Coevolutionary Multiobjective Evolutionary Algorithms: Survey of the State-of-the-Art

Luis Miguel Antonio and Carlos A. Coello Coello, *Fellow, IEEE*

Abstract—In the last 20 years, evolutionary algorithms (EAs) have shown to be an effective method to solve multiobjective optimization problems (MOPs). Due to their population-based nature, multiobjective EAs (MOEAs) are able to generate a set of tradeoff solutions (called nondominated solutions) in a single algorithmic execution instead of having to perform a series of independent executions, as normally done with mathematical programming techniques. Additionally, MOEAs can be successfully applied to problems with difficult features such as multifrontality, discontinuity and disjoint feasible regions, among others. On the other hand, coevolutionary algorithms (CAs) are extensions of traditional EAs which have become subject of numerous studies in the last few years, particularly for dealing with large-scale global optimization problems. CAs have also been applied to the solution of MOPs, motivating the development of new algorithmic and analytical formulations that have advanced the state-of-the-art in CAs research, while simultaneously opening a new research path within MOEAs. This paper presents a critical review of the most representative coevolutionary MOEAs (CMOEAs) that have been reported in the specialized literature. This survey includes a taxonomy of approaches together with a brief description of their main features. In the final part of this paper, we also identify what we believe to be promising areas of future research in the field of CMOEAs.

Index Terms—Coevolutionary algorithms (CAs), competitive coevolution, cooperative coevolution (CC), multiobjective optimization.

I. INTRODUCTION

COEVOLUTIONARY algorithms (CAs) are natural extensions of traditional evolutionary algorithms (EAs). Such extensions consist mainly in the way in which the fitness function is evaluated for each of the members of a coevolutionary system. In CAs, the fitness value of each individual is assigned based on its interactions with members from other species. Coevolution is then a reciprocal evolutionary exchange between species that have interaction with each other. CAs arise from the biological observation which shows that coevolving a number of species, defined as groups of individuals with similar phenotype, is more in line with the reality than just evolving a group of individuals which represent a single species [1]. So, instead of evolving one population

(spatially or globally distributed) of similar individuals which represent a global solution, it is more suitable to coevolve subpopulations of individuals representing different specific parts of the global solution. A coevolutionary search involves the use of multiple groups of similar individuals (species) to be the representation of a solution to a multiobjective optimization problem (MOP). Species can either compete or cooperate during the search process. For this reason, coevolutionary models have been historically classified as *competitive* or *cooperative*. In the cooperative coevolutionary framework, individuals are rewarded when they have a good performance working with other individuals and they are punished when they have a poor performance [2]. Under this model, each subpopulation represents a piece of a larger problem, and it is the labor of those subpopulations to gradually evolve (i.e., improve) more competent pieces for the entire problem. On the other hand, in the competitive coevolutionary framework, individuals are rewarded at the expense of those with which they interact [3]. An example can be found in the predator-prey model, where individuals in one population represent a device or model, while individuals in another population are the representation of inputs for such device. The objective of the first population is to evolve better instances of the devices to manage the input, while the objective of the second population is to evolve increasingly difficult inputs for the population of devices.

In recent years, there has been an increasing interest for adopting the coevolutionary framework as a technique to solve MOPs [4]–[6]. In this paper, the main focus is on how coevolution is integrated into multiobjective EAs (MOEAs).

The remainder of this paper is organized in the next way. To make this paper self-contained, we provide some basic concepts related to multiobjective optimization in Section II. Section III presents an introduction to coevolutionary MOEAs. Sections IV–VI are devoted to present a comprehensive review of CAs for multiobjective optimization including both cooperative and competitive approaches. Section VII describes some of the pathologies that the coevolutionary framework has to deal with. In Section VIII, some possible future research paths in this area are briefly discussed. Finally, the conclusion is drawn in Section IX.

The authors are with the Department of Computer Science, Evolutionary Computation Group, CINVESTAV-IPN, Mexico City 07360, Mexico (e-mail: lmiguel@computacion.cs.cinvestav.mx; ccoello@cs.cinvestav.mx).

II. PROBLEM STATEMENT

We are interested in solving problems of the type[1]

$$\text{minimize } \vec{f}(\vec{x}) := [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})] \tag{1}$$

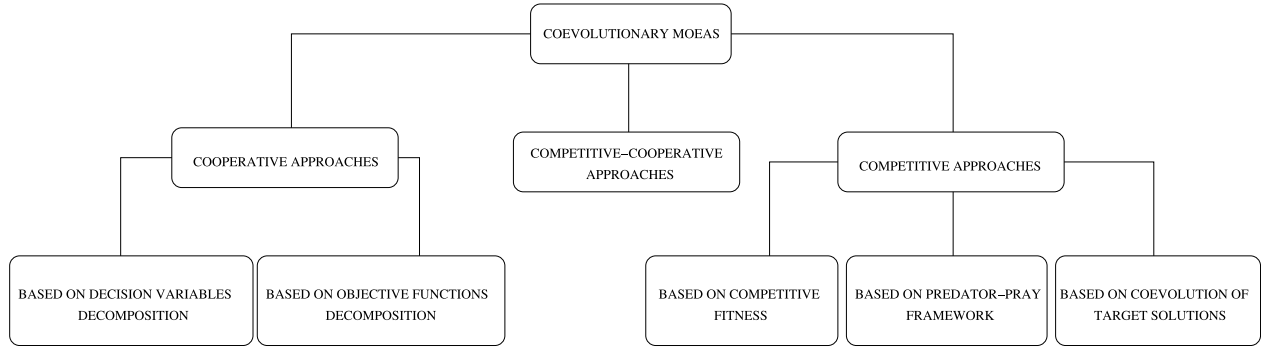[1]Without loss of generality, we will assume only minimization problems.

