conclusions are drawn in Section VIII.

## II. BACKGROUND INFORMATION

### A. Problem Definition

Without loss of generality, a minimization problem is considered here. The dynamic MO problem can be formally defined as

$$\begin{aligned} \min_{\vec{x} \in \vec{X}^{n_x}} \vec{f}(\vec{x}, t) &= \{f_1(\vec{x}, t), f_2(\vec{x}, t), \dots, f_M(\vec{x}, t)\} \\ \text{s.t. } \vec{g}(\vec{x}, t) &> 0, \vec{h}(\vec{x}, t) = 0 \end{aligned} \quad (1)$$

where  $\vec{x}$  is the vector of decision variables bounded by the decision space,  $\vec{X}^{n_x}$ ;  $\vec{f}$  is the set of objectives to be minimized with respect to time,  $t$ . The terms “solution space” and “search space” are often used to denote the decision space and will be used interchangeably throughout this paper. The functions of  $\vec{g}$  and  $\vec{h}$  represent the set of inequality and equality constraints, respectively, which defines the feasible region of the  $n_x$ -dimensional continuous or discrete feasible solution space that changes with  $t$ .

One of the main differences between SO and MO optimization is that MO problems constitute a multidimensional objective space  $\vec{F}^M$ . In addition, a set of solutions representing the tradeoff among the different objectives rather than an unique optimal solution is sought in MO optimization. This set of solutions is also known as the Pareto optimal set and these solutions are also termed “noninferior,” “admissible,” or “efficient” solutions. The corresponding objective vectors of these solutions are termed “nondominated” and each objective component of

any nondominated solution in the Pareto optimal set can only be improved by degrading at least one of its other objective components [55].

In static MO optimization, the concept of Pareto dominance and Pareto optimality will form the basis of solution quality. Specifically, the important concepts are defined as follows.

**Definition 1:** Pareto Dominance:  $\vec{f}_1 \in \vec{F}^M$  dominates  $\vec{f}_2 \in \vec{F}^M$ , denoted by  $\vec{f}_1 \prec \vec{f}_2$  iff  $f_{1,i} \leq f_{2,i} \forall i \in \{1, 2, \dots, M\}$  and  $f_{1,j} < f_{2,j} \exists j \in \{1, 2, \dots, M\}$

**Definition 2:** Pareto Optimal Front: The Pareto optimal front, denoted as  $\text{PF}^*$ , is the set of nondominated solutions with respect to the objective space such that  $\text{PF}^* = \{\vec{f}_i^* \mid \nexists \vec{f}_j \prec \vec{f}_i^*, \vec{f}_j \in \vec{F}^M\}$ .

**Definition 3:** Dynamic Pareto Optimal Front: The Pareto optimal front at time  $t$ , denoted as  $\text{PF}_t^*$ , is the set of nondominated solutions with respect to the objective space at  $t$  such that  $\text{PF}_t^* = \{\vec{f}_{i,t}^* \mid \nexists \vec{f}_{j,t} \prec \vec{f}_{i,t}^*, \vec{f}_{j,t} \in \vec{F}^M\}$ .

**Definition 4:** Pareto Optimal Set: The Pareto optimal set, denoted as  $\text{PS}^*$ , is the set of solutions that are nondominated in the objective space such that  $\text{PS}^* = \{\vec{x}_i^* \mid \nexists \vec{f}(\vec{x}_j) \prec \vec{f}(\vec{x}_i^*), \vec{f}(\vec{x}_j) \in \vec{F}^M\}$ .

**Definition 5:** Dynamic Pareto Optimal Set: The Pareto optimal set, denoted as  $\text{PS}_t^*$ , is the set of nondominated solutions with respect to the decision space such that  $\text{PS}_t^* = \{\vec{x}_i^* \mid \nexists \vec{f}(\vec{x}_j, t) \prec \vec{f}(\vec{x}_i^*, t), \vec{f}(\vec{x}_j, t) \in \vec{F}^M\}$ .

In dynamic SO problems, a solution can either deteriorate due to a shift in the fitness landscape or becomes obsolete due to the emergence of a new optimum. Likewise, such traits can be found in dynamic MO problems, except that we are now dealing with a set of solutions which make the tracking process a lot trickier. Another distinct characteristic of dynamic MO problems is that the shape and distribution of  $\text{PF}_t^*$  are susceptible to change as well. This makes it necessary to consider the dynamics in both feature spaces during the investigation of dynamic MOEAs.

Accordingly, Farina *et al.* [14] identified four different types of dynamic MO problems according to the changes affecting the Pareto optimal front and the Pareto optimal set.

- Type I, where  $\text{PS}_t^*$  changes while  $\text{PF}_t^*$  remains invariant.
- Type II, where both  $\text{PS}_t^*$  and  $\text{PF}_t^*$  changes.
- Type III, where  $\text{PF}_t^*$  changes while  $\text{PS}_t^*$  remains invariant.
- Type IV, where both  $\text{PS}_t^*$  and  $\text{PF}_t^*$  remain invariant.

Farina *et al.* further noted that, even though both  $\text{PS}_t^*$  and  $\text{PF}_t^*$  are time-invariant in Type IV problems, it is possible that the fitness topology is changing with time. This alone may pose sufficient challenge to dynamic MOEAs in finding the desired solutions.

### B. Related Work

As mentioned in Section I, studies on evolutionary dynamic optimization are mainly carried out in the domain of SO optimization. Nonetheless, the concepts adopted in these works are also applicable to dynamic MO optimization. From the literature, it is clear that EAs for dynamic optimization in any problem domain must be capable of detecting changes in the fitness landscape and maintaining diversity within the evolving population.

Generally, it is usually assumed that a change in the fitness landscape can be detected [4], while the different techniques

proposed to handle population diversity are based on the following three approaches.

- **Diversity Introduction:** This approach introduces diversity upon the detection of landscape change [6], [21], [62]. Random restart or reinitialization is one of the simplest techniques for generating diversity. Other common techniques include hypermutation where mutation is increased dramatically, and variable local search where mutation is increased gradually if no improvement is achieved. These approaches can be easily extended to MOEAs. The main drawback of this approach is that information gained is lost after the introduction of diversity.
- **Diversity Maintenance:** This approach sought to maintain diversity throughout the run [18], [20], [44]. One of the techniques that can be easily incorporated in MOEAs is the random immigrant which is conceptually similar to the idea of random restart. In the approach of random immigrant, however, random individuals are introduced into the evolving population at fixed intervals and only a part of the population is replaced. Diversity preservation techniques such as niche sharing can also be used, except that the diversity assessment should be performed in the decision space.
- **Multiple Population:** The basic idea of applying multiple populations is to conduct simultaneous exploration in different regions in order to track any change or emergence of new optimal solutions [5], [60], [65]. Typically, this approach involves a population which exploits the current optimal solution, while the other populations are encouraged to explore the search space.

Farina *et al.* [14] presented the FDA continuous dynamic test suite based upon the ZDT test functions [70]. An extension of the dynamic traveling salesman problem (DTSP) is made and a dynamic MO optimal control problem is also suggested as an instance of real-world problem. An interesting approach of aggregating objective functions of existing test problems through dynamically changing weights to form a lower dimensional dynamic problem is proposed in [31]. On the other hand, Mehnen *et al.* [43] considered the behavior of MOEAs and presented the DSW test functions that facilitate theoretical analysis in dynamic MO optimization. The techniques of strength Pareto evolutionary algorithm 2 (SPEA2) [69], nondominated sorting genetic algorithm II (NSGAII) [11], and multiple SO Pareto sampling (MSOPS) [26] are employed as test algorithms, and the effects of different genetic operators in dynamic environments are studied upon the FDA [14] and DSW [43] test suites. However, it is not clear how these different MOEAs are adapted for MO optimization in dynamic environments.

It should be noted that modifications to existing MOEAs for dynamic optimization need to account for the following.

- **Outdated elitist solutions:** One potential problem of MOEAs in dynamic environment is the exploitation of solutions. When the landscape changes, the discovered nondominated solutions may no longer be indicative of the optimal solution set at that particular time instant and thus may misguide the optimization process.
- **Diversity loss:** Diversity preservation techniques adopted in MOEAs are designed to maintain diversity in the objec-

tive space with the aim of obtaining a well-distributed and well-spread Pareto front. Unless the new optimal solution set is within the vicinity of the previous optimal solution set, it is unlikely that the MOEA is able to track any landscape changes.

Deb *et al.* [10] extended NSGAII for the optimization of dynamic hydrothermal power scheduling problem. In order to detect problem changes, 10% of individuals in the population are selected randomly and reevaluated in every generation. When a change is detected, all outdated solutions are reevaluated, and diversity is introduced either through random initialization or mutation. Contrary to the norm in SO optimization, only a portion of the evolving population undergo the diversity enhancement process. The effect of the population ratio is also investigated, which shows that random initialization is more susceptible to the setting of the population ratio in this approach.

One of the challenges in evolutionary dynamic optimization is to exploit past information for better tracking performance [4]. It is often inefficient to restart the entire optimization process every time a change in the landscape is detected, particularly when the new optimal solution set is somewhat similar to the previous solutions. In [66], Zeng *et al.* proposed a dynamic orthogonal MOEA (DOMOE) as the baseline MOEA for dynamic MO optimization. The DOMOE regards the dynamic MO problem as a new problem instance after every landscape changes. However, it only exploits past information using the nondominated solutions found prior to the landscape changes as the new initial population. Here, the diversity is maintained in the evolving population through a linear crossover operator, which generates an offspring different from its parents.

Instead of reintroducing past optimal solutions into the evolving population, information is exploited to predict future behavior of the dynamic MO problem in [23]. An autoregressive model is employed to estimate the location of optimal solutions for the next change, and the generated individuals are used to seed the population when a change in the problem landscape is detected.

### III. COMPETITION, COOPERATION, AND COMPETITIVE-COOPERATION IN COEVOLUTION

Existing coevolutionary techniques can be divided into two main classes: competitive coevolution and cooperative coevolution. Regardless of the approach adopted, the design of coevolutionary algorithms for MO optimization requires one to address many issues that are unique to the MO problems. In this aspect, insights such as incorporation of various elitist and diversity mechanisms obtained from the design of MOEAs can be similarly exploited in the design of MOCAs. On the other hand, successful implementation of coevolution requires one to consider various design issues [49], such as problem decomposition, handling of parameter interactions, and credit assignment. The issues of problem decomposition and parameter interactions are often problem dependent, and the approaches for solving these issues may not be known *a priori*. These factors motivated the work for an alternative coevolutionary model presented in this paper.

This section begins with a review of both the competitive and cooperative coevolutionary algorithms for MO optimization, highlighting various features and limitations of existing approaches. A competitive-cooperative coevolutionary model is then proposed, including discussions on how the different design issues in coevolutionary algorithms are addressed.

#### A. Competitive Coevolution

The model of competitive coevolution is often compared to predator-prey or host-parasite interactions, where preys (or hosts) implement the potential solutions to the optimization problem, while the predators (or parasites) implement individual “fitness-cases.” When applying this idea into optimization [1], [51], there are usually two subpopulations and an inverse fitness interaction exists between the two subpopulations. To survive, the losing subpopulation adapts to counter the winning subpopulation in order to become the new winner.

Although the competitive coevolution has been applied in many SOEA studies [24], [48], this model is rarely investigated in the domain of EMOO. Laumanns *et al.* [40] embodied the model of competitive coevolution in MO optimization through a spatial Predator-Prey model. In this model, the solutions are the preys and the associated solution vectors are represented as vertices on an undirected and connected graph. There are as many predators as the number of objectives and these predators perform a random walk on the graph along its associated objective. The worst prey in the neighborhood of the predator in terms of the associated objective will be replaced by its offspring.

Lohn *et al.* [41] presented a different competitive coevolutionary model which contains population of candidate solutions and target population with the target objective vectors. A distinct characteristic of this algorithm is the lack of any explicit diversity preservation mechanism to guide the coevolutionary optimization process. Empirical studies are conducted with well-known MOEAs such as SPEA and NSGA, and performance of this competitive MOEA is found to be better than the test algorithms.

There are several limitations to this coevolutionary model for numerical optimization. While competitive coevolution is a natural model for evolving objects such as game playing programs for which it is difficult to write an external fitness function, the need to hand-decompose the problem into antagonistic subcomponents places severe limitation on its range of applicability. Adding to its complexity is the need to adapt the predator population, which is the population of target vectors in the case of [41], such that it exerts an appropriate pressure of convergence. In the context of MO optimization, this pressure must be exerted to promote individuals in a direction that is normal as well as tangential to the tradeoff region at the same time. Intuitively, such a competitive coevolutionary approach may be sensitive to the shape of PF\* in MO optimization.

#### B. Cooperative Coevolution

Cooperative coevolution is inspired by the ecological relationship of symbiosis where different species live together in a mutually beneficial relationship. The basic idea of cooperative coevolution is to divide and conquer [50]: divide a large

system into many modules, evolve the modules separately, and then combine them together to form the whole system. The cooperative coevolutionary algorithm involves a number of independently evolving species that together form complex structures for solving difficult problems. The fitness of an individual depends on its ability to collaborate with individuals from other species, which favors the development of cooperative strategies and individuals. In addition, these techniques can be implemented at two basic levels depending on the type of modules that are evolved simultaneously [33]. In the case of single-level coevolution [7], [28], [32], [42], each evolving subpopulation represents a subcomponent of the problem to be solved. On the other hand, a two-level coevolutionary process involves simultaneous optimization of the system and modules in separate subpopulations [2], [17].

An explicit way of implementing cooperative coevolution in optimization techniques is to split a solution vector into different subcomponents and assign multiple evolving subpopulations to optimize the individual subcomponents [50]. Contrary to SO optimization, MO optimization is associated with a set of non-dominated solutions which inevitably leads to the issue of fitness assignment and representative selection. In this aspect, appropriate representatives are crucial for the search of a diverse and uniformly distributed solution set, and suitable cooperative schemes must be incorporated in order to drive the subpopulations in tandem towards the PF\*.

An early attempt to integrate the cooperative model for MO optimization is to decompose the problem along the decision space, and each subpopulation is optimized by the MO genetic algorithm (MOGA) [16]. In this MO cooperative coevolutionary genetic algorithm (MOCCGA) [32], each individual is evaluated twice in collaboration with either a random or the best representative from the other subpopulations, and the best Pareto rank is assigned as the fitness. However, the performance of MOCCGA is limited due to the lack of elitism and the localized perception of Pareto optimality.

Maneeratana *et al.* [42] later incorporate elitism in the form of a fixed size archive to store the set of nondominated solutions, and the same cooperative model is successfully extended to other MOEAs such as Niche Pareto GA [25] and NSGA [55], with significant improvements over their canonical counterparts. Like MOCCGA, however, these MOCCAs also suffer from the problem that fitness assignment conducted within a species may not be a good indicator of optimality.

Iorio and Li [28] presented a nondominated sorting cooperative coevolutionary algorithm (NSCCGA), which is essentially the coevolutionary extension of NSGAI. In NSCCGA, elitist solutions are reinserted into the subpopulations and fitness assignment takes into account the set of nondominated solutions obtained via nondominated sorting. Instead of selecting nondominated individuals with the best degree of crowding, representatives are selected randomly from the best nondominated front.