Fig. 1. Taxonomy for CMOEAs based on the ways coevolution has been adopted.

subject to

$$g_i(\vec{x}) \leq 0 \quad i = 1, 2, \ldots, m \tag{2}$$

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \ldots, p \tag{3}$$

where $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \ldots, k$ are the objective functions and $g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \ldots, m, j = 1, \ldots, p$ are the constraint functions of the problem.

Definition 1: Given two vectors $\vec{u}, \vec{v} \in \mathbb{R}^k$, we say that $\vec{u} \leq \vec{v}$ if $u_i \leq v_i$ for $i = 1, \ldots, k$, and that $\vec{u} < \vec{v}$ if $\vec{u} \leq \vec{v}$ and $\vec{u} \neq \vec{v}$.

Definition 2: Given two vectors $\vec{u}, \vec{v} \in \mathbb{R}^k$, we say that $\vec{u}$ dominates $\vec{v}$ (denoted by $\vec{u} \prec \vec{v}$) iff $\vec{u} < \vec{v}$.

Definition 3: We say that a vector of decision variables $\vec{x}^* \in \mathcal{F}$ ($\mathcal{F}$ is the feasible region) is *Pareto optimum* if there does not exist another $\vec{x} \in \mathcal{F}$ such that $\vec{f}(\vec{x}) \prec \vec{f}(x^*)$.

Definition 4: The *Pareto optimal set* $\mathcal{P}^*$ is defined by

$$\mathcal{P}^* = \{\vec{x} \in \mathcal{F} | \vec{x} \text{ is Pareto optimum}\}.$$

The vectors $\vec{x}^*$ corresponding to the solutions included in the Pareto optimal set are called *nondominated*.

Definition 5: The *Pareto front* $\mathcal{PF}^*$ is defined by

$$\mathcal{PF}^* = \left\{\vec{f}(\vec{x}) \in \mathbb{R}^n | \vec{x} \in \mathcal{P}^*\right\}.$$

We thus wish to determine the Pareto optimal set from the set $\mathcal{F}$ of all the decision variable vectors that satisfy (2) and (3).

III. COEVOLUTIONARY MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS

Within nature, coevolution is the process of reciprocal genetic change in one group of individuals (species) in response to another. In other words, coevolution is a reciprocal evolutionary change between species that interact with each other. The term *coevolution* arises from a study about the interaction between plants and butterflies conducted by Ehrlich and Raven [7] in which the coevolutionary responses of ecologically intimate organisms and community evolution in general were observed. The relation among the individuals of two different species *S*1 and *S*2 can be defined taking into account all their possible kinds of interactions. Such interaction can be positive or negative, depending on the consequences that the interaction produces on individuals of each

of the species. Coevolution is then used as the biological process responsible for diversity, speciation, arms races and open-ended evolution. The main issue in CAs is that the way fitness of an individual from a species is computed depends on the individuals of a different species. According to the way this fitness is computed, there are two main kinds of CAs in the evolutionary computation literature.

1) *Those Based on Competitive Interactions:* In this kind of CAs, fitness of an individual is the result of a series of encounters with other individuals from other species [3].
2) *Those Based on Cooperative Interactions:* In this sort of algorithms, fitness of an individual is given by the performance of such individual when collaborating with individuals from other species [2], [8].

Competition and cooperation between groups species in nature has inspired researchers to incorporate coevolutionary dynamics into MOEAs. Evolutionary computation researchers have developed many coevolutionary approaches, in which two or more species (i.e., populations) that relate to each other are applied to deal with MOPs, using one of the previously indicated schemes. Also, in most cases, these species evolve independently by means of an EA and interaction occurs when individuals need to be evaluated. Fig. 1 presents a taxonomy for CMOEAs, based on the main ways in which coevolution has been applied to MOEAs.

In the next sections, we describe in more detail each of the ways in which coevolution has been applied to MOEAs, as well as the most representative coevolutionary MOEAs for each of the categories previously indicated.

IV. COOPERATIVE COEVOLUTIONARY MOEAs

The pioneering work of Potter and De Jong [8] started the research on cooperative CEAs in 1994 by introducing the first framework of cooperative coevolution (CC) adopted within EAs with the so-called cooperative coevolutionary genetic algorithm (CCGA), which was first applied to global optimization and later to neural network learning [2]. Potter and De Jong's [8] framework adopts a divide-and-conquer technique to divide the vector of decision variables of a problem into subsets of smaller size, so that each of these subsets is optimized using a population with a separate optimizer. The core idea is to split a high-dimensional problem into n subcomponents of lower dimension and then

evolve each of these subcomponents cooperatively. Here, evolution occurs through cycles, where a *cycle* consists of one complete evolution of all subcomponents. For the problem decomposition, Potter and De Jong [8] took each decision variable of the problem as a subcomponent. In CCGA, each species has a population containing individuals which represent a component of the whole solution, and evolution of these populations occurs almost independently, in tandem with one another, interacting only to obtain the individuals' fitness. The decomposition process can be static, i.e., the divisions for the separate components are decided *a priori* and are never altered along the process, or dynamic, where components may be added or removed to the species as the search progresses [9]. After CCGA, there were many other *cooperative coevolutionary* approaches, most of which were used for large-scale global optimization [10]–[14].

In general, the most common cooperative coevolutionary framework for global (single-objective) optimization using EAs can be summarized as follows. First, the vector of decision variables of the problem is divided into S subcomponents of lower dimension. Then, the iterations are started and the optimization of the ith species, using a specific EA for a predefined number of generations, is performed. This is then applied through all of the species. The whole process is repeated for a number of *cycles* or until a stopping condition is reached. The coevolutionary effect in the CCGA is produced by a cooperation among all species.

Because this cooperative coevolutionary approach can be relatively easily extended to the multiobjective optimization context, there have been several proposals which incorporate it with the aim of improving the performance of an MOEA. As the nature of MOPs changes, the decomposition can be made not only in decision variable space, but also in objective function space or in a mixture of both. Next, we will review the main cooperative coevolutionary MOEAs currently available.

A. Cooperative CMOEAs Based on Decision Variables Decomposition

In this type of CMOEAs, the MOP is decomposed along the search space of the problem. So, every decision variable of the problem is assigned to a species population and each species population optimizes one or more decision variables at the same time.

In other words, each population has individuals which represent a particular part (in decision variables space) of the MOP. Thereafter, every member from each population is needed in order to assemble a full solution to the problem. The evaluation of individuals from a particular species' population is then performed by making the individual collaborate with members from the other species. A graphical description of the decision variables decomposition is presented in Fig. 2. The drawback of this kind of problem decomposition approach is that information about the ideal number of components or the optimal way to assign them is, in most cases, not known *a priori*. Also, many problems present highly complex interdependencies and the decomposition becomes
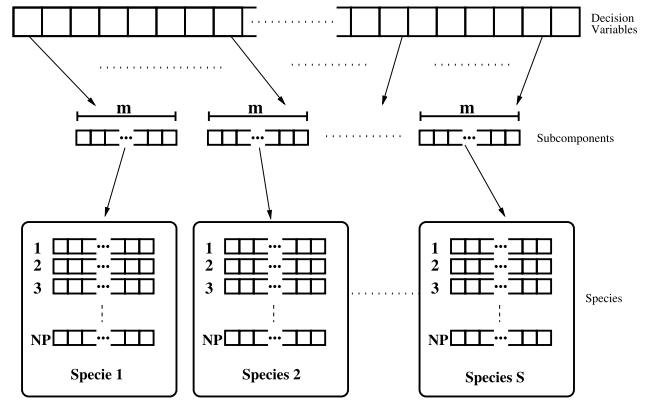

Fig. 2.   Graphical representation of the species creation.

harder to perform. Next, we present the most representative approaches within this category, but more examples can be found in [6] and [15]–[29].

1) Indicator-Based Cooperative CMOEAs: Antonio and Coello Coello [18] proposed the so-called indicator-based cooperative CMOEA (IBCCMOEA). This algorithm uses the CCGA [8] framework (adapted to multiobjective optimization) and differential evolution (DE) as the main multiobjective optimizer. IBCCMOEA includes the following processes: it starts with a division of the decision variables space where the vector $\vec{x}$ representing the set of $D \in \mathbb{N}$ decision variables is divided into $d \in \mathbb{N}$ smaller vectors of equal size. Each subcomponent is created using a random assignment of decision variables, so that the probability of grouping interacting variables into the same species in nonseparable problems is increased. Also, d species having NP individual each, are created. Then, each of these d species are assigned a D/d number of different decision variables in a random way. In other words, each subpopulation is in charge of a subcomponent from D. After the subpopulations are created, a random initialization of all the species' populations is performed. Thereafter, IBCCMOEA performs the evolution of each of the subpopulations for a given number of *generations*, throughout a certain number of *cycles*. The evolution of each species is performed using DE operators combined with some nondominated sorting genetic algorithm (NSGA)-II techniques. Such process continues for a certain (predefined) number of cycles and generations or until a stopping criterion is fulfilled. The outcome of the algorithm is the set of solutions that are globally nondominated (i.e., with respect to all of the nondominated solutions from all the species' populations). The way cooperation between subpopulations is performed, in order to compose a complete solution to be evaluated, is described next. At the start of the algorithm, when no knowledge has been acquired, collaborations are performed in a random way, by selecting a random individual from each species and assembling a complete set of solutions to be evaluated in the set of objective functions of the MOP. Thereafter, the results from the evaluations are given back to each individual. After the first generation, collaborations will take place when the resulting offspring subpopulations need to be evaluated. Such evaluation is performed

by joining the individual under evaluation with individuals from the other species which have the best contribution to the hypervolume indicator. At the end, the nondominated solutions from the set of nondominated solutions of each species' populations is computed and given as the final result. IBCCMOEA was benchmarked with respect to a variation of itself which adopts a Pareto-based selection for collaboration of species, called Pareto-based CCMOEA (PBCCMOEA). The authors adopted the DTLZ test suite [30] for their experiments, with MOPs of three objective functions and 12 decision variables. IBCCMOEA outperformed PBCCMOEA in all problems, showing faster convergence. The main drawback of this approach is that, as it adopts the hypervolume indicator as a selection mechanism, it would be unaffordable to solve MOPs with more than three objectives, since even in 3-D problems, the executions took more time than the ones for PBCCMOEA.

2) Multiobjective Cooperative CMOEA for Hyperspectral Sparse Unmixing: Gong *et al.* [31] proposed a cooperative CMOEA for the solution of hyperspectral sparse unmixing problems. In this approach, its authors divide the vector of decision variables into different subsets, assigned to different species to be optimized. For this sake, they adopted a grouping strategy based on sparsity and a random grouping scheme to deal with nonseparable problems. Also, the nonzero elements of every solution in the population are selected to create a nonzero species. In this case, an improved uniform mutation operator based on sparsity and a modified arithmetical crossover operator are adopted to perform the optimization of the solutions. As the knee solution is considered as the best solution for the sparse unmixing problem, such solution is used as an effective leader to guide each species toward a certain direction and to speed up the convergence of the approach. They achieved this by performing a cooperative interaction between each subcomponent in a species and the knee solution, i.e., by taking the individual under evaluation and combining it with the kneeă solution to form a complete solution, in order to compute the fitness of the complete solution and assign it back to the individual.