Contrary to the trend of integrating the cooperative model with well-known MOEAs, Tan *et al.* [56] proposed the cooperative-coevolution evolutionary algorithm (CCEA) based on a simple MOEA. Although the ranking scheme [16] of MOCCGA is adopted in CCEA, each individual is ranked

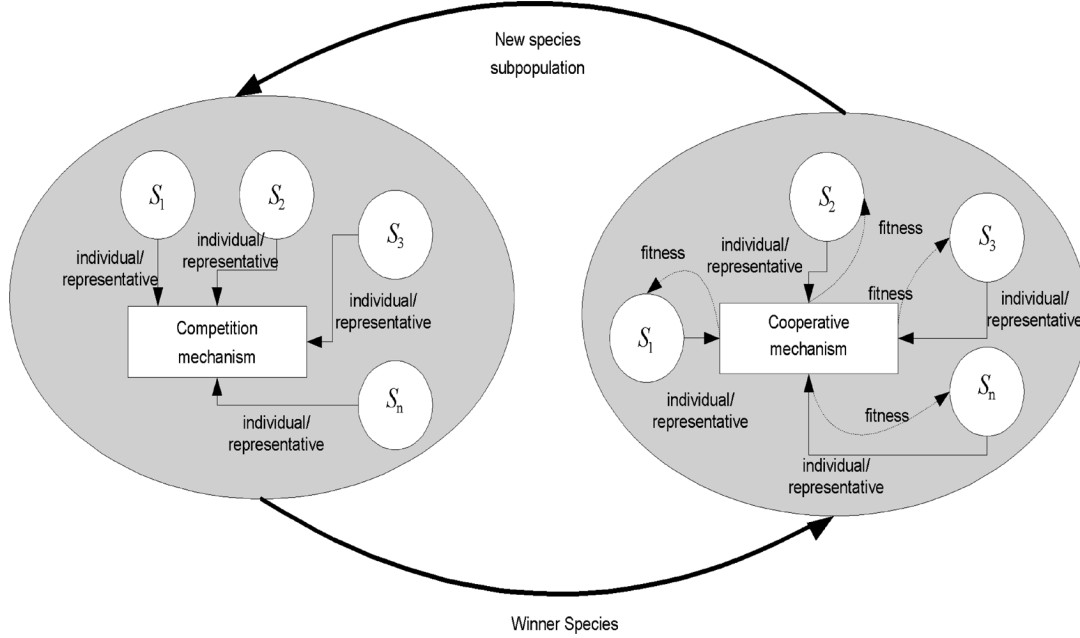


Fig. 1. Framework of competitive-cooperation coevolution.

against the nondominated solutions stored in the archive instead of within the subpopulation. In addition, an extending operator is implemented in CCEA to improve diversity and distribution of the PF by reinserting nondominated individuals with the best niche count into the evolving subpopulation. Various representative selection schemes are also examined and good performance is observed for the scheme that retains the better solution in the cooperation with two representatives from each subpopulation.

One major issue of these MOCAs is their dependence on appropriate manual decomposition of the problem into various subcomponents. Since many problems exhibit parameter interdependencies, the decomposition of solution vector and the optimization of each subcomponent independently may lead to the phenomenon of fitness landscape warping [49] and convergence to suboptimal solutions. It should be noted that parameter interactions are usually not considered explicitly in EAs. Notable exceptions include the estimation-of-distribution algorithms (EDAs) [67] that sought to learn the interrelation through joint probability distribution models and the covariance matrix adaptation evolution strategies (CMA-ES) [46], which has been recently extended to MO optimization [27].

Iorio and Li [28] also highlighted that coevolutionary algorithms are susceptible to parameter interactions, although a higher mutation rate can often improve the algorithmic performance of rotated problems. Apparently, there is an inherent tradeoff between the fine-grain search capability and the lack of diversity due to the smaller size of subpopulations in coevolutionary algorithms. The game-theoretic approach of modeling cooperation in [54] attempts to alleviate the issue of parameter dependencies by decomposing the optimization problem into only two subpopulations. Without restricting to a single computational paradigm, an interesting approach of

switching iteratively between canonical particle swarm optimization (PSO) and cooperative PSO is proposed by Van den Bergh and Engelbrecht [61] for SO optimization problems.

Applying a variant of the cooperative models discussed so far, Coello Coello and Sierra [7] proposed a coevolutionary MOEA (CO-MOEA), where different subpopulations cooperate to form the PF instead of a valid candidate solution. The CO-MOEA starts with a single evolving population and adaptively assigns different regions of the decision variable space to new subpopulations. This assignment process is performed by analyzing the contribution of each decision variable to the PF stored in adaptive grid [38]. Furthermore, the subpopulation size is changed in proportion to the discovery of new nondominated solutions, and any subpopulations without significant contributions are eliminated. Although such an approach removes design considerations like representative selection and parameter interactions, the CO-MOEA does not incorporate the fine-grain search capability of MOCAs with Potter and Jong's model.

### C. Competitive-Cooperative Coevolution

The issue of problem decomposition posts restriction on existing algorithmic designs and performances of both competitive and cooperative models. In retrospect, this problem should not arise in the context of coevolutionary algorithms, since the role that each species plays is an emergent property in nature. On the other hand, the collaboration and competition among different species are modeled independently in coevolutionary algorithms, although these two types of interactions are rarely exclusive within an ecological system. For example, there is competition even in the veneer of seemingly perfect plant-pollinator coevolution in nature [53], where different species of bees will compete for nectar and different species of flowers will compete

to attract more bees. By incorporating both elements of cooperation and competition, the proposed model represents a more holistic view of the coevolutionary forces in nature.

The proposed competitive-cooperative model involves two tightly coupled coevolutionary processes, as illustrated in Fig. 1. Similar to conventional cooperative coevolutionary algorithms, individuals from different species collaborate to solve the problem during the cooperative process. Each subpopulation evolves in isolation and there is no restriction on the form of representation or on the underlying EA. On the other hand, the cooperative species will also collaborate with other subpopulations for the right to represent the various subcomponents of the problem.

The interaction between the cooperative and competitive processes may take place iteratively after each generation or at a frequency determined by the user. For the ensuing discussions, we consider that the problem at hand is decomposed along the decision variables. Each decision variable may be assigned to a number of subpopulations, and a subpopulation may optimize for more than one decision variable.

1) *Credit Assignment*: The credit assignment for the competitive and cooperative processes is performed at the subpopulation and individual level, respectively. In the cooperative process, the different objectives are evaluated by assembling each individual with representatives of the other species to form a valid candidate solution. Accordingly, appropriate fitness assignment such as Pareto ranking can be performed for that particular individual. In the competitive process, the fitness of a particular species is computed by estimating how well it performs relative to its competitors in cooperating with other species to produce good solutions. For example, the species selected out of  $N$  competing subpopulations is given a higher probability for representing a particular variable in subsequent generations, while the losing species of the competition is penalized and given a lower probability.

2) *Problem Decomposition and Subcomponent Interdependency*: As mentioned earlier in the section, the issue of problem decomposition needs to be addressed for coevolutionary algorithms. The difficulty lies in the fact that information pertinent to the number or role of subcomponents is usually unknown *a priori*, and many problems can only be decomposed into subcomponents with complex interdependencies. The competitive-cooperation coevolutionary model addresses such an issue through emergent problem decomposition.

As illustrated above, the competitive process leads to a potential “arms race” among the cooperative species to improve their contributions in the associated subcomponents. It should be noted that the collaboration between these two coevolutionary models can lead to the natural formation of competitive subpopulations rather than subcomponents. In addition, it facilitates the interactions among different species, in possibly various roles, right at the onset of the optimization process which benefits the discovery of interdependencies among the species. Therefore, the interplay of competition and cooperation provides an environment in which interdependent subcomponents end up within the similar species, and reasonable problem decomposition emerges due to evolutionary pressure rather than being specified by the user.

TABLE I  
NOMENCLATURE

Notation	Definition
$A$	Archive of nondominated solutions
$A^T$	Temporal archive of past nondominated solutions
$a_i$	The $i$ -th nondominated solution of $A$
$C_{freq}$	Frequency of competition
$n_x$	Number of decision variables
$P_i^c$	Competition pool for the $i$ -th variable
$R_{size}$	Number of $a_i$ to be updated to $A^T$
$S_i$	The $i$ -th subpopulation
$s_{i,j}$	The $j$ -th individual of $S_i$
$s_{i,rep}$	Representative of $S_i$
$SC_{ratio}$	Ratio of stochastic competitors in $P_i^c$

The emergent attribute of the competitive-cooperation coevolutionary model is distinctively different from the cooperative model proposed by Potter and Jong [50]. Although the participation of a subpopulation is based on its contribution made to the collaboration among species in both approaches, this feature is due to the emergence of fitter species for a particular problem subcomponent in the proposed model. One limitation of the approach in [50] is that stagnant subpopulations are simply replaced by randomly initialized subpopulations, implying that any useful information obtained previously can be discarded.

3) *Diversity*: The competitive-cooperation coevolutionary model provides a means of exploiting the complementary diversity preservation mechanism of both competitive and cooperative models. In the cooperative model, the evolution of isolated species tends to produce more diversified individuals across the different subpopulations, although this property does not necessarily extend to within each subpopulation. On the other hand, a diverse subpopulation is driven by the necessity to deal with different situations posed by the other subpopulations in the competitive model. Furthermore, the competitive process in the competitive-cooperation coevolutionary model also allows for a more diversified search since the optimization of each subcomponent is no longer restricted to one species. The competing species provides another round of optimization for each subcomponent, thus increases the extent of the search and maintains an overall low computation requirement.

#### IV. APPLYING COMPETITIVE-COOPERATION COEVOLUTION FOR MO OPTIMIZATION

Based on the competitive-cooperation coevolutionary paradigm described in Section III, this section presents a COEA for MO optimization. The list of notations used in subsequent sections are summarized in Table I. The mechanism of cooperative coevolution is described in Section IV-A, while the competitive element of the proposed paradigm is presented in Section IV-B. Finally, the implementation details of COEA are given in Section IV-C.

---

**Cooperative Process**

---

$S_i$ : The  $i$ -th subpopulation  
 $s_{i,j}$ : The  $j$ -th individual of  $S_i$   
 $A$ : Archive of nondominated solutions

**for**  $j=1$  to  $|S_i|$  **do**

- Assemble complete solution with  $s_{i,j}$  and representatives from the other subpopulations
- Evaluate solution
- Update  $A$

**end**

**for**  $j=1$  to  $|S_i|$  **do**

- Assign Pareto rank to  $s_{i,j}$
- Calculate niche count of  $s_{i,j}$

**end**

Update representative of  $S_i$

---

Fig. 2. Pseudocode of the cooperative coevolutionary mechanism.

### A. Cooperative Mechanism

The cooperative mechanism of the proposed COEA is extended from the model introduced by Tan *et al.* [56]. By adopting this strategy, the algorithm can exploit the fine-grained search capability desirable in many applications and maintain good diversity across the subpopulations.

The pseudocode of the cooperative mechanism is shown in Fig. 2. At the start of the optimization process, the  $i$ th subpopulation is initialized to represent the  $i$ th variable. Concatenation between individuals in  $S_i$  and representatives from the other subpopulations is necessary to form a valid candidate solution for evaluation. As an example, consider a three-decision variable problem, where subpopulations  $S_1$ ,  $S_2$ , and  $S_3$ , represent the variables,  $x_1$ ,  $x_2$ , and  $x_3$ , respectively. When assessing the fitness of  $s_{1,j}$ , it will combine with the representatives of  $S_2$  and  $S_3$  to form a valid candidate solution.

In this approach, archive updating is conducted after the evaluation of each individual. Pareto ranking and niche count computation of individual,  $s_{i,j}$ , are then conducted with respect to the archive. Note that only fitness of the individuals from  $S_i$  is updated at the  $i$ th cycle. The Pareto rank of each individual is based on the number of archived solutions dominating it, i.e.,

$$\text{rank}(s_{i,j}) = 1 + |\{a_k \in A | a_k \prec s_{i,j}\}|. \quad (2)$$

Similar to the ranking process, the niche count ( $nc$ ) of each individual is calculated with respect to the archive of nondominated solutions. The dynamic sharing scheme proposed in [58] is employed here to estimate the sharing radius.

The cooperative process is carried out in turn for all  $n_x$  subpopulations where  $n_x$  is the number of decision variables. Before proceeding to the evaluation of next subpopulation, the representative of  $S_i$  denoted as  $s_{i,rep}$  is updated to improve the speed of convergence. This updating process is based on a partial order such that ranks will be considered first, and followed by niche count if there is a tie in the rank. For any two individuals,  $s_{i,j}$  and  $s_{i,k}$ ,  $s_{i,j}$  is selected over  $s_{i,k}$  if  $\text{rank}(s_{i,j}) < \text{rank}(s_{i,k})$  or  $\{\text{rank}(s_{i,j}) = \text{rank}(s_{i,k})$  and

---

**Competitive Process**

---

$P_i^c$  : Competition pool for variable  $i$

**for**  $i=1$  to  $|S_i|$  **do**

- Insert representative of subpopulation representing variable  $i$ ,  $s_{i,rep}$  into  $P_i^c$

**if**  $n_x > |S_i|$  **do**

- Select competing subpopulations randomly
- Insert competitors from selected subpopulations into  $P_i^c$

**elseif**  $n_x \leq |S_i|$  **do**

- Insert competitors from other subpopulations into  $P_i^c$
- Insert random individuals from  $S_i$  into  $P_i^c$

**end**

- Cooperative process
- Determine winning subpopulation  $S_k$
- Update  $S_i = S_k$

**end**

---

Fig. 3. Pseudocode of the competitive coevolutionary mechanism.

$nc(s_{i,j}) < nc(s_{i,k})\}$ . The rationale of selecting a nondominated representative with the lowest niche count is to promote diversity of the solutions via the approach of cooperation among multiple subpopulations.

### B. Competitive Mechanism

Given that the cooperative scheme optimizes a single variable in each subpopulation, one simple approach is to allow the different subpopulations to take up the role of a particular problem subcomponent in a round-robin fashion. The most competitive subpopulation is then determined and the subcomponent will be optimized by the winning species in the next cooperative process. Ideally, the competition is performed such that all individuals from a particular subpopulation compete with all other individuals from the other subpopulations in order to determine the extent of its suitability. However, such an exhaustive approach requires extensive computational effort that is often practically infeasible. A more practical approach is thus to conduct competition with only selected individuals among a certain number of competitor subpopulations to estimate the species fitness and suitability.

The pseudocode of the competitive mechanism is shown in Fig. 3. The competitive process to discover the most suitable subpopulation is performed for each variable in an iterative manner. For the  $i$ th variable, the representative of the associated subpopulation, i.e.,  $s_{i,rep}$ , is selected along with the competitors from the other subpopulations to form a competition pool. The COEA adopts a simple competitor selection scheme of choosing a random individual from each competing subpopulation. Intuitively, the selection of a random competitor will enable the COEA to explore relationships among the different variables. Other types of competition schemes will be presented and analyzed in Section VI-E. In the case where  $n_x > |S_i|$ , i.e.,

the number of subpopulations is larger than the subpopulation size, the participating subpopulations will be selected randomly before the start of the competition process. This provides an opportunity for the other subpopulations left out in this instance to participate in future competitions.

These competitors will then compete via the cooperative mechanism described earlier to determine the extent of the cooperation achieved with representative of the other subpopulations. In this approach, the winning species is determined by checking the originating subpopulation of the representative after the representative update. At the end of the competitive process,  $S_i$  will remain unchanged if its representative wins the competition. In the case that a winner emerges from other subpopulations,  $S_i$  will be replaced by the individuals from the winning subpopulation. The rationale of replacing the losing subpopulation instead of associating the winning subpopulation with the decision variable directly is that different variables may have close but not identical properties. Therefore, it is more appropriate to seed the losing subpopulation with the desirable information and to allow it to evolve independently.

By embedding the competitive mechanism within the cooperative process, the adaptation of problem decomposition and the optimization process are conducted simultaneously. Hence, no additional computation cost is incurred from the competition. Moreover, this approach also has the advantage of allowing different subpopulations to solve a single component as a collective unit, with the competitors acting as a source of diversity.

### C. Implementation

The flowchart of the proposed COEA is illustrated in Fig. 4. The initialization process involves the creation of  $n_x$  subpopulations of random individuals, where the  $i$ th subpopulation will represent the  $i$ th decision variable. The individuals will then undergo the competitive-cooperation process until the stopping criteria is satisfied, which can be set based upon a fixed number of function evaluations. In this work, the number of fitness function evaluations is determined according to past experience and complexity of the test functions, which can be in multiples of the number of decision variables.

At each generation, either the cooperative or competitive mechanism is activated. In particular, the competitive mechanism is applied at a fixed frequency of  $C_{\text{freq}} = 10$ ; and the cooperative process is adopted otherwise. During the cooperative process, individuals of each subpopulation are evaluated by combining them with representatives from the other subpopulations to form a complete candidate solution. As mentioned in Section IV-A, the archive will be updated after each evaluation. After all individuals in the subpopulations have been evaluated in the cooperative process, binary tournament selection based on Pareto rank is applied to select the parents for each of the subpopulations. Here, the individual with a lower niche count will be selected in the case of a tie in rank. The selected parents will then undergo the process of uniform crossover and bit-flip mutation.