In order to assess the performance of this approach, its authors compared it against a version of the algorithm which does not make use of the cooperative coevolutionary strategy when solving a multiobjective sparse unmixing problem. They adopted the two set coverage (SC) indicator to assess the diversity in the population. The cooperative coevolutionary strategy (along the nonzeros grouping scheme) was found to be able to speed up the convergence of the algorithm and allowed it to have a better diversity of the population than its version without the cooperative coevolutionary approach. This approach was shown to be sensitive to the M value of the modified nonuniform mutation operator: in cases when the M value was small, the algorithm suffered of premature convergence. Also, it is worth noting that no performance indicator was adopted to assess spread of the solutions. Additionally, it is clear that the interaction with the knee solution may prevent the generation of solutions in some parts of the Pareto front.
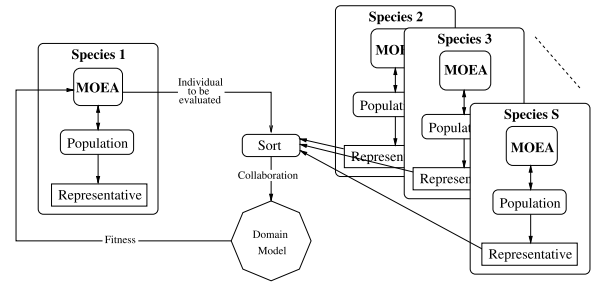

Fig. 3. Cooperative collaboration based on objective function decomposition architecture. Here, each species is representing a single objective function. In order to obtain its fitness value, a solution from species 1 is joined with the representatives of each species and then is evaluated under the domain model of each species, i.e., over their corresponding objective function.

B. Cooperative CMOEAs Based on Objective Functions Decomposition

In this case, the MOP is decomposed along the objective functions of the problem. Each objective function of the problem is assigned to a certain species' population and all populations cooperate to approximate the whole Pareto front. Each species has a population formed by individuals that represent a solution to the MOP. Individuals from each species compute all the objective functions evaluations in the same way as traditional MOEAs. However, the main difference is that the fitness value of an individual in a certain species is given only by its corresponding objective function. Hence, individuals are guided by their specific objective function, in order to search in different regions of the Pareto front. Fig. 3 shows the way collaboration is performed in order to create the coevolutionary effect. Next, we present the most representative approaches within this category, but more examples can be found in [32]–[34].

1) Cooperative Differential Evolution With Multiple Populations for Multiple Objectives: A cooperative CMOEA which uses multiple populations for multiple objectives (MPMOs) was presented in [34]. MPMO creates multiple species' populations to deal with the several objectives of the MOP, in such a way that each species represents only one objective function and all species cooperate to approximate the whole Pareto front. MPMO is considered a cooperative coevolutionary approach because it uses multiple species (populations) to solve problems in a cooperative way. At each generation, individuals from each species are evaluated using all the objective functions as done in traditional MOEAs. Nevertheless, once evaluated, the fitness value of an individual from the sth species is assigned by the sth objective function of the MOP, where $1 \leq s \leq S$ and S is the number of species and objective functions of the MOP. This way, individuals search along their respective objective function, assigned to their corresponding species, so that making use of all the S species MPMO will search different regions of the Pareto front at the same time. Since every species is assigned a different objective function, an information-sharing mechanism is used to share their search knowledge and communicate it among each other.

The authors of this approach adopted DE as their optimizer for each species' population and created a coevolutionary

multiobjective DE (CMODE) algorithm making use of the MPMO scheme to solve MOPs. CMODE adopts an external archive as the instance of the information-sharing mechanism from MPMO, with two extra techniques to enhance the approach's performance. One is used to modify the way in which the mutation of individuals (by DE) is performed. Such modification consists on the use of the information obtained from the external archive. The shared archive stores the nondominated solutions found so far by different species and is updated at each cycle of CMODE. The DE operation applied to each species' individual is performed by taking into account its personal experience as well as the species' global experience in addition to the experience taken from the external archive of nondominated solutions. In this way, all the species can share their knowledge to speed up convergence toward the whole Pareto front. The second approach is to adopt an elitist learning strategy, which updates the archive in order to produce an appropriate diversity so that CMODE is able to deal with MOPs having multiple local Pareto fronts or with complicated Pareto sets. For this sake, a crowding distance estimator is adopted for the archive truncation process. Also, the external archive is improved further by DE to provide a better guidance information to each species. CMODE is benchmarked against some state-of-the-art MOEAs (see [34] for further details about the test functions adopted) and results are compared using the inverted generational distance (GD) [35] indicator and the hypervolume. Experimental results show how CMODE outperforms, in most cases, the other algorithms on the UF [36] and WFG [37] test problems.

One of the advantages of the CMODE algorithm is that each species is in charge of only one objective to prevent the difficulty of performing fitness assignment and at the same time to obtain a benefit from having each species population with a conventional MODE for optimizing a single objective. Also, as an external archive is adopted to store the nondominated solutions found so far in the process, each species can communicate with the others through this mechanism and, therefore, they can use the knowledge obtained by the other species. The algorithm is able to deal with MOPs having multiple local fronts, due to the adopted elitist strategy in the archive update mechanism. Finally, CMODE can be adapted to solve MOPs with more than three objective functions. The main drawback of CMODE is that, since each species optimizes only one objective function, this might contribute to the creation of more solutions on the extreme values of each objective, resulting in a poor approximation of the whole Pareto front. Also, it has a poor performance in multifrontal problems. Another observation has to be made regarding the selection method that the authors adopted when using archive information, since a random selection is misleading and can lead to an undesired waste of resources (by means of function evaluations).

2) Cooperative Coevolution Based on Nash Equilibrium: The Nash cooperative coevolutionary (NashCC) algorithm [6] is a novel cooperative coevolutionary scheme based on the concept of Nash equilibrium. NashCC is meant to alleviate some of the pathologies of the cooperative coevolutionary framework. The description of the algorithm is presented next. In the first part of the process, NashCC separates the vector

of decision variables into m smaller components, where m is equal to the number of objective functions in the MOP. On the other hand, m species are created, each one composed by NP individuals, and these m species are assigned to some decision variables using the random grouping approach. That is, to say, to each species it corresponds a subcomponent which has been already computed. Therefore, every species will be in charge of D/m decision variables, where D is the total number of decision variables of the MOP. Also, each species will be related to a specific objective function of the MOP, which means that we will have as many species as objective functions and each of the individuals from each of the species will perform their fitness computation based on this as described next. After the species' populations are created, NashCC performs a random initialization of all the solutions for each species' population. Thereafter, the *generations*, required for the evolution of each of the subpopulations, take place for a given number of *cycles*. When the algorithm finishes, the nondominated solutions from all the species form the solution. The collaboration procedure among species takes place as follows: when NashCC starts, no knowledge about the individuals exists and, therefore, in order to assemble a solution, one random solutions from the other species is chosen and evaluated in its objective functions. Thereafter, the outcome from the evaluation is saved within the individual being evaluated. In any other case, individuals are evaluated by forming collaborations with individuals from other species which are in a Nash equilibrium [38], according to their actual fitness values.

For this sake, NashCC elaborates combinations from the function belonging to the species of the individual being evaluated and the possible values given by the rest of the other species' populations, taking each objective function as an strategy. NashCC builds a utility function from each species' perspective, in order to create a noncooperative game from the problem. In such a game, each player is represented by a species, and its aim is to optimize the objective function which corresponds to each species. For this matter, SPEA2's fitness assignment strategy [39] is adopted within each species to create the noncooperative game. Having an individual which belongs to the sth species, NashCC computes all the possible combinations of the objective functions from the current species under evolution with the values of the objective functions of the others. Having formed these composed vectors of objective functions, NashCC computes the fitness of each of these combinations according to the current members of each species. Therefore, individuals which provide a Nash equilibrium in the created game are selected from the species to collaborate.

NashCC showed a very good performance when dealing with the ZDT [40] and DTLZ [30] test problems. Its authors analyzed the convergence rate of NashCC with respect to that of NSCCGA [19] and GCEA [41]. The results showed that NashCC was able to outperform the other two CCAs and that the collaboration framework based on Nash equilibrium had a great impact on CCAs' performance.

The main drawback of this approach is the computational time that it requires (it is exponential with respect to

TABLE I
SUMMARY OF COOPERATIVE CMOEAS

Reference	Algorithm	Type of decomposition	Subpopulations	Representation	Optimizer
[18]	IBCCMOEA	Decision variables space	Groups of decision variables	Real	MODE
[31]	CCMOEA-HSU	Decision variables space	Groups of decision variables	Real	Genetic Algorithm
[34]	CMODE	Objective space	One per objective function	Real	Differential Evolution
[6]	NashCC	Objective and decision variables space	One per objective function	Real	NSGA-II
[42]	PICEA-w	Objective space	Two populations	Real	Evolution Strategies

the number of strategies), which makes it unaffordable for many-objective optimization.

3) Preference-Inspired Coevolutionary Algorithm Using Weight Vectors: A preference-inspired coevolutionary algorithm using weight vectors (PICEA-w) is presented in [42]. PICEA-w was created with the objective of alleviating the difficulties that decomposition-based approaches have when dealing with MOPs having complex Pareto front geometries. This approach adopts a methodology that decomposes an MOP into a set of single objective subproblems defined by means of several scalarizing functions with different weights, where each weight vector is used as a search direction to define a scalar function. In PICEA-w, weights are adaptively modified by co-evolving them with candidate solutions along the search process, in order to construct suitable weights in an adaptive manner during the optimization process and use them to guide candidate solutions toward the Pareto optimal front, in an effective manner. In PICEA-w, candidate solutions are ranked by each of the weighted scalarizing functions and, therefore, a ranking matrix is created. Then, the fitness of candidate solutions is computed based on such ranking matrix. Thereafter, weight vectors are coevolved with the candidate solutions to create an optimal distribution, and these at the same time work to create a balance between exploration and exploitation. For each selected solution, a weight which ranks this solution as the best is selected. Such weight must maintain the convergence and exploitation as well as its distance from the solution.

PICEA-w is implemented within an elitist $(\mu + \lambda)$-ES (evolution strategy) framework. Populations of candidate solutions and weight vectors, S and W, respectively, (of size N and N_w), are evolved for a fixed number of generations. At each generation, parents are subjected to genetic variation operators to produce N offspring (S_c). At the same time, N_w new weight vectors (W_c), are randomly generated. Thereafter, $S \bigcup S_c$ and $W \bigcup W_c$ are sorted according to fitness and a truncation selection procedure is applied to select the best N solutions and N_w weight vectors are the new elements of PICEA-w for the next generation. Additionally, an offline archive is adopted to store all the nondominated solutions found during the search. In order to obtain a good distribution of solutions, SPEA2's [39] clustering technique is applied after the optimization process has been applied. PICEA-w was benchmarked against four variations of its framework (see [42] for further details) using eight test problems constructed by applying different shape functions provided in the WFG toolkit to WFG4 [37]. The WFG parameters k (position parameter) and l (distance parameter) were set to 18 and 14, respectively, creating MOPs

with $n = k + l = 32$ decision variables. These problems were adopted with 2, 4, and 7 objective functions. PICEA-w showed to be less sensitive to the problem geometry, and outperformed other leading decomposition-based algorithms on problems with more than four objectives (many-objective instances). Moreover, it was found that when guiding candidate solutions toward the Pareto optimal front, the weights adopted by PICEA-w also evolve toward the optimal distribution when adopting a coevolutionary strategy.

One of the observations about this approach is that its experimental validation did not include other MOP difficulties apart from geometrical features. Additionally, a comparison with other state-of-the-art MOEAs (e.g., PICEA-g [43]) is required to have a more general idea of the actual performance of the proposed approach.