During the competitive process, the archiving is also performed after each evaluation, as described in Section IV-B. In contrast to the cooperative process, tournament selection is not

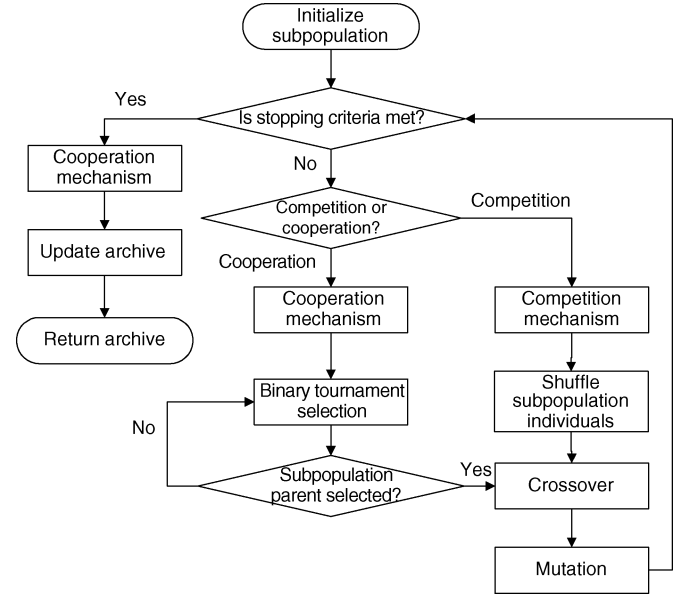


Fig. 4. Flowchart of COEA.

employed here for the selection of parents, and the subpopulation individuals are shuffled randomly before undergoing crossover and mutation. It is not necessary to perform selection based on fitness measure since the replacement individuals have not been evaluated for their fitness and may not perform in an identical manner in their new role of optimizing another subcomponent.

The algorithm applies a fixed-size archive to store nondominated individuals along the evolution. A complete candidate solution formed by the subpopulations will be added to the archive if it is not dominated by any archived solutions. Likewise, any archive members dominated by this candidate solution will be removed. When the predetermined archive size is reached, a recurrent truncation process [35] based on niche count is used to eliminate the most crowded archive member.

## V. ADAPTING COEA FOR DYNAMIC MO OPTIMIZATION

Besides considering the different requirements of MO problems, the issues of diversity and outdated archived solutions should also be addressed before the proposed COEA is capable of dealing with environmental variations in dynamic optimization. Section V-A describes a scheme for achieving good diversity to be introduced, while exploiting useful past information. Section V-B describes a simple temporal memory approach, which stores and reintroduces outdated nondominated individuals into the archive when necessary.

### A. Introducing Diversity Via Stochastic Competitors

Generally, the population diversity desirable for tracking the dynamic  $PS_t^*$  in COEA can either be introduced explicitly through mechanisms such as random restart and hypermutation or be maintained by means of niching and other diversity preservation schemes. The approach of using multiple populations to explore the different regions of the search space is not applicable here, since the application of subpopulations



in COEA serves another purpose of optimizing a specific subcomponent of the problem. Although explicit generation of diversity will allow the algorithm to react faster to severe environmental changes, such an approach is unable to utilize useful past information. On the other hand, the potential for information exploitation in diversity preservation schemes is often achieved at the expense of a slower convergence. This is known as the exploration–exploitation dilemma for dynamic optimization [4].

A diversity scheme which exploits the competitive mechanism of COEA is thus implemented. In every generation, a fixed number of archived solutions are reevaluated and the current objective values are checked against the previous values for discrepancies. If there is any environmental variation in the evolution, the competitive mechanism will be started, in addition to its fixed schedule. This strategy allows the algorithm to assess the potential of existing information within the various subpopulations for exploitation in the new problem landscape.

Furthermore, the competitive process provides a natural conduit in which the introduction of diversity into the subpopulations can be regulated. Instead of reinitialization or subjecting the entire subpopulation to hypermutation, a set of stochastic competitors are introduced together with the competitors from the other subpopulations, where the ratio between the two types of competitors is given by the parameter  $SC_{ratio}$ . The idea is to compare the potential of new regions in the search space and the past information to decide whether the subpopulation should be initialized. The latin hypercube sampling is applied to generate individuals along each dimension uniformly. In the case that stochastic competitor emerges as the winner, the particular subpopulation is reinitialized in the region that the winner is sampled from. Hence, diversity is introduced into the subpopulations only when it presents an advantage over the current information at hand.

### B. Handling Outdated Archived Solutions

If there is any environmental change in the evolution, it is likely that the archived solutions will not remain nondominated, and these outdated archived solutions will keep out the nondominated solutions if they are being left unchecked. Therefore, appropriate measures must be incorporated to minimize the detrimental effects of any outdated archived solutions. A simple approach is to reevaluate all the outdated solutions and to remove only the dominated solutions from the archive. Since most MOEAs are elitist in general, such an approach may mislead the optimization process with nondominated but outdated archived solutions. Moreover, the process of reevaluation will result in extra computational cost, which is undesirable. An alternative approach is to simply discard all the archived solutions. But useful information about past  $PF_t$  cannot be exploited in the case where  $PS_t^*$  is cyclic in nature.

In order to store the potentially useful information in dCOEA, an additional external population denoted as the temporal memory is used in conjunction with the archive. In the ideal situation, the temporal memory is a repository of all the nondominated solutions prior to any environmental variation. Due to the limited computational resources, however, decision must be made on what solutions and how the solutions are

Temporal Archive Update	
$P^T$ : Temporary pool for best archived solutions in $A$	
Select and remove best archived solution along each dimension from $A$ to $P^T$	
if	$R_{size} \geq M$ do
▪	Add $P^T$ to $A_T$
for	$i = M$ to $R_{size}$
▪	Randomly select and remove $a_k \in A$
▪	Add $a_k$ to $A_T$
end	
else	
for	$i = 1$ to $R_{size}$
▪	Randomly select and remove $a_k \in P^T$
▪	Add $a_k$ to $A_T$
end	
end	
if	$ A_T  > \text{Archive limit}$
▪	Remove oldest member from $A_T$
end	

Fig. 5. Pseudocode of the temporal archival mechanism.

stored in the temporal memory. The pseudocode of the temporal archive updating mechanism is shown in Fig. 5. To store the outdated solutions, a fixed number  $R_{size}$  of the archive is added to the temporal memory upon a landscape change. When the upper bound of the temporal memory is reached, the oldest set of  $R_{size}$  outdated solutions is removed for newer solutions. To select the  $R_{size}$  outdated solutions, the dCOEA stores the extreme solutions along each dimension in the objective space. In the case where  $R_{size}$  is greater than the number of extreme solutions, the rest of the solutions to be stored are randomly selected from the archive. On the other hand, if  $R_{size}$  is smaller than the number of extreme solutions, then  $R_{size}$  extreme solutions will be randomly selected into the temporal memory. Intuitively, the value of  $R_{size}$  controls the tradeoff between the storage of information across different environmental changes and the information for a particular instance of landscape change. In particular, a smaller value of  $R_{size}$  will allow for a more diverse range of past solutions.

After the  $R_{size}$  outdated archived individuals have been added to the temporal memory, all archived solutions will be discarded. Subsequently, the temporal memory will be reevaluated and the archive updating is conducted on this external population. The computational cost incurred by this reevaluation process is necessary so as to exploit any useful information about the current  $PS_t^*$ . To address the concern that solutions updated into the archive through this approach may misguide the optimization process, no archived solutions will be reinserted back to the subpopulations at the generation immediately after the environmental change.

## VI. EMPIRICAL STUDY FOR STATIC ENVIRONMENT

This section starts with a description of various benchmark problems and performance metrics in Sections VI-A and VI-B, respectively. Subsequently, a comparative study between COEA and various MOEAs that are representatives of the state-of-the-

TABLE II  
DEFINITION OF STATIC TEST FUNCTIONS

Test function	Definition
1 FON	$f_1(x_1, \dots, x_8) = 1 - \exp[-\sum_{i=1}^8 (x_i - \frac{1}{\sqrt{8}})^2],$ $f_2(x_1, \dots, x_8) = 1 + \exp[-\sum_{i=1}^8 (x_i - \frac{1}{\sqrt{8}})^2],$ where $-2 \leq x_i < 2, \forall i = 1, 2, \dots, 8$
2 KUR	$f_1(x_2, x_3) = \sum_{i=1}^2 [-10 \exp(-0.2 \sqrt{x_i^2 + x_{i+1}^2})],$ $f_1(x_2, x_3) = \sum_{i=1}^3 [ x_i ^{0.8} + 5 \cdot \sin(x_i^3)]$ $x_i \in [-5, 5]$
3 DTLZ3	$f_1(\vec{x}) = (1 + g(\vec{x}_M)) \cdot \cos(0.5\pi x_1) \cdots \cos(0.5\pi x_{M-1}),$ $f_2(\vec{x}) = (1 + g(\vec{x}_M)) \cdot \cos(0.5\pi x_1) \cdots \sin(0.5\pi x_{M-1})$ $\vdots$ $f_M(\vec{x}) = (1 + g(\vec{x}_M)) \cdot \sin(0.5\pi x_1)$ $g(\vec{x}_M) = 100 \left\{  \vec{x}_M  + \sum_{x_i \in \vec{x}_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right\}$ where $M = 5, \vec{x}_M = \{x_M, \dots, x_{M+9}\}, x_i \in [0, 1]$

TABLE III  
PARAMETER SETTING FOR DIFFERENT ALGORITHMS

Parameter	Settings
Populations	Population size 100 in NSGAI and SPEA2; Subpopulation size 10 in COEA and CCEA; Archive (or secondary population) size 100.
Chromosome	Binary coding; 30 bits per decision variable.
Selection	Binary tournament selection
Crossover operator	Uniform crossover
Crossover rate	0.8
Mutation operator	Bit-flip mutation
Mutation rate	$\frac{1}{L}$ for DTLZ3 where $L$ is the chromosome length; $\frac{1}{B}$ for FON and KUR where $B$ is the bit size per decision variable;
Niche Radius	Dynamic sharing [58].

arts will be conducted in Section VI-C. Further analysis on the dynamics of competitive-cooperation evolution are presented in Sections VI-D and VI-E.

#### A. Static MO Test Functions

Three benchmark problems, FON, KUR, and DTLZ3 are used to examine the algorithmic performance of multiobjective optimization techniques. These test functions have different problem characteristics [9], such as multimodality, convexity, discontinuity, and nonuniformity, which may challenge the MOEA's ability to converge and maintain population diversity. Recent works [26], [34] showed that the performances of MOEAs often do not scale well with respect to the number of objectives. The DTLZ3 is thus formulated here as a five-objective optimization problem. Many researchers, such as [8], [11], [57], [59], and [70] have applied these test problems to examine their proposed algorithms in static environments. The definition of these test functions is summarized in Table II.

#### B. Static Performance Metrics

The performance metrics reflect the quality of the scrutinized solution set with respect to some measures. Many comparative studies [11], [29], [57], [63] made use of a suite of unary performance metrics [68] pertinent to the MO optimization goals of proximity, diversity, and distribution. Knowles and Corne [37], and Zitzler *et al.* [68] have discussed the suitability and limitation of various performance metrics. Four metrics including three different unary metrics and a proposed  $n$ -ary domination ratio are adopted in this paper.

1) *Proximity Indicator*: The metric of generational distance (GD) gives a good indication of the gap between the PF\* and the evolved PF. Mathematically, the metric is a function of individual distance given as

$$GD = \frac{1}{n_{PF}} \cdot \left( n_{PF} \sum_{i=1}^{n_{PF}} d_i^2 \right)^{\frac{1}{2}} \quad (3)$$

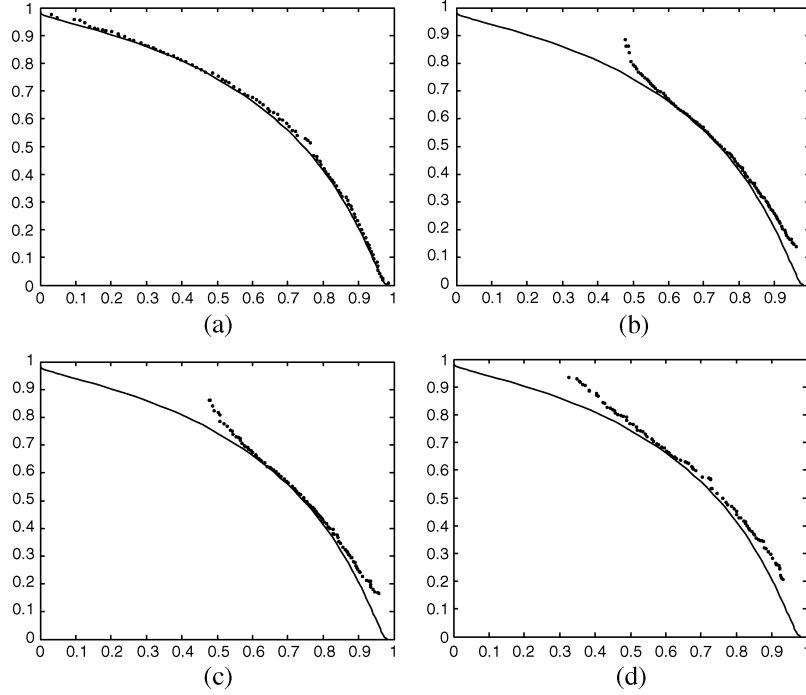


Fig. 6. The evolved Pareto front from (a) COEA, (b) CCEA, (c) NSGAII, and (d) SPEA2 for FON.

where  $n_{PF} = |PF|$ ,  $d_i$  is the Euclidean distance (in objective space) between the  $i$ th member of PF and the nearest member of  $PF^*$ . A low value of GD is desirable, which reflects a small deviation between the evolved and the true Pareto front.

2) *Diversity Indicator*: A modified maximum spread  $MS'$  [19] is applied to measure how well the  $PF^*$  is covered by the PF. Specifically, the modified metric takes into account the proximity to  $PF^*$ , e.g., a higher value of  $MS'$  reflects that a larger area of the  $PF^*$  is covered by the PF. The metric is given as

$$MS' = \sqrt{\frac{1}{M} \sum_{i=1}^M \left[ \frac{\min[\overline{PF}_i, \overline{PF}_i^*] - \max[\underline{PF}_i, \underline{PF}_i^*]}{\overline{PF}_i^* - \underline{PF}_i^*} \right]^2} \quad (4)$$

where  $\overline{PF}_i$  and  $\underline{PF}_i$  is the maximum and minimum of the  $i$ th objective in PF, respectively;  $\overline{PF}_i^*$  and  $\underline{PF}_i^*$  is the maximum and minimum of the  $i$ th objective in  $PF^*$ , respectively.

3) *Distribution Indicator*: The metric of spacing [52] shows how evenly the nondominated solutions are distributed along the discovered Pareto front. It is given as

$$S = \frac{1}{\bar{d}'} \cdot \left( \frac{1}{n_{PF}} \cdot \sum_{i=1}^{n_{PF}} (d'_i - \bar{d}')^2 \right)^{\frac{1}{2}}$$

$$\bar{d}' = \frac{1}{n_{PF}} \sum_{i=1}^{n_{PF}} d'_i \quad (5)$$

where  $n_{PF} = |PF|$ ,  $d'_i$  is the Euclidean distance (in objective space) between the  $i$ th member and its nearest member in PF.

4) *Pareto Dominance Indicator*: In [68], Zitzler *et al.* showed that no combinations of unary performance metrics can provide a clear indication of whether an evolved set is better than another in the Pareto dominance sense. Therefore, an  $n$ -ary Pareto

dominance indicator is proposed here as a complement to the above metrics. Considering the different PFs,  $A_1, A_2, \dots, A_n$  evolved by  $n$  algorithms, this metric measures the ratio of non-dominated solutions that is contributed by a particular solution set  $A_i$  to the nondominated solutions provided by all solution sets. Mathematically, the nondominance ratio (NR) is given as

$$NR(A_1, A_2, \dots, A_n) = \frac{|B \cap A_1|}{|B|}$$

$$B = \{b_i | \forall b_i \exists a_j \in (A_1 \cup A_2 \dots \cup A_n) \prec b_i\} \quad (6)$$

where  $A_1$  is the solution set under evaluation.