Table I summarizes the cooperative CMOEAs we have just described.

V. COMPETITIVE COEVOLUTION

The computational study of CAs conducted by Hillis [44] gave birth to several competitive CAs. His main motivation was to take advantage of some coevolutionary techniques for evolving sorting networks and data sets adopting a predator-prey relationship interaction. Hillis [44] used two independent species' populations: one formed by a set of sorting networks (the hosts) and another whose components were test cases (the parasites). Fitness was assigned to each sorting network according to its capability to solve the given test cases. Meanwhile, each test case was given a fitness value according to the number of times it was not correctly sorted by the networks instances. In this way, both populations evolved simultaneously and interacted through the fitness function evaluations.

Under this scheme, the coevolutionary effect among subpopulations is the result of a competition for survival by the individuals. Competitive coevolution can lead to an arms race in which populations' interaction makes them to improve their performance and capacity of solving more complex problems. In other words, competitive coevolution consists of entities that compete among them for dominance in the population. Next, we will review the main approaches of competitive coevolution that have been applied to MOEAs.

A. Competitive CMOEAs Based on the Predator-Prey Framework

In the predator-prey coevolutionary framework, a prey represents a decision variables space instance and a predator is
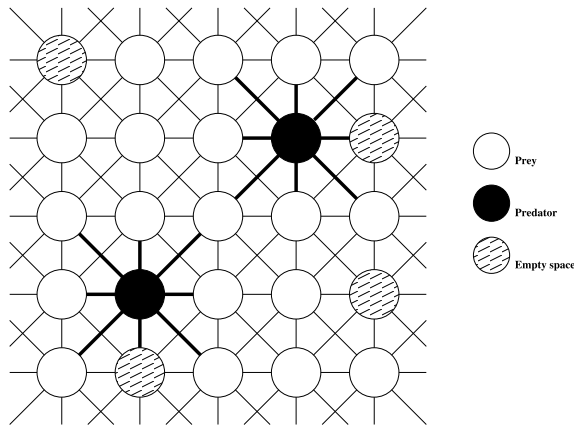
Fig. 4. Predators and prey distributed in a 2-D lattice. For the predator in the top-right of the figure, we show the possible movements that it can perform. It is worth noting that prey can also move in the directions of their corresponding edges.

the entity that deals with the objective functions. The idea is to imitate the way in which a predator (which encodes the decision variables of the problem) hunts a prey (where a prey represents an objective function). The weakest prey (i.e., those solutions with the worst objective function values) are eliminated by the predator. Fig. 4 depicts this procedure. Next, we present the most representative algorithms within this class. For further details and examples please refer to [45]–[54].

1) Predator-Prey Biogeography-Based Optimization: In [51], a predator-prey approach based on biogeography optimization is developed to optimize the construction parameters of a brushless dc wheel motor (a multiobjective constrained problem with two objective functions: 1) the efficiency and 2) the mass of the motor). In this paper, the predator-prey coevolutionary framework is used to improve the diversity of the solutions as well as to deal with local optimum traps. For this sake, in predator-prey biogeography-based optimization (PPBBO) the predators are modeled based on the worst individuals of the current set of solutions to the problem. Also, a fugue mechanism for the prey is proposed, in order to allow prey to maintain a distance from the predator. The authors of this approach consider one type of predator for each objective function, in order to avoid the use of a Pareto-based measure. Therefore, each predator will be hunting individuals with the worst values (prey) in their respective objective function. In other words, predators eliminate the less fit prey in their neighborhood, which is analog to eliminating the worst value of the objective function on which the predator has its sight. With this mechanism, the other individuals will try to run away from those having the worst values, in order to avoid the predators. The biogeography-based mechanism adopted by PPBBO consists of a migration operator used to share information among the problem solutions. In this context, the problem solutions are identified as habitats, and therefore each one has particular aspects related to their suitability.

PPBBO is used to optimize three variants of the brushless dc wheel motor problem. A comparison between PPBBO and the biogeography-based optimization (BBO) standalone algorithm is performed. The authors conclude that adding a predator-prey scheme to the original BBO allows the algorithm to have a better diversity of the solutions, since the predator-prey mechanism avoids premature convergence to a point in the search space, improving both exploration capability and convergence of the approach toward the true Pareto front.

A limitation of this paper is that it adopts a fairly limited number of test problems and it lacks the use of the performance measures that are traditionally adopted to assess performance of MOEAs.

2) Modified Predator-Prey Algorithm: A modified predator-prey (MPP) algorithm is presented in [52], where a computationally inexpensive MOEA able to handle both linear/nonlinear equality and inequality constraints is proposed. This approach is inspired by the predator-prey behavior and proposes the use of a dynamic spatial structure for the predator-prey instantiation. For this sake, MPP uses a 2-D lattice where prey and predators interact. In order to erase any boundary conditions, the 2-D lattice structure enfolds its boundaries to their opposite edges. In MPP, the number of prey is large and the number of predators is smaller in the initial populations. Each prey represents a possible solution and each predator is meant to check prey in their vicinity to look for the weakest (i.e., the ones with the worst objective values) and delete them. Its basic procedure is the following.

1) Perform a random initialization of the prey.
2) Assign a place (vertex) to each of the prey on an undirected connected graph.
3) Compute the associated model for each prey with a weighted sum value of the objectives.
4) Assign a place (vertex) to each of the predators on the aforementioned undirected connected graph.
5) Evaluate prey and select the ones with the worst objective value, according to its neighboring predator, and delete them.
6) Create a new offspring from the strongest local prey around the previously deleted prey.
7) Change the location of predators to a neighboring node.
8) Copy the nondominated solutions in the prey population to a external archive of best solutions.
9) Incorporate certain number of randomly selected solutions from the external archive into the main population (prey in the toroidal space) at the expense of some of the dominated solutions (dominated by at least one other prey).
10) Repeat the process until the stopping condition is fulfilled.