### C. Comparative Study of COEA

In order to examine the effectiveness of COEA, a comparative study including COEA, CCEA [56], SPEA2 [69], and NSGAII [11] is carried out based upon the three benchmark problems listed in Table II. The simulations are implemented in C++ on an Intel Pentium 4 2.8 GHz personal computer. Thirty independent runs are performed for each of the test functions to obtain the statistical information, such as consistency and robustness of the algorithms. The various parameter settings are listed in Table III. All the algorithms here are implemented using the same binary coding scheme, tournament selection, uniform crossover, and bit flip mutation.

1) *FON*: The FON challenges the algorithm's ability to find and maintain the entire tradeoff curve uniformly. Since the tradeoff curve is nonconvex and nonlinear in FON, it is difficult to maintain a stable evolving population for this problem. A stopping criterion of 20 000 evaluations is used here. The PFs obtained from the different algorithms using the same random seed are showed in Fig. 6(a)–(d), while the distributions of the

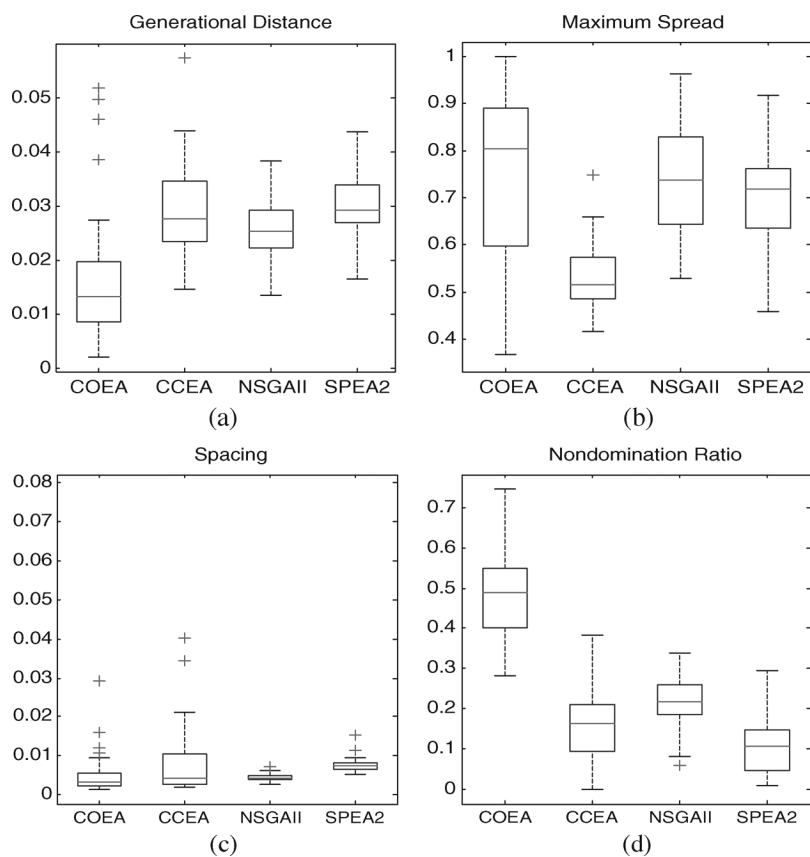


Fig. 7. Performance metric of (a) GD, (b) MS, (c) S, and (d) NR for FON.

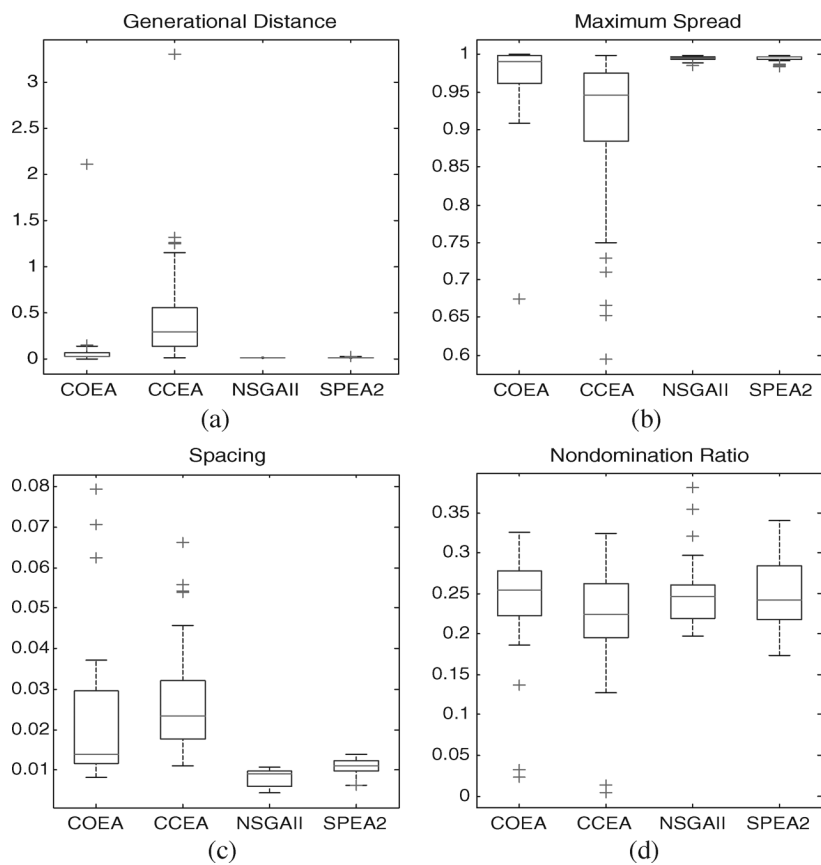


Fig. 8. Performance metric of (a) GD, (b) MS, (c) S, and (d) NR for KUR.

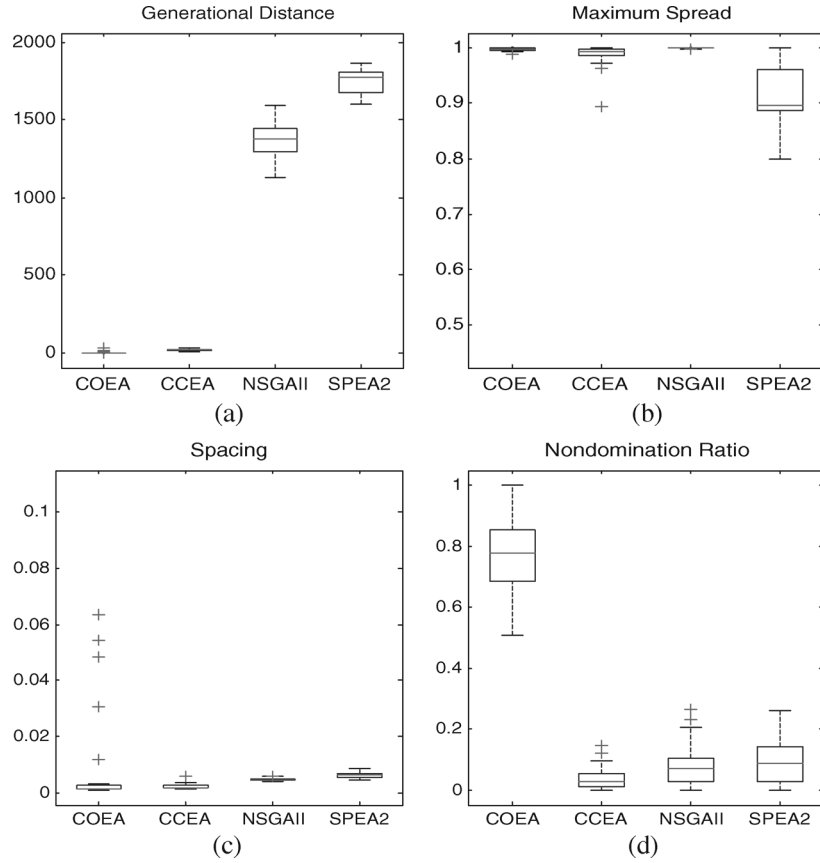


Fig. 9. Performance metric of (a) GD, (b) MS, (c) S, and (d) NR for DTLZ3.

different performance metrics are represented by box plots in Fig. 7(a)–(d). The advantage of the proposed competitive-cooperation model in handling parameter interactions is shown in Fig. 7 and by comparing the evolved PFs in Fig. 6(a) and (b).

2) *KUR*: The KUR is characterized by an  $PF^*$  that is non-convex and disconnected, which contains three distinct and disconnected regions on the final tradeoff. The decision variables corresponding to the final tradeoff for KUR are difficult to find, since they are disconnected in the decision variable space. Like FON, there are high interactions between the decision variables that pose a challenge to MOEAs. A stopping criterion of 30 000 evaluations is used for this problem. The distributions of the different performance metrics are represented by box plots in Fig. 8(a)–(d). The main difficulty stemming from the high parameter interactions in this problem is the finding of all the four disconnected regions of PF. Although CCEA is capable of evolving a PF that is close to  $PF^*$ , it can be observed from Fig. 8(b) and (c) that it faces difficulty in finding a diverse PF. As shown in the metric of MS, the competitive-cooperation paradigm allows COEA to evolve a more diverse solution set as compared with the CCEA.

3) *DTLZ3*: DTLZ3 is used to challenge the MOEA's capability to produce adequate pressure in driving individuals towards the high-dimensional  $PF^*$ . Moreover, the DTLZ3 is also characterized by the challenge of multimodality. A stopping criterion of 28 000 evaluations is used for this problem. The distributions of the different performance metrics for DTLZ3 are

shown in Fig. 9(a)–(d). It can be observed that although SPEA2 and NSGAII are unable to find good solutions near the  $PF^*$ , they manage to evolve a good spread of solutions. On the other hand, the COEA is seen to scale well with increasing objectives and to produce competitive performance for GD, S, and MS. The metric of NR also shows that the COEA outperforms CCEA, as given in Fig. 9(d).

#### D. Effects of the Competitive Mechanism

In this section, experiments are conducted at  $C_{freq} = \{1, 5, 10, 30, 50, \text{inf}\}$  to study the effects and dynamics of incorporating both competitive and cooperative processes in a common framework based on the benchmark problems of FON, KUR, and DTLZ3. As mentioned earlier, the FON and KUR have severe parameter interactions, which are useful to examine the performance improvement of competitive mechanism in COEA. The DTLZ3 is included here since most algorithms are unable to deal with this problem effectively as observed in previous section.

The performances of COEA with  $C_{freq} = \{1, 5, 10, 30, 50, \text{inf}\}$  for FON, KUR, and DTLZ3 are summarized in Tables IV–VI, respectively. Note that no competition takes place when  $C_{freq} = \text{inf}$ , which effectively reduces the competitive-cooperative paradigm to a conventional cooperative model. From the tables, it can be observed that COEA gives better performances for the three benchmark problems at lower settings of  $C_{freq}$ , and the performances

TABLE IV  
PERFORMANCE OF COEA FOR FON WITH DIFFERENT  $C_{\text{freq}}$ .  
THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

		1	5	10	30	50	Inf
GD	1st quartile	0.0080	<b>0.0050</b>	0.0086	0.0107	0.0119	0.0235
	Median	0.0116	<b>0.0075</b>	0.0133	0.0157	0.0207	0.0276
	3rd quartile	0.0171	<b>0.0090</b>	0.0198	0.0217	0.0243	0.0347
MS	1st quartile	<b>0.9492</b>	0.5394	0.5991	0.6313	0.6121	0.4857
	Median	<b>0.9741</b>	0.8916	0.8036	0.7510	0.6882	0.5159
	3rd quartile	<b>0.9975</b>	0.9466	0.8891	0.8547	0.7280	0.5732

TABLE V  
PERFORMANCE OF COEA FOR KUR WITH DIFFERENT  $C_{\text{freq}}$ .  
THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

		1	5	10	30	50	Inf
GD	1st quartile	0.0349	<b>0.0256</b>	0.0329	0.0370	0.0521	0.1414
	Median	0.0425	<b>0.0365</b>	0.0376	0.0864	0.2946	0.2941
	3rd quartile	<b>0.0499</b>	0.0549	0.0807	0.3078	0.4924	0.5592
MS	1st quartile	<b>0.9995</b>	0.9822	0.9608	0.9458	0.9214	0.8841
	Median	<b>0.9998</b>	0.9939	0.9902	0.9678	0.9610	0.9461
	3rd quartile	<b>1.0000</b>	0.9988	0.9987	0.9906	0.9730	0.9752

TABLE VI  
PERFORMANCE OF COEA FOR DTLZ3 WITH DIFFERENT  $C_{\text{freq}}$ .  
THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

		1	5	10	30	50	Inf
GD	1st quartile	28.6021	0.0000	0.0000	0.0000	0.0000	15.0409
	Median	58.4115	0.0039	0.0009	0.0000	0.0000	18.4015
	3rd quartile	100.8232	0.0252	<b>0.0248</b>	0.0271	0.1414	23.4576
MS	1st quartile	0.6744	<b>0.9990</b>	0.9972	0.9950	0.9958	0.9860
	Median	0.7575	<b>0.9998</b>	0.9990	0.9987	0.9979	0.9933
	3rd quartile	0.8702	<b>1.0000</b>	0.9998	0.9996	0.9995	0.9986

deteriorate when the competitive mechanism is absent. By comparing the results over different  $C_{\text{freq}}$ , it is clear that a larger  $C_{\text{freq}}$  allows the COEA to adapt faster to the problem requirements and to evolve a more diverse and near optimal PF. On the other hand, it can be seen that the improvement for MS is attained at the expense of GD for FON and KUR. In the case of DTLZ3, the algorithmic performance deteriorates sharply at  $C_{\text{freq}} = 1$ , since constant competition may restrict the time necessary for the subpopulations to adapt to the decision variables. Nonetheless, it can be observed that the incorporation of competitive mechanism with reasonable  $C_{\text{freq}}$  can result in significant improvement of convergence and diversity for the problems of FON, KUR, and DTLZ3.

Fig. 10 shows the evolutionary trend of the best solution for each variable in DTLZ3 with  $C_{\text{freq}} = 10$  and  $C_{\text{freq}} = 50$ . In order to evolve a near-optimal, diverse, and uniformly distributed PF, the algorithm needs to maintain a wide range of

values for  $x_1 - x_4$ , while finding the optimal value of 0.5 for  $x_5 - x_{14}$ . For both settings, it can be seen that  $x_1 - x_4$  oscillate continuously along the evolution process in order to span the entire range of feasible values. Likewise,  $x_5 - x_{14}$  are able to converge to the optimal value of 0.5, as shown in Fig. 10(c)–(d). By comparing Fig. 10(a) and (b), it can be seen that the COEA with  $C_{\text{freq}} = 10$  converges to the optimal value of 0.5 at the tenth generation. However, the COEA with  $C_{\text{freq}} = 50$  only converges to the optimal value at the 50th generation. It is also observed that the convergence of the algorithm coincides with each competition process in the evolution.

To analyze the influence of the competitive mechanism on the emergent decomposition process, the winning subpopulation for each round of the competition is shown in Fig. 11. To facilitate the introduction of diversity for variables  $x_1 - x_{14}$ , it is observed that  $S_1 - S_3$  emerged as the most suitable subpopulations and each takes over the role of optimizing a variable within  $x_1 - x_4$

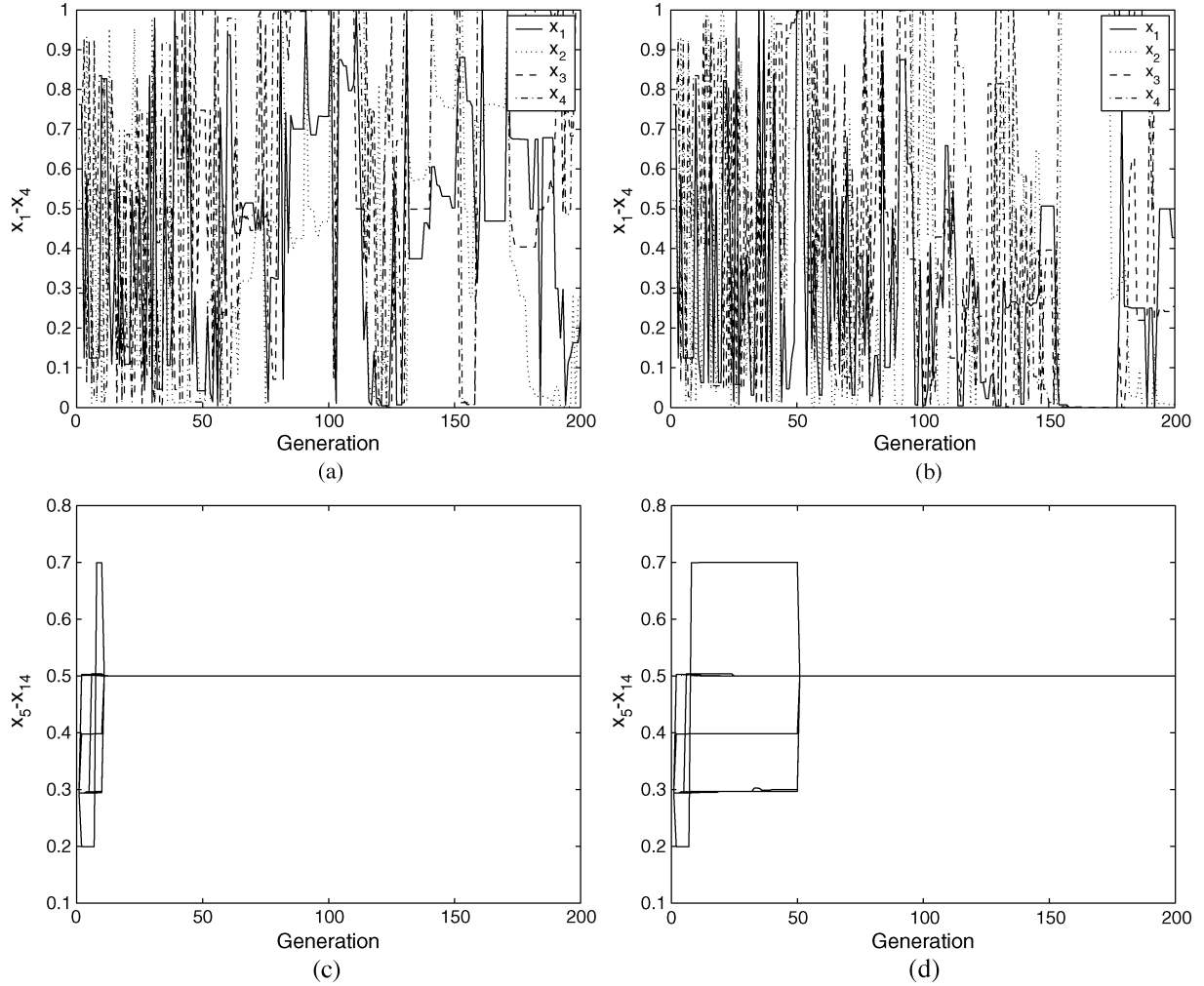


Fig. 10. Evolutionary trend of variables (a)  $x_1 - x_4$  for  $C_{\text{freq}} = 10$ , (b)  $x_1 - x_4$  for  $C_{\text{freq}} = 50$ , (c)  $x_5 - x_{14}$  for  $C_{\text{freq}} = 10$ , and (d)  $x_5 - x_{14}$  for  $C_{\text{freq}} = 50$ .  $x_5 - x_{14}$  are all represented by straight lines.

in an almost iterative manner. For the variables  $x_5 - x_{14}$ , it is observed that  $S_8$  took over the rest of the subpopulations at the first competition. Although subsequent winners include  $S_4$ ,  $S_7$ ,  $S_8$ ,  $S_9$  and  $S_{10}$ ,  $S_9$  is shown to be the dominant subpopulation for these variables. It can also be observed from the subpopulation distribution that individuals of  $S_1 - S_3$  are distributed throughout the search space, while individuals of  $S_4 - S_{14}$  are concentrated around the value of 0.5.

#### E. Effects of Different Competition Schemes

In this section, three different competition models are incorporated in COEA and their effectiveness for MO optimization are investigated. These models are as follows.

- **Random:** Before the start of each competition process, an individual is selected randomly from each competing subpopulation as the participant. This set of competitors will remain fixed during the entire course of the competition for that particular subcomponent. This scheme is implemented in the COEA adopted for the comparative study in previous section.
- **Elitist:** Before the start of each competition process, each competing subpopulation selects the best individual for

its associated subcomponents as the participant. This set of competitors will remain fixed during the whole course of the competition for that particular subcomponent. This scheme is expected to perform well when the different subcomponents have similar properties.

- **Hybrid:** Before the start of each competition process, each competing subpopulation randomly select either the best individual or a random individual as the participant. This set of competitors will remain fixed during the entire course of the competition for that particular subcomponent. This model represents a tradeoff between the random and the elitist scheme.

The experiments are conducted for COEA having different competition schemes with  $C_{\text{freq}} = 10$ . The results of 30 independent runs for the problems of FON, KUR, and DTLZ3 are summarized in Tables VII–IX, respectively. It can be seen that the elitist scheme is capable of evolving the PF with a good convergence for all the three problems. It also gives the best performance in the metric of GD for DTLZ3. This result is expected since the optimal values for variables  $x_5 - x_{14}$  are identical, and the elitist scheme is able to exploit this relationship quickly. On the other hand, it is observed that the random scheme and the hy-

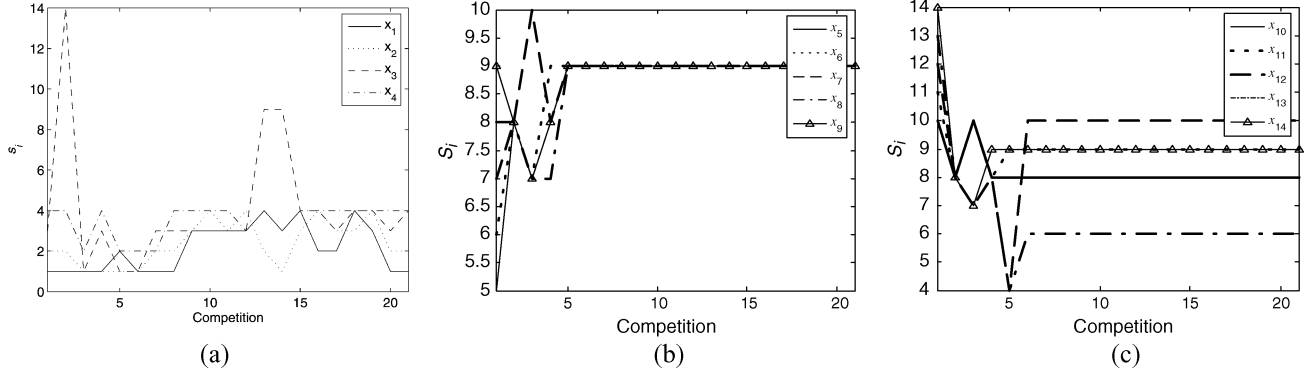


Fig. 11. Trace of the winning subpopulation.

TABLE VII  
PERFORMANCE OF COEA FOR FON WITH DIFFERENT COMPETITION MODELS.  
THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

		Random	Elitist	Hybrid
GD	1st quartile	0.0086	<b>0.0069</b>	0.0071
	Median	0.0133	<b>0.0083</b>	0.0102
	3rd quartile	0.0198	0.0190	<b>0.0158</b>
MS	1st quartile	0.5991	<b>0.671203</b>	0.6646
	Median	0.8036	0.7565	<b>0.8288</b>
	3rd quartile	0.8891	0.8796	<b>0.9125</b>

TABLE VIII  
PERFORMANCE OF COEA FOR KUR WITH DIFFERENT COMPETITION MODELS.  
THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

		Random	Elitist	Hybrid
GD	1st quartile	0.0329	<b>0.0264</b>	0.0306
	Median	<b>0.0376</b>	0.0400	0.0537
	3rd quartile	<b>0.0807</b>	0.1056	0.0918
MS	1st quartile	<b>0.9608</b>	0.8244	0.9491
	Median	<b>0.9902</b>	0.9691	0.9868
	3rd quartile	<b>0.9986</b>	0.9948	0.9955

brid scheme demonstrate better performances when parameter interactions are present. The limitation of high selection pressure introduced by the elitist scheme is also evident from the relatively poor performance in the metric of MS for all problems. Although the random scheme demonstrates the best performance for KUR where the PS\* is discontinuous in the decision space, it produces relatively poor convergence results for FON and DTLZ3. It can also be seen that the hybrid scheme provides competitive results in all cases, and gives the best performance in the metric of MS for FON and DTLZ3.

The elitist scheme is the greediest method which may restrict the exploration of possible relationships among the variables. This explains the reason that it performs well for problems with low variable interactions but provides relatively poor results for problems with high parameter interactions. In contrast, the random scheme is the least greedy approach which allows it

TABLE IX  
PERFORMANCE OF COEA FOR DTLZ3 WITH DIFFERENT COMPETITION MODELS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

		Random	Elitist	Hybrid
GD	1st quartile	0.0000	0.0000	0.0000
	Median	<b>0.0009</b>	0.0010	0.0033
	3rd quartile	0.0248	<b>0.0125</b>	0.0172
MS	1st quartile	<b>0.9972</b>	0.9956	0.9956
	Median	0.9990	0.9979	<b>0.9995</b>
	3rd quartile	0.9998	0.9992	<b>0.9999</b>

to consider the different parameter relationships for maintaining diverse solutions in the evolution. Hence, it performs well for problems with high parameter interactions, but the random nature of the competitor selection also makes it incapable of exploiting the characteristic that the optimal solutions for FON and DTLZ3 lie in the same region. Nonetheless, it is also such a property that allows the random scheme to evolve a more diverse PF as compared with the elitist scheme. On the other hand, the hybrid scheme demonstrates features of both the random and elitist schemes, thus allowing it to attain competitive results that are at least comparable to the other two schemes. Although the three competition schemes behave differently for the different problems, the proposed coevolutionary model produces better performance as compared with conventional approaches. Note that these three schemes are only examples of how different competition models can be applied, and other variants can also be considered if so desired.

## VII. EMPIRICAL STUDY FOR DYNAMIC ENVIRONMENT

This section studies the tracking performance of dCOEA in dynamic environments. The four dynamic MO benchmark problems used in this study are described in Section VII-A, and two static performance metrics are extended from Section V-B to assess the algorithmic performance of dynamic MOEAs in Section VII-B. A basic MOEA and CCEA are also implemented with the mechanisms of random restart and temporal memory, and the performances are compared against dCOEA in Section VII-C. The effects of the proposed stochastic competitors and temporal memory are also examined in Section VII-D.



TABLE X  
DEFINITION OF DYNAMIC TEST FUNCTIONS

Test Function	Definition
1 FDA1	$f_1(x_1) = x_1,$ $f_2(x_2, \dots, x_m) = g \cdot h,$ $g(x_2, \dots, x_m) = 1 + \sum_{i=2}^m (x_i - G(t))^2,$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$ $G(t) = \sin(0.5\pi \cdot t)$ where $m = 10, x_1 \in [0, 1], -1 \leq x_i < 1, \forall i = 2, \dots, 10$
2 dMOP1	$f_1(x_1) = x_1,$ $f_2(x_2, \dots, x_m) = g \cdot h,$ $g(x_2, \dots, x_m) = 1 + 9 \cdot \sum_{i=2}^m x_i^2,$ $h(f_1, g) = 1 - (\frac{f_1}{g})^{H(t)}$ $H(t) = 0.75 \cdot \sin(0.5\pi \cdot t) + 1.25$ where $m = 10, x_i \in [0, 1]$
3 dMOP2	$f_1(x_1) = x_1,$ $f_2(x_2, \dots, x_m) = g \cdot h,$ $g(x_2, \dots, x_m) = 1 + \sum_{i=2}^m (x_i - G(t))^2,$ $h(f_1, g) = 1 - (\frac{f_1}{g})^{H(t)}$ $H(t) = 0.75 \cdot \sin(0.5\pi \cdot t) + 1.25$ $G(t) = \sin(0.5\pi \cdot t)$ where $m = 10, x_i \in [0, 1]$
4 dMOP3	$f_1(x_r) = x_r,$ $f_2(\vec{x} \setminus x_r) = g \cdot h,$ $g(\vec{x} \setminus x_r) = 1 + \sum_{i=1}^{\vec{x} \setminus x_r} (x_i - G(t))^2,$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$ $H(t) = 0.75 \cdot \sin(0.5\pi \cdot t) + 1.25$ $G(t) = \sin(0.5\pi \cdot t)$ where $m = 10, r = U(1, 2, \dots, m), x_i \in [0, 1]$

#### A. Dynamic MO Test Functions

Four different dynamic MO test functions are applied here to examine the performance of dCOEA in tracking the changing MO fitness landscape. The first problem is FDA1 [14], a Type I problem where only the  $PS^*$  is dynamic. The other three test functions are based on the construction guidelines provided by Farina *et al.* [14]. The test function of dMOP1 is a Type III problem where only the  $PF^*$  is dynamic, and the function of dMOP2 is a Type II problem where both the  $PS^*$  and  $PF^*$  change with time. Like FDA1, dMOP3 is also a Type I problem but the variable that controls the spread of the  $PF^*$  changes as well. The definition of these dynamic MO test functions is summarized in Table X.

The dynamics of these test functions is governed by the equation

$$t = \frac{1}{n_T} \left\lfloor \frac{\tau}{\tau_T} \right\rfloor \quad (7)$$

where  $n_T$  and  $\tau_T$  specifies the severity and frequency of landscape change, respectively. Due to the sinusoidal behavior of

$G(t)$  and  $H(t)$ ,  $n_T$  also determines the periodicity of the same  $PS_t^*$  reemerging. In particular, a low value of  $n_T$  implies that the number of different  $PS_t^*$  is small. Both  $n_T$  and  $\tau_T$  have a lower bound of 1. A small value of  $n_T < 1$  will result in a magnitude change that is out of range, while a larger value of  $n_T > 1$  will produce a smaller magnitude of change. Likewise, a larger value of  $\tau_T$  will result in an increasingly condition of static environments.

#### B. Dynamic Performance Metrics

The objective of dynamic MO optimization is not only to evolve a near optimal and diverse  $PF_t^A$ , but also to track the dynamic  $PF_t^*$ . Here,  $PF_t^A$  refers to the set of nondominated solutions stored in the archive at time  $t$ . The performance metrics of dynamic MOEAs should, thus, indicate the following criteria.

- The effectiveness of the dynamic MOEA in attaining the MO optimization goals of proximity, diversity, and distribution in the face of changing physical spatial attributes.
- The speed of the dynamic MOEA in converging to the new solution set, since there could be a restriction on time.

The static performance metrics described in Section VI-B can be extended to provide performance assessment of MOEAs in dynamic environments. A simple extension is to illustrate the performance trend over time [10], [14], [66] by assessing the evolved solutions with respect to static metrics at different time instances. For example, Zeng *et al.* [66] calculated and tabulated the GD and spread [11] of  $PF_t^A$  at time instances just before the next landscape change is triggered. However, as pointed out by Branke [4], it is desirable to have a more compact form of assessment about algorithmic performance. One approach is to extend the idea of offline error applied in dynamic SO optimization and calculate the time averages of static metrics [23]. Intuitively, the sampling of performance metrics should be performed at instances right before the next landscape variation in order to eliminate unnecessary penalty in situations where the change is small on dynamic MOEAs having diversity introduction schemes such as random restart or hypermutation.