For the case of MOPs with constraints, MPP creates an extra objective function by adding up the constraints using a tolerance factor for the equality constraints. For the generation of new solutions, the blend crossover and nonuniform mutation are adopted. MPP was validated using several test problems from the state-of-the-art (see [52] for further details about these test problems), having constrained and unconstrained instances. MPP showed to be very competitive in producing reliable solutions for the adopted unconstrained MOPs. Also, in the case of constrained MOPs, MPP was capable of driving solutions into the feasible domain, with the use of a reasonable number of extra function evaluations required by the constraint-handling technique adopted.
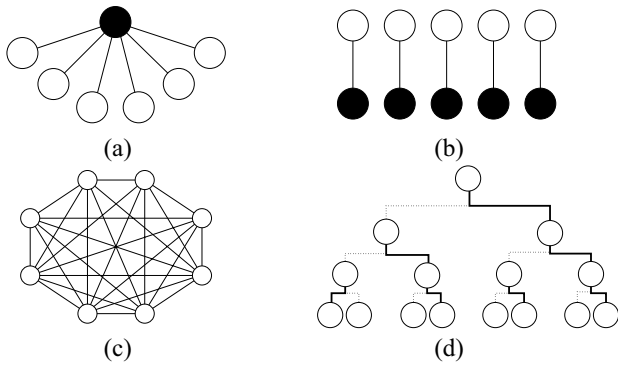
Fig. 5. Types of competitive fitness schemes. (a) All versus all. (b) Bipartite. (c) All versus best. (d) Tournament.

It was observed that a higher number of predators led to premature convergence due to excessive selection pressure (compared to evolutionary pressure), whereas a lower number of predators demanded a higher number of function evaluations, without the warranty of any improvement in objective function values. The main drawback of this approach is that, as MPP employs the concept of weighted sum of objectives without any normalization of the objectives, in some cases it obtains solutions with a relatively poor distribution of Pareto solutions.

B. CMOEAs Based on Competitive Fitness

CMOEAs based on competitive fitness differ from standard MOEAs in the way they are evaluated. When using competitive fitness, a special fitness function (where dependence between species is taken into account) is adopted. Such dependence can be applied in different ways, for example, making use of a full competition approach, where an all-against-all competition scheme is adopted. Another type of interaction is the bipartite competition, in which a one-versus-one or a one-versus-many competitions scheme may be adopted. Also, a binary tournament can be used, with tournaments of single elimination (i.e., the least fit of the two competing individuals is eliminated). Another technique is the elitist competition, that performs an all-versus-best competition [55]. In these types of competitions, the fitness values of individuals are compared with respect to those of the others. The individuals with the best fitness values are the ones who *win* the competition. Fig. 5 shows a graphical representations of four types of competitive fitness schemes. Next, we present some examples of this sort of approach. Interested readers may refer to [25], [43], and [56]–[63] for more details and examples of this sort of approach.

1) Competitive Coevolutionary Algorithm for Robust Multiobjective Optimization: A coevolutionary multiobjective optimization scheme, based on competitive fitness functions, was proposed by Meneghini *et al.* [63]. In this paper, a competitive coevolutionary approach for robust MOO is introduced, without the incorporation of any robustness measures neither in the objective function nor in the constraints. In the proposed approach, two populations compete in the environment. One population represents candidate solutions and is

meant to minimize the objectives. The second population represents uncertainties and is meant to maximize the objectives in a worst case scenario. The proposed approach is a coevolutionary version of MOEA/D called coevolutionary robust MOEA/D (C-RMOEA/D). In C-RMOEA/D, the first population of individuals, representing candidate solutions, evolves toward the minimization of the *g* scalar subproblems (created by the MOEA/D scheme) given the values of perturbations in the second population. Conversely, the second population, representing perturbation values (defined by the decision maker), evolves toward the maximization of the *g* scalar subproblems given the values of the variables. As can be seen, both populations are in conflict or competition with one another, i.e., they compete in the environment defined by the robust MOP. Both populations have the same size and for each solution in the first population, there is another solution which competes against it in the second population.

The populations are optimized by MOEA/D using a coevolutionary competitive fitness process and by repeating this process (until a certain stopping criteria is met), C-RMOEA/D is able to find an estimate of the robust Pareto front, i.e., an estimate that is good even considering the worst case scenario of perturbations for each specific solution. C-RMOEA/D was validated adopting five MOPs from the state-of-the-art— ZDT1, ZDT2, TP2, TP2, and TP4 (see [63] for further details about these test problems). C-RMOEA/D was able to converge to an estimate of the robust Pareto front in all cases. One drawback of this approach is that since two populations are considered in the coevolutionary method, the number of function evaluations is duplicated in comparison to the use of a single-population MOEA.