In this study, the variable space GD (VD) and MS in the following form:

$$VD_{\text{offline}} = \frac{1}{\tau} \sum_{t=1}^{\tau} VD \cdot I(t) \quad (8)$$

$$MS_{\text{offline}} = \frac{1}{\tau} \sum_{t=1}^{\tau} MS \cdot I(t) \quad (9)$$

$$I(t) = \begin{cases} 1, & \text{if } t \% \tau_T = 0 \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

where % is the modulus operator. Similar to the metric of GD in static environment, a low value of  $VD_{\text{offline}}$  is desirable and reflects good tracking performance. Likewise, a high value of  $MS_{\text{offline}}$  shows that the MOEA is capable of evolving a diverse  $PF_t^A$  in a dynamic environment. The VD measures the degree of convergence between the  $PS_t^*$  and  $PS_t^A$ , where the  $PS_t^A$  refers to the Pareto solution set stored in the archive at time  $t$ , e.g.,

$$VD = \frac{1}{n_{PS}} \cdot \left( n_{PS} \sum_{i=1}^{n_{PS}} d_i^2 \right)^{\frac{1}{2}} \quad (11)$$

where  $n_{PS} = |PS_t^A|$ ,  $d_i$  is the Euclidean distance (in decision space) between the  $i$ th member of  $PS_t^A$  and the nearest member of  $PS_t^*$ .

### C. Comparative Study

To examine the performance of dCOEA, two different dynamic MOEAs based on a basic MOEA and the CCEA, respectively, are adopted in this study. The basic MOEA is similar to the model presented in [19]. The algorithm employs a fixed-size population and an archive to store nondominated solutions along the evolution. The archive is updated at each cycle, i.e., a candidate solution will be added to the archive if it is not dominated by any members in the archive. Likewise, any archive members dominated by this solution will be removed from the archive. When the predetermined archive size is reached, a recurrent truncation process based on niche count is used to eliminate the most crowded archive member. Elitism is implemented by selecting individuals to a mating pool through binary tournament selection of the combined archive and evolving population. The selection criterion is based on Pareto rank, and niche count is

TABLE XI  
PARAMETER SETTING FOR DIFFERENT ALGORITHMS

Parameter	Settings
Populations	Population size 100 in dMOEA; Subpopulation size 10 in dCOEA and dCCEA; Archive (or secondary population) size 100.
Chromosome	Binary coding; 30 bits per decision variable.
Selection	Binary tournament selection
Crossover operator	Uniform crossover
Crossover rate	0.8
Mutation operator	Bit-flip mutation
Mutation rate	$\frac{1}{L}$ for FDA1, dMOP1, dMOP2 and dMOP3;
Niche Radius	Dynamic sharing.
Evaluation number	20,000

used in the event of a tie. In both the dynamic MOEA (dMOEA) and dynamic CCEA (dCCEA), a fixed number of archived solutions are reevaluated in every generation. In the case where a change in the landscape is detected, the temporal memory described previously will be applied, and random restart is incorporated to generate diversity within the evolving population.

The parameter settings for the different algorithms are tabulated in Table XI. Thirty independent simulation runs with randomly generated initial populations are performed for each of the test problems. The experiments are conducted at different severity levels of  $n_T = \{1, 10, 20\}$  and different frequencies of  $\tau_T = \{5, 10, 25\}$  so as to study the impact of dynamics in uncertain environments. Since each generation involves 100 function evaluations, the setting of  $\tau_T = 5$  implies a change of the landscape in very 500 evaluations. In this study,  $SC_{\text{ratio}}$  and  $R_{\text{size}}$  are set as 0.5 and 5, respectively.

1) *FDA1*: The FDA1 challenges the dynamic MOEAs' ability to track and converge towards the  $PF_t^*$  with every landscape change. One interesting characteristic of this problem is that the distribution and diversity of the solutions along the  $PF_t$  are not affected by the landscape change. The simulation results for  $VD_{\text{offline}}$  and  $MS_{\text{offline}}$  with various settings of  $\tau_T$  and  $n_T$  are summarized in Table XII. In general, the coevolutionary paradigm is shown to be more appropriate than canonical MOEAs in handling dynamic landscapes. As can be seen, the dCOEA outperforms dCCEA in both the aspect of tracking and finding a diverse solution set. Table XII also shows a better convergence and diversity performance of dMOEA, dCCEA, and dCOEA for larger value of  $\tau_T$  or less frequent landscape changes. Although the dMOEA gives a better convergence for larger value of  $n_T$  or less severe landscape changes, it is observed that better results of dCCEA and dCOEA can be obtained with a more severe landscape change.

2) *dMOP1*: Unlike FDA1, the convexity of dMOP1 changes with time, while the location of  $PS^*$  remains fixed. The dMOP1 challenges the dynamic MOEA's ability to maintain a diverse  $PF_t^*$  with every landscape change. The simulation results for  $VD_{\text{offline}}$  and  $MS_{\text{offline}}$  with various settings of  $\tau_T$  and  $n_T$  are summarized in Table XIII. Similar to the problem of FDA1, the

TABLE XII  
PERFORMANCE OF MOEA, dCCEA, AND dCOEA FOR FDA1 AT DIFFERENT SETTINGS OF  $\tau_T$  AND  $n_T$ .  
THE BEST RESULTS ARE HIGHLIGHTED IN BOLD ONLY IF IT IS STATISTICALLY DIFFERENT BASED ON THE KS TEST

$(\tau_t, n_T)$		$VD_{offline}$			$MS_{offline}$		
		MOEA	dCCEA	dCOEA	MOEA	dCCEA	dCOEA
(5,10)	1st quartile	0.666	0.243	<b>0.107</b>	0.789	0.829	<b>0.939</b>
	Median	0.683	0.255	<b>0.110</b>	0.801	0.834	<b>0.944</b>
	3rd quartile	0.695	0.264	<b>0.113</b>	0.801	0.841	<b>0.953</b>
(10,10)	1st quartile	0.489	0.154	<b>0.034</b>	0.870	0.863	<b>0.963</b>
	Median	0.508	0.163	<b>0.038</b>	0.878	0.873	<b>0.970</b>
	3rd quartile	0.521	0.167	<b>0.039</b>	0.890	0.882	<b>0.977</b>
(25,10)	1st quartile	0.485	0.080	<b>0.001</b>	0.876	0.926	<b>0.979</b>
	Median	0.528	0.091	<b>0.002</b>	0.894	0.939	<b>0.985</b>
	3rd quartile	0.583	0.102	<b>0.003</b>	0.914	0.947	<b>0.989</b>
(10,1)	1st quartile	1.008	0.135	<b>0.020</b>	0.535	0.857	<b>0.973</b>
	Median	1.031	0.149	<b>0.022</b>	0.585	0.866	<b>0.981</b>
	3rd quartile	1.064	0.156	<b>0.025</b>	0.599	0.883	<b>0.984</b>
(10,20)	1st quartile	0.542	0.152	<b>0.039</b>	0.847	0.858	<b>0.970</b>
	Median	0.584	0.162	<b>0.042</b>	0.868	0.875	<b>0.975</b>
	3rd quartile	0.606	0.171	<b>0.044</b>	0.881	0.888	<b>0.979</b>

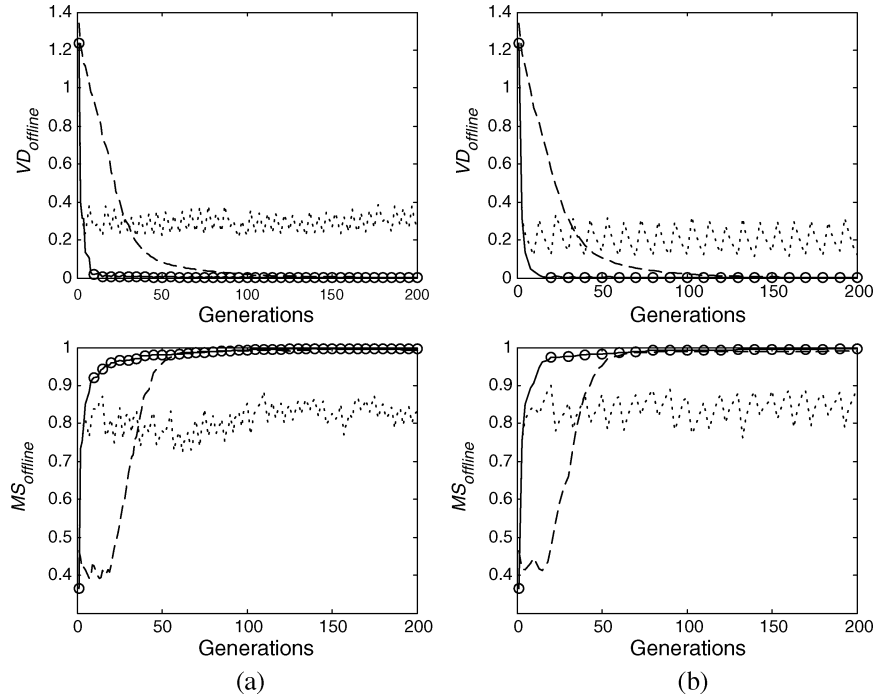


Fig. 12. Evolutionary trace of dMOEA (—), dCCEA (---), and dCOEA (o) for dMOP1 at (a)  $\tau_T = 5$  and  $n_T = 10$  and (b)  $\tau_T = 10$  and  $n_T = 10$ .

dCOEA outperforms dMOEA and dCCEA in both the aspect of tracking and finding a diverse solution set. It is also observed that the dMOEA produces better results than dCCEA for the settings of  $\tau_T = 5$  and  $\tau_T = 10$ . The evolutionary trace of  $VD_{offline}$  and  $MS_{offline}$  for these settings are shown in Fig. 12. While the dCOEA and dCCEA behave similarly in the initial generations before the first landscape change, it can be seen that the dCCEA is greatly affected by the change in the shape of PF.

On the other hand, the dMOEA is capable of finding the  $PS_t^*$  and a diverse  $PF_t$  despite having a slower convergence. Based on previous studies in dynamic SO optimization, diversity scheme like random restart tends to perform poorly in situation where the change is small. As compared with the problem of FDA1, the severity of change has less impact on the metric of  $VD_{offline}$  for these three algorithms. This is due to the incorporation of temporal memory that allows the algorithm to rediscover the  $PS^*$

TABLE XIII  
PERFORMANCE OF MOEA, dCCEA, AND dCOEA FOR dMOP1 DIFFERENT SETTINGS OF  $\tau_T$  AND  $n_T$ .  
THE BEST RESULTS ARE HIGHLIGHTED IN BOLD ONLY IF IT IS STATISTICALLY DIFFERENT BASED ON THE KS TEST

$(\tau_t, n_T)$		VD <sub>offline</sub>			MS <sub>offline</sub>		
		MOEA	dCCEA	dCOEA	MOEA	dCCEA	dCOEA
(5,10)	1st quartile	0.114	0.230	<b>0.005</b>	0.891	0.825	<b>0.977</b>
	Median	0.128	0.242	<b>0.007</b>	0.911	0.838	<b>0.983</b>
	3rd quartile	0.137	0.252	<b>0.008</b>	0.933	0.846	<b>0.989</b>
(10,10)	1st quartile	0.103	0.111	<b>0.002</b>	0.916	0.880	<b>0.988</b>
	Median	0.114	0.121	<b>0.003</b>	0.916	0.880	<b>0.988</b>
	3rd quartile	0.131	0.132	<b>0.004</b>	0.935	0.888	<b>0.994</b>
(25,10)	1st quartile	0.065	0.023	<b>0.001</b>	0.916	0.931	<b>0.989</b>
	Median	0.077	0.026	<b>0.00</b>	0.940	0.948	<b>0.991</b>
	3rd quartile	0.093	0.030	<b>0.001</b>	0.962	0.962	<b>0.996</b>
(10,1)	1st quartile	0.106	0.120	<b>0.002</b>	0.891	0.870	<b>0.986</b>
	Median	0.116	0.126	<b>0.003</b>	0.914	0.877	<b>0.990</b>
	3rd quartile	0.128	0.137	<b>0.004</b>	0.934	0.893	<b>0.992</b>
(10,20)	1st quartile	0.101	0.115	<b>0.002</b>	0.904	0.871	<b>0.982</b>
	Median	0.117	0.123	<b>0.003</b>	0.921	0.881	<b>0.988</b>
	3rd quartile	0.130	0.133	<b>0.003</b>	0.939	0.890	<b>0.993</b>

TABLE XIV  
PERFORMANCE OF MOEA, dCCEA, AND dCOEAS FOR dMOP2 AT DIFFERENT SETTINGS OF  $\tau_T$  AND  $n_T$ .  
THE BEST RESULTS ARE HIGHLIGHTED IN BOLD ONLY IF IT IS STATISTICALLY DIFFERENT BASED ON THE KS TEST

$(\tau_t, n_T)$		VD <sub>offline</sub>			MS <sub>offline</sub>		
		MOEA	dCCEA	dCOEA	MOEA	dCCEA	dCOEA
(5,10)	1st quartile	0.642	<b>0.285</b>	0.352	0.973	0.852	<b>0.988</b>
	Median	0.666	<b>0.291</b>	0.372	0.981	0.861	<b>0.991</b>
	3rd quartile	0.680	<b>0.300</b>	0.384	0.986	0.871	<b>0.994</b>
(10,10)	1st quartile	0.495	0.159	0.173	0.976	0.886	<b>0.991</b>
	Median	0.517	0.169xx	0.180	0.980	0.902	<b>0.993</b>
	3rd quartile	0.535	0.187	0.192	0.987	0.915	<b>0.996</b>
(25,10)	1st quartile	0.462	0.069	<b>0.059</b>	0.9817	0.949	<b>0.991</b>
	Median	0.514	0.075	<b>0.063</b>	0.989	0.958	<b>0.994</b>
	3rd quartile	0.557	0.093	<b>0.071</b>	0.993	0.964	<b>0.997</b>
(10,1)	1st quartile	1.137	0.176	<b>0.140</b>	0.965	0.881	<b>0.991</b>
	Median	1.166	0.186	<b>0.152</b>	0.978	0.899	<b>0.996</b>
	3rd quartile	1.188	0.202	<b>0.176</b>	0.985	0.912	<b>0.998</b>
(10,20)	1st quartile	0.466	0.166	0.162	0.966	0.889	<b>0.991</b>
	Median	0.487	0.177	0.170	0.979	0.899	<b>0.992</b>
	3rd quartile	0.519	0.185	0.184	0.986	0.916	<b>0.996</b>

quickly, even though random restart is utilized in the dMOEA and dCCEA.

3) *dMOP2*: The convexity and  $PS_t^*$  of dMOP2 change with time, which challenge the dynamic MOEA's ability to track the  $PS_t^*$  and to maintain a diverse  $PF_t^*$  with every landscape change. The simulation results for VD<sub>offline</sub> and MS<sub>offline</sub> with various settings of  $\tau_T$  and  $n_T$  are summarized in Table XIV. In contrast

to the previous two problems, the dCOEA is outperformed by dCCEA for VD<sub>offline</sub> when  $(\tau_T, n_T)$  is set as (5,10) and (10,10). Since random restart is applied in dCOEA, further investigations in the next section demonstrate that a lower  $SC_{ratio}$  allows the dCOEA to give a better performance. On the other hand, the dCOEA outperforms both dMOEA and dCCEA in tracking and maintaining better diversity for other parameter settings. By

TABLE XV  
PERFORMANCE OF MOEA, dCCEA, AND dCOEAS FOR dMOP3 AT DIFFERENT SETTINGS OF  $\tau_T$  AND  $n_T$ .  
THE BEST RESULTS ARE HIGHLIGHTED IN BOLD ONLY IF IT IS STATISTICALLY DIFFERENT BASED ON THE KS TEST

$(\tau_t, n_T)$		$VD_{offline}$			$MS_{offline}$		
		MOEA	dCCEA	dCOEA	MOEA	dCCEA	dCOEA
(5,10)	1st quartile	0.679	0.226	<b>0.083</b>	0.619	0.824	<b>0.906</b>
	Median	0.701	0.2398	<b>0.087</b>	0.637	0.835	<b>0.913</b>
	3rd quartile	0.727	0.249	<b>0.09</b>	0.658	0.841	<b>0.927</b>
(10,10)	1st quartile	0.460	0.140	<b>0.013</b>	0.802	0.856	<b>0.943</b>
	Median	0.482	0.149	<b>0.017</b>	0.822	0.867	<b>0.957</b>
	3rd quartile	0.507	0.162	<b>0.021</b>	0.843	0.880	<b>0.965</b>
(25,10)	1st quartile	0.424	0.068	<b>0.001</b>	0.903	0.927	<b>0.976</b>
	Median	0.467	0.078	<b>0.002</b>	0.914	0.9338	<b>0.983</b>
	3rd quartile	0.515	0.096	<b>0.003</b>	0.927	0.949	<b>0.987</b>
(10,1)	1st quartile	1.055	0.129	<b>0.011</b>	0.505	0.861	<b>0.977</b>
	Median	1.087	0.138	<b>0.014</b>	0.539	0.873	<b>0.981</b>
	3rd quartile	1.108	0.15	<b>0.018</b>	0.565	0.886	<b>0.987</b>
(10,20)	1st quartile	0.477	0.138	<b>0.019</b>	0.837	0.855	<b>0.946</b>
	Median	0.505	0.147	<b>0.022</b>	0.857	0.865	<b>0.954</b>
	3rd quartile	0.538	0.155	<b>0.025</b>	0.866	0.883	<b>0.966</b>

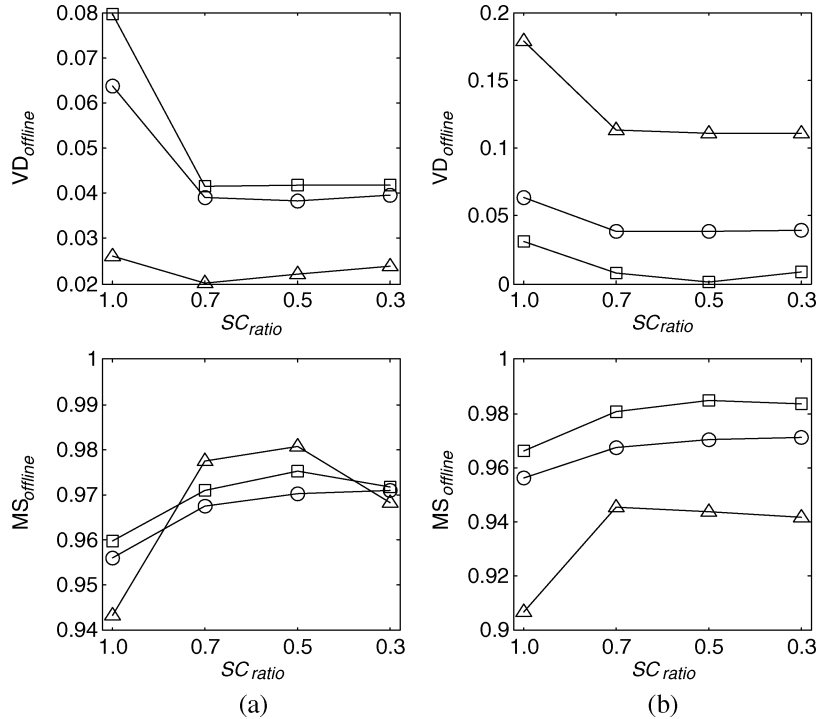


Fig. 13. Performance metrics of  $VD_{offline}$  (top) and  $MS_{offline}$  (bottom) for FDA1 over different settings of  $SC_{ratio}$  (a) at  $n_T = 1(\Delta)$ ,  $n_T = 10(\circ)$ , and  $n_T = 20(\square)$  and (b)  $\tau_T = 5(\Delta)$ ,  $\tau_T = 10(\circ)$ , and  $\tau_T = 25(\square)$ .

comparing the values of  $MS_{offline}$  in Tables XII–XIV, it can be observed that dCCEA is unable to find a diverse  $PF_t$  when the shape of  $PF_t^*$  is dynamic.

4) *dMOP3*: Although the dMOP3 has similar characteristic as FDA1, the variable that determines the spread of the solution set in dMOP3 is not fixed and changes with time. The dynamic

MOEA, thus, faces an additional challenge in tracking the diverse  $PF_t^*$  for this problem. The simulation results for  $VD_{offline}$  and  $MS_{offline}$  with various settings of  $\tau_T$  and  $n_T$  are summarized in Table XV. By comparing the results in Tables XII and XIII, it can be seen that the three algorithms give poorer performances for  $MS_{offline}$  with  $\tau_T = 5$  and  $\tau_T = 10$ . Nonetheless,

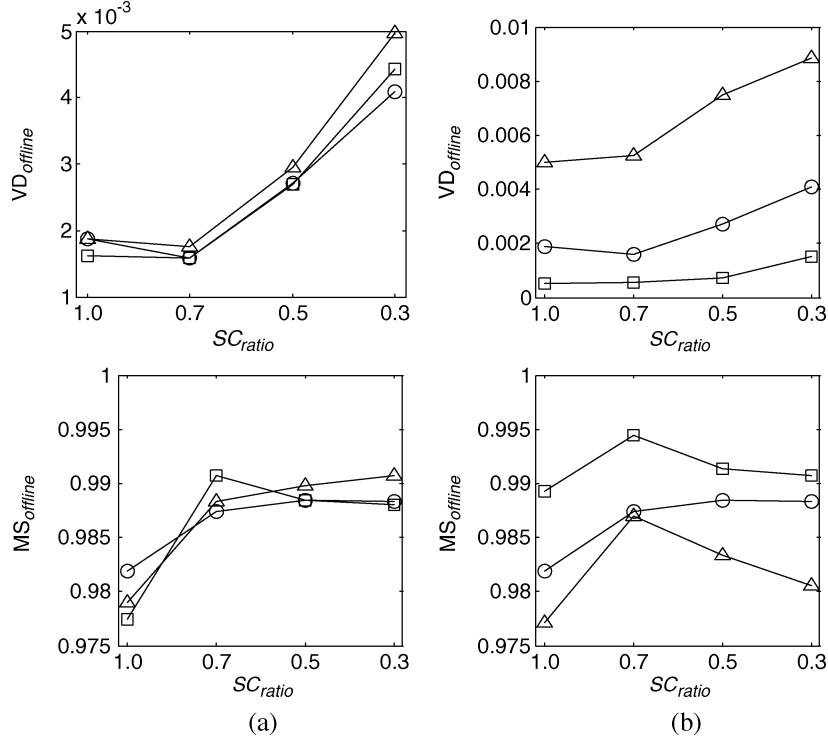


Fig. 14. Performance metrics of  $VD_{\text{offline}}$  (top) and  $MS_{\text{offline}}$  (bottom) for dMOPI over different settings of  $SC_{\text{ratio}}$  (a) at  $n_T = 1(\Delta)$ ,  $n_T = 10(\circ)$ , and  $n_T = 20(\square)$  and (b)  $\tau_T = 5(\Delta)$ ,  $\tau_T = 10(\circ)$ , and  $\tau_T = 25(\square)$ .

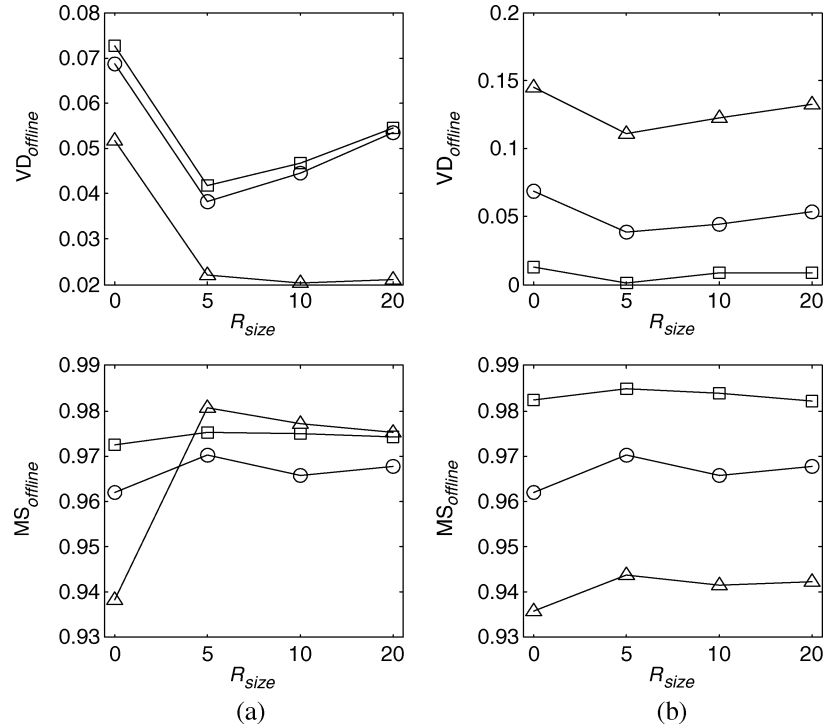


Fig. 15. Performance metrics of  $VD_{\text{offline}}$  (top) and  $MS_{\text{offline}}$  (bottom) for FDA1 over different settings of  $R_{\text{size}}$  at (a)  $n_T = 1(\Delta)$ ,  $n_T = 10(\circ)$ , and  $n_T = 20(\square)$  and (b)  $\tau_T = 5(\Delta)$ ,  $\tau_T = 10(\circ)$ , and  $\tau_T = 25(\square)$ .

similar to the problems of FDA1 and dMOP2, it is observed that the dCOEA outperforms dMOEA and dCCEA in both the aspect of tracking and finding a diverse solution set for the different settings of  $\tau_T$  and  $n_T$ .

#### D. Effects of Stochastic Competitors

The  $SC_{\text{ratio}}$  determines the degree of diversity introduced in the proposed dCOEA after every landscape change for good

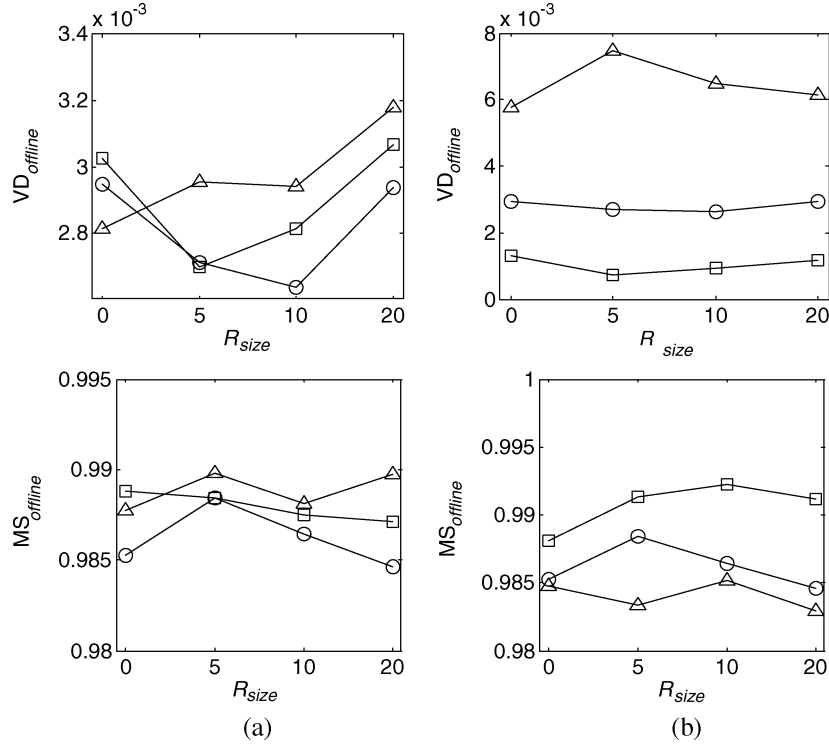


Fig. 16. Performance metrics of  $VD_{\text{offline}}$  (top) and  $MS_{\text{offline}}$  (bottom) for dMOP1 over different settings of  $R_{\text{size}}$  at (a)  $n_T = 1(\triangle)$ ,  $n_T = 10(\circ)$ , and  $n_T = 20(\square)$  and (b)  $\tau_T = 5(\triangle)$ ,  $\tau_T = 10(\circ)$ , and  $\tau_T = 25(\square)$

tracking performance. The relationships between  $SC_{\text{ratio}} = \{0.3, 0.5, 0.7, 1\}$  and various settings of  $n_T$  and  $\tau_T$  for FDA1 are shown in Fig. 13. Note that no stochastic competitors are introduced when  $SC_{\text{ratio}} = 1$ . These relationships are similarly investigated for the problems of dMOP1, dMOP2, and dMOP3. The trends for dMOP1 are illustrated in Fig. 14, while the plots for dMOP2 and dMOP3 are omitted here since similar traits with FDA1 have been exhibited.

It can be seen from the metric of  $MS_{\text{offline}}$  in Figs. 13 and 14 that the diversity of the evolved  $PF_t$  improves generally with the introduction of stochastic competitors. On the other hand, Fig. 14 shows that the tracking performance deteriorates with increasing diversity for dMOP1. This is because the location of  $PS_t^*$  remains unchanged for this problem, and it is improbable that the new set of nondominated solutions introduced by the stochastic competitors will be better or even comparable to the archived solutions before any landscape change. Nonetheless, it is clear that stochastic competitors play an important role in the tracking of dynamic  $PS_t^*$  for the problems of FDA1, dMOP2, and dMOP3.

It can be also observed from Fig. 13 that the dCOEA produces the best results when  $n_T = 1$  for FDA1. In contrast, Fig. 14 shows that the dCOEA gives the worst results for dMOP1 for the same setting of  $n_T = 1$ . This observation is similar to past findings from dynamic SO optimization that a higher degree of diversity is required with severe environment changes in the environment.

#### E. Effects of Temporal Memory

The  $R_{\text{size}}$  determines the extent in which information about past  $PS_t^*$  is stored. A larger  $R_{\text{size}}$  results in a higher degree of

information exploitation at the expense of a more diverse repertoire of past  $PS_t^*$ . On the other hand, limited information regarding past  $PS_t^*$  is available when  $R_{\text{size}}$  is small. The relationships between  $R_{\text{size}} = \{0, 5, 10, 20\}$  and various settings of  $n_T$  and  $\tau_T$  for FDA1 are shown in Fig. 15. Note that no memory is kept when  $R_{\text{size}} = 0$ . These relationships are similarly investigated for dMOP1, dMOP2, and dMOP3. The trends for dMOP1 are shown in Fig. 16, while the plots for dMOP2 and dMOP3 are excluded here since similar traits have been observed.

Similar to the findings in Section VII-D, better tracking performances are observed for higher  $\tau_T$  and at  $n_T = 1$  in FDA1, dMOP2, and dMOP3 for the different settings of  $R_{\text{size}}$ . Fig. 16 illustrates that the incorporation of temporal memory tends to improve convergence according to the metric of  $VD_{\text{offline}}$ . The only exception occurs at the setting of  $n_T = 1$  and  $\tau_T = 5$  for dMOP1. The tradeoff between the exploration and exploitation of information is also evident from the figures with increasing  $R_{\text{size}}$ . For instance, when the repetition of similar  $PS_t^*$  is very frequent for  $n_T = 1$ , a large  $R_{\text{size}}$  can be used to mine information from past  $PS_t^*$ , since the number of different  $PS_t^*$  that needs to be represented in the memory is small, and *vice versa*.

## VIII. CONCLUSION

This paper presented two main contributions: 1) the proposal of a new coevolutionary paradigm that incorporates both competitive and cooperative mechanisms observed in nature to facilitate adaptive problem decomposition and 2) the application of the new paradigm to solve MO optimization problems and to track the Pareto front in a dynamic environment. The proposed competitive-cooperative coevolution overcomes limitations in conventional coevolutionary models by allowing decomposition

process of the optimization problem to emerge based on the problem requirements. It also exploits the high speed of convergence in coevolution to allow the algorithm to adapt quickly to the changing environment.

Based on this coevolutionary model, a competitive-cooperative coevolutionary algorithm (COEA) for MO optimization has been proposed. Extensive simulations upon benchmark problems demonstrate that the COEA is capable of evolving near-optimal, diverse, and uniformly distributed Pareto fronts even for problems with severe parameter interactions. The dynamics and performance of the various competitive mechanisms have also been analyzed, illustrating the importance and robustness of both competitive and cooperative elements in a common coevolution framework.

Besides, a dynamic COEA (dCOEA) has also been proposed, which incorporates the features of: 1) stochastic competitors to track the changing solution set and 2) temporal memory to exploit past information. The performance of the dCOEA has been examined based upon different severity and frequency of change in dynamic environments. Simulation results show that the dCOEA is capable of tracking the Pareto front over different environmental changes effectively and efficiently.

#### ACKNOWLEDGMENT

The authors wish to thank the Associate Editor and anonymous reviewers for their valuable comments and helpful suggestions which greatly improved the paper's quality.

#### REFERENCES

- [1] P. J. Angeline and J. B. Pollack, "Competitive environments evolve better solutions for complex tasks," in *Proc. 5th Int. Conf. Genetic Algorithms*, 1993, pp. 264–270.
- [2] H. J. C. Barbosa and A. M. S. Barreto, "An interactive genetic algorithm with co-evolution of weights for multiobjective problems," in *Proc. 2001 Genetic Evol. Comput. Congr.*, 2001, pp. 203–210.
- [3] P. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, 2003.
- [4] J. Branke, *Evolutionary Optimization in Dynamic Environments*. Norwell, MA: Kluwer, 2001.
- [5] J. Branke and H. Schmuck, "Designing evolutionary algorithms for dynamic optimization problems," in *Theory and Application of Evolutionary Computation: Recent Trends*, S. Tsutsui and A. Ghosh, Eds. Berlin, Germany: Springer-Verlag, 2002, pp. 239–262.
- [6] H. G. Cobb, "An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments," Naval Research Lab., Washington, DC, Tech. Rep. AIC-90-001, 1990.
- [7] C. A. C. Coello and M. R. Sierra, "A coevolutionary multi-objective evolutionary algorithm," in *Proc. 2003 Congr. Evol. Comput.*, 2003, vol. 1, pp. 482–489.
- [8] D. W. Corne, J. D. Knowles, and M. J. Oates, "The Pareto envelope-based selection algorithm for multiobjective optimization," in *Proc. 6th Int. Conf. Parallel Problem Solving From Nature*, 2000, pp. 839–848.
- [9] K. Deb, "Multi-objective genetic algorithms: Problem difficulties and construction of test problem," *Evol. Comput.*, vol. 7, no. 3, pp. 205–230, 1999.
- [10] K. Deb, U. B. N. Rao, and S. Karthik, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling," Kanpur Genetic Algorithms Lab. (KanGAL), Indian Instit. Technol., Kanpur, India, Tech. Rep. 2006008, 2006.
- [11] 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, 2002.
- [12] K. Deb and H. Gupta, "Introducing robustness in multiobjective optimization," Kanpur Genetic Algorithms Lab. (KanGAL), Indian Inst. Technol., Kanpur, India, Tech. Rep. 2004016, 2004.
- [13] K. Deb and D. E. Goldberg, "An investigation on niche and species formation in genetic function optimization," in *Proc. 3rd Int. Conf. Genetic Algorithms*, 1989, pp. 42–50.
- [14] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: Test cases, approximations, and applications," *IEEE Trans. Evol. Comput.*, vol. 8, no. 5, pp. 425–442, 2003.
- [15] M. Farina and P. Amato, "A fuzzy definition of 'optimality' for many-criteria optimization problems," *IEEE Trans. Syst., Man, Cybern.-Part A: Systems and Humans*, vol. 34, no. 3, pp. 315–326, 2004.
- [16] C. M. Fonseca and P. J. Fleming, "Genetic algorithm for multiobjective optimization, formulation, discussion and generalization," in *Proc. 5th Int. Conf. Genetic Algorithms*, 1993, pp. 416–423.
- [17] N. Garcia-Pedrajas, C. Hervás-Martínez, and D. Ortiz-Boyer, "Cooperative coevolution of artificial neural network ensembles for pattern classification," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 271–302, 2005.
- [18] A. Ghosh, S. Tsutsui, and H. Tanaka, "Function optimization in non-stationary environment using steady state genetic algorithms with aging of individuals," in *Proc. 1998 IEEE Congr. Evol. Comput.*, 1998, pp. 666–671.
- [19] C. K. Goh and K. C. Tan, "An investigation on noisy environments in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 3, pp. 354–381, 2007.
- [20] J. J. Grefenstette, "Genetic algorithms for changing environments," in *Proc. 2nd Int. Conf. Parallel Problem Solving from Nature*, 1992, pp. 137–144.
- [21] J. J. Grefenstette and C. L. Ramsey, "An approach to anytime learning," in *Proc. 9th Int. Conf. Mach. Learn.*, 1987, pp. 41–49.
- [22] H. Gupta and K. Deb, "Handling constraints in robust multi-objective optimization," in *Proc. 2005 IEEE Congr. Evol. Comput.*, 2005, pp. 25–32.
- [23] I. Hatzakis and D. Wallace, "Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach," in *Proc. 2006 Genetic Evol. Comput. Congr.*, 2006, pp. 1201–1208.
- [24] D. W. Hillis, "Coevolving parasites improve simulated evolution as an optimization procedure," in *Artificial Life II*, C. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, Eds. Reading, MA: Addison-Wesley, 1991, pp. 313–324.
- [25] J. Horn and N. Nafpliotis, "Multiobjective optimization using the niched Pareto genetic algorithm," Illinois Genetic Algorithms Lab. (IlligAL), Univ. Illinois at Urbana-Champaign, Urbana, IL, Tech. Rep. 930005, 1993.
- [26] E. J. Hughes, "Evolutionary many-objective optimization: Many once or one many?," in *Proc. 2005 IEEE Congr. Evol. Comput.*, 2005, vol. 1, pp. 222–227.
- [27] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evol. Comput.*, vol. 15, no. 1, pp. 1–28, 2007.
- [28] A. W. Iorio and X. Li, "A cooperative coevolutionary multiobjective algorithm using non-dominated sorting," in *Proc. 2004 Genetic Evol. Comput. Congr.*, 2004, pp. 537–548.
- [29] A. Jaszkiewicz, "Do multi-objective metaheuristics deliver on their promises? A computational experiment on the set-covering problem," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 133–143, 2003.
- [30] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments—A survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, 2005.
- [31] Y. Jin and B. Sendhoff, "Constructing dynamic optimization test problems using the multi-objective optimization concept," in *Proc. 2004 EvoWorkshops*, 2004, pp. 525–536.
- [32] N. Keeratitumrong, N. Chaiyaratana, and V. Varavithya, "Multi-objective co-operative co-evolutionary genetic algorithm," in *Proc. 7th Int. Conf. Parallel Problem Solving From Nature*, 2002, pp. 288–297.
- [33] V. Khare, X. Yao, and B. Sendhoff, "Credit assignment among neurons in co-evolving populations," in *Proc. 8th Int. Conf. Parallel Problem Solving From Nature*, 2004, pp. 882–891.
- [34] V. Khare, X. Yao, and K. Deb, "Performance scaling of multi-objective evolutionary algorithms," in *Proc. 2nd Int. Conf. Evol. Multi-Criterion Opt.*, 2003, pp. 376–390.
- [35] E. F. Khor, K. C. Tan, T. H. Lee, and C. K. Goh, "A study on distribution preservation mechanism in evolutionary multi-objective optimization," *Artif. Intell. Rev.*, vol. 23, no. 1, pp. 31–56, 2005.
- [36] J. D. Knowles, "ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 50–66, 2006.
- [37] J. D. Knowles and D. W. Corne, "On metrics for comparing nondominated sets," in *Proc. 2002 IEEE Congr. Evol. Comput.*, 2002, vol. 1, pp. 711–716.
- [38] J. D. Knowles and D. W. Corne, "Approximating the non-dominated front using the Pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, 2000.



- [39] M. Laumanns, E. Zitzler, and L. Thiele, "A unified model for multi-objective evolutionary algorithms with elitism," in *Proc. 2000 Congr. Evol. Comput.*, 2000, vol. 1, pp. 46–53.
- [40] M. Laumanns, G. Rudolph, and H.-P. Schwefel, "A spatial predator-prey approach to multi-objective optimization: A preliminary study," in *Proc. 5th Int. Conf. Parallel Problem Solving From Nature*, 1998, pp. 241–249.
- [41] J. D. Lohn, W. F. Kraus, and G. L. Haith, "Comparing a coevolutionary genetic algorithm for multiobjective optimization," in *Proc. 2002 Congr. Evol. Comput.*, 2002, pp. 1157–1162.
- [42] K. Maneeratana, K. Boonlong, and N. Chaiyaratana, "Multi-objective optimisation by co-operative co-evolution," in *Proc. 8th Int. Conf. Parallel Problem Solving From Nature*, 2004, pp. 772–781.
- [43] J. Mehnen, T. Wagner, and G. Rudolph, "Evolutionary optimization of dynamic multi-objective test functions," in *Proc. 2nd Italian Workshop Evol. Comput.*, 2006.
- [44] N. Mori, H. Kita, and Y. Nishikawa, "Adaptation to a changing environment by means of the thermodynamical genetic algorithm," in *Proc. 4th Int. Conf. Parallel Problem Solving From Nature*, 1996, pp. 513–522.
- [45] R. Morrison, *Designing Evolutionary Algorithms for Dynamic Environments*. Berlin, Germany: Springer-Verlag, 2004.
- [46] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evol. Comput.*, vol. 9, no. 2, pp. 159–195, 2001.
- [47] Y. S. Ong, A. J. Keane, and P. B. Nair, "Surrogate-assisted coevolutionary search," in *Proc. 9th Int. Conf. Neural Inf. Process.*, 2002, pp. 2195–2199.
- [48] J. Paredis, "Coevolutionary constraint satisfaction," in *Proc. 3rd Int. Conf. Parallel Problem Solving From Nature*, 1994, pp. 46–55.
- [49] M. A. Potter, "The design and analysis of a computational model of cooperative coevolution," Ph.D. dissertation, George Mason Univ., Fairfax, VA, 1997.
- [50] M. A. Potter and K. A. De Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evol. Comput.*, vol. 8, no. 1, pp. 1–29, 2000.
- [51] C. D. Rosin and R. K. Belew, "New methods for competitive coevolution," *Evol. Comput.*, vol. 5, no. 1, pp. 1–29, 1997.
- [52] J. R. Scott, "Fault tolerant design using single and multi-criteria genetic algorithms," M.S. thesis, Dept. Aeronautics and Astronautics, Mass. Inst. Technol., Cambridge, MA, 1995.
- [53] W. M. Schaffer, D. W. Zeh, S. L. Buchmann, S. Kleinhaus, M. V. Schaffer, and J. Antrim, "Competition for nectar between introduced honeybees and native North American bees and ants," *Ecology*, vol. 64, pp. 564–577, 1983.
- [54] K. B. Sim, J. Y. Kim, and D. W. Lee, "Game model based co-evolutionary solution for multiobjective optimization problems," *Int. J. Control, Autom., Syst.*, vol. 2, no. 2, pp. 247–255, 2004.
- [55] N. Srinivas and K. Deb, "Multiobjective optimization using non-dominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1994.
- [56] K. C. Tan, Y. J. Yang, and C. K. Goh, "A distributed cooperative co-evolutionary algorithm for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 527–549, 2006.
- [57] K. C. Tan, C. K. Goh, Y. J. Yang, and T. H. Lee, "Evolving better population distribution and exploration in evolutionary multi-objective optimization," *Eur. J. Oper. Res.*, vol. 171, no. 2, pp. 463–495, 2006.
- [58] K. C. Tan, E. F. Khor, T. H. Lee, and R. Sathikannan, "An evolutionary algorithm with advanced goal and priority specification for multiobjective optimization," *J. Artif. Intell. Res.*, vol. 18, pp. 183–215, 2003.
- [59] K. C. Tan, T. H. Lee, and E. F. Khor, "Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 5, no. 6, pp. 565–588, 2001.
- [60] R. K. Ursem, "Multinational GA optimization techniques in dynamic environments," in *Proc. 2000 Genetic Evol. Comput. Congr.*, 2000, pp. 19–26.
- [61] F. Van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, 2004.
- [62] F. Vavak, K. Jukes, and T. C. Fogarty, "Adaptive combustion balancing in multiple burner boiler using a genetic algorithm with variable range of local search," in *Proc. 7th Int. Conf. Genetic Algorithms*, 1997, pp. 719–726.
- [63] D. A. V. Veldhuizen and G. B. Lamont, "On measuring multiobjective evolutionary algorithm performance," in *Proc. 2000 Congr. Evol. Comput.*, 2000, vol. 1, pp. 204–211.
- [64] G. Venter and R. T. Haftka, "A two species genetic algorithm for designing composite laminates subjected to uncertainty," in *Proc. 37th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conf.*, 1996, pp. 1848–1857.
- [65] M. Wineberg and F. Oppacher, "Enhancing the GA's ability to cope with dynamic environments," in *Proc. 2000 Genetic Evol. Comput. Congr.*, 2000, pp. 3–10.
- [66] S. Y. Zeng, G. Chen, L. Zheng, H. Shi, H. de Garis, L. Ding, and L. Kang, "A dynamic multi-objective evolutionary algorithm based on an orthogonal design," in *Proc. 2006 IEEE Congr. Evol. Comput.*, 2006, pp. 573–580.
- [67] Q. Zhang, A. Zhou, and Y. Jin, "Modelling the regularity in an estimation of distribution algorithm for continuous multiobjective optimization with variable linkages," *IEEE Trans. Evol. Comput.*, accepted for publication.
- [68] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, 2003.
- [69] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Computer Engineering and Networks Lab. (TIK), Swiss Federal Inst. Technol. (ETH), Zurich, Switzerland, Tech. Rep. 103, 2001.
- [70] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

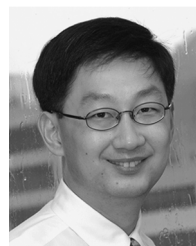


**Chi-Keong Goh** received the B.Eng. and Ph.D. degrees in electrical engineering from the National University of Singapore in 2003 and 2007, respectively.

He joined the Data Storage Institute, Agency for Science, Technology and Research (A\*Star), Singapore, in 2007, as a Research Fellow, where he became a Senior Research Fellow in 2008. His current research interests include evolutionary computation and neural networks, specifically in the design and application of hybrid evolutionary techniques for multiobjective optimization in uncertain

environments. He is the Co-Author of *Evolutionary Multiobjective Optimization in Uncertain Environments: Issues and Algorithms* (Springer-Verlag) to appear in 2008.

Dr. Goh has served as a Reviewer for various international journals such as the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON SYSTEM, MAN, AND CYBERNETICS, PART B, the IEEE TRANSACTIONS ON SYSTEM, MAN, AND CYBERNETICS, PART C, *Neurocomputing*, etc. He has Co-Chaired the Special Sessions on Evolutionary Computation for Expensive Problems and Evolutionary Computation in Uncertain Environments in the IEEE Congress on Evolutionary Computation.



**Kay Chen Tan** is currently an Associate Professor in the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. He has published over 70 journal papers, 100 papers in conference proceedings, and coauthored five books including *Multiobjective Evolutionary Algorithms and Applications* (Springer-Verlag, 2005), *Modern Industrial Automation Software Design* (Wiley, 2006), *Evolutionary Robotics: From Algorithms to Implementations* (World Scientific, 2006), *Neural Networks: Computational Models and Applications* (Springer-Verlag, 2007), and *Evolutionary Multiobjective Optimization in Uncertain Environments: Issues and Algorithms* (Springer-Verlag, expected in 2008). He is actively pursuing research in the field of computational intelligence, with applications to multiobjective optimization, scheduling, design automation, and games.

Dr. Tan received the NUS Outstanding Educator Awards (2004), the NUS Engineering Educator Awards (2002, 2003, and 2005), the NUS Annual Teaching Excellence Awards (2002, 2003, 2004, 2005, and 2006), and the NUS Teaching Awards Honor Roll (2007). He has been invited to be a keynote/invited speaker for many international conferences. He served in the international program committee for over 60 conferences and was involved in the organizing committee for over 20 international conferences, including the General Co-Chair for the IEEE Congress on Evolutionary Computation (2007) in Singapore and the General Co-Chair for the IEEE Symposium on Computational Intelligence in Scheduling (2007) in Hawaii. He is currently a member of Board of Directors in Evolutionary Programming Society, USA. He currently serves as an Associate Editor/Editorial Board member of eight international journals, including IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, *Journal of Scheduling*, *European Journal of Operational Research*, and the *International Journal of Systems Science*.