2) Competitive Coevolution With K-Random Opponents (SPEA2-CE): Tan *et al.* [57] presented a competitive coevolutionary MOEA with a *K*-random opponents strategy in their strength Pareto EA 2 with competitive coevolution (SPEA2-CE). This CMOEA adopts a framework which is very similar to the one used by SPEA2 [39] but adds two new mechanisms to it. One of these mechanisms is used to manage the competitive selection scheme and the other one is adopted for the assignment rewarding. After computing the raw fitness of every individual, an opponent selection mechanism is applied. This mechanism selects in a random way individuals to compete, using a *K*-random opponents algorithm (KROA). This algorithm tries to avoid competition between similar opponents and forbids self-play. KROA is applied with values of 10 up to 90. After KROA is applied, individuals compete against all possible opponents. While a tournament takes place, the rewarding value is computed based on the reward function. This value is then summed up to form the final fitness of individuals, using a reward assignment technique. Individuals compete among them by means of Pareto dominance. Since competitive coevolution is based on the survival of the fittest's principle, when the opponent dominates the nonopponent solution, the opponent individual replaces the nonopponent individual. This competition is performed adopting an elimination tournament of pairs, where all individuals are randomly coupled to compete. Therefore, in each encounter, individuals who lose are eliminated and only the winners survive and
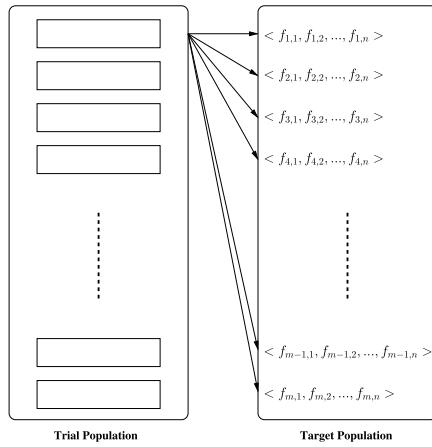
Fig. 6. Overview of the coevolution of target solutions. The trial population is the set of possible solutions and the target populations is the set of TOVs with the desirable values.

conform the population of the next level. Thereafter, the winners are paired to compete against each other and the process is continued until a single winner is left. Then, the archive is updated with the nondominated solutions, and solutions that are dominated by the new ones are removed. Finally, individuals from the archive are selected to breed using simulated binary crossover and polynomial-based mutation [64]. SPEA2-CE was benchmarked with respect to SPEA2 using the DTLZ test problems. Results showed that SPEA2-CE had a better performance, in terms of convergence. Furthermore, the set of nondominated solutions obtained by SPEA2-CE dominated the solutions found by SPEA2. However, SPEA2 had better spacing (SP) values (i.e., it produced better distributed Pareto fronts).

One missing study in this paper is the analysis of the impact that the number of objective functions has on the performance of this approach. Also, it is necessary to compare it with respect to other coevolutionary approaches in order to get more insights about its main advantages and limitations.

C. CMOEAs Based on Coevolution of Target Solutions

The principal idea of this class of approaches is to perform a competition between two populations: one containing possible solutions to the problem and another one containing certain desirable values that we aim to reach for each of the objectives of the problem. Next, we describe an approach that adopts this kind of coevolutionary scheme and show a graphical representation of this idea in Fig. 6.

1) Coevolutionary Genetic Algorithm: A coevolutionary genetic algorithm (CGA) with a fitness computation method based on developmental theory is presented in [59]. CGA adopts a coevolutionary framework to regulate the level of difficulty of a process in a self-acting scheme. The difficulty is changed from easy to hard according to a population of tests. CGA is based on competitive coevolution between two species. The first is a species of possible solutions, whose behavior is very similar to that of a standard MOEA, that is to say, it uses selection, crossover and mutation operators

inside the population. The second species is composed of target individuals. Such individuals are target objective vectors (TOVs), which, as the name indicates, contain targets for each of the objectives being optimized. TOVs population is used to represent the level of difficulty that the trial population must confront. TOVs are evolved through the use of a genetic algorithm to change from easy to difficult based on the level of performance of the trial solutions. For this sake, at the beginning of the algorithm, the user has to provide a range of expectations conformed by an easy and difficult TOV, the latter to be used as the stopping condition of the approach. When populations of TOVs are created, they are initialized with random values that are inside the predefined range of difficulties. CGA makes use of the easy TOVs at the early stages of the algorithm to let the coevolutionary framework to act along the search process. If TOVs have a high difficulty at the beginning of the search process, no competition would be activated and the coevolutionary process of competition would not take place. TOVs are target objectives which act like thresholds, such that all thresholds must be reached or exceeded in order for the TOV to be *solved* and, therefore, gain fitness.

The procedure to compute fitness takes place in the following way. The TOV which is solved by just one solution in the current generation is considered to be the most difficult one, and, therefore, it is assigned with the best fitness score. Unsolvable TOVs and very easy ones are assigned the lowest fitness values. Fitness of an individual TOV is computed as follows. The number of trial solutions that solve a TOV are counted, and then this value is normalized adopting a tractability constraint. This constraint gives a target vector the *worst* score (1.0) when no trial individuals can solve it, so that the TOV population creates difficult but solvable problems to the trial population. On the other hand, trial individuals are rewarded when their solutions solve difficult TOVs. Their fitness is computed as follows. Individual *j* tries to solve each TOV in the other species and a score for the result is computed. This result is normalized making use of the predefined range of difficulty, in such a way that 0.0 is the best score and 1.0 is the worst. The effect of this score is to reward trial solutions that solve the hardest TOVs. When several trial solutions are able to solve the same TOV, their fitness contribution to the score is distributed among such solutions.

CGA was compared against seven MOEAs and one random search algorithm when solving the ZDT test problems [40]. In the experiments, the species population of target individuals is manipulated like the single population of a standard genetic algorithm and no explicit niching operators were used to enhance diversity. Also, a constrained mutation technique was used to ensure TOVs remained valid after mutation. One-point crossover was adopted, by choosing cut-points between individual objective values. Results showed that CGA is a very competitive approach: it always outperformed random search and had a good performance in terms of quality of the solutions against MOEAs such as the SPEA [65] and the NSGA [66]. However, this approach seems to generate solutions with a poor spread along the Pareto front. This can be observed in the results reported by its authors in which some regions of the corresponding Pareto fronts are not properly covered.

TABLE II
SUMMARY OF COMPETITIVE CMOEAS

Reference	Algorithm	Framework	Subpopulations	Representation	Spatial Structure
[51]	PPBBO	Predator-prey	Two	Real	None
[52]	MPP	Predator-prey	Two	Real	2D lattice
[63]	C-RMOEA/D	Competitive fitness	Two	Real	None
[57]	SPEA2-CE	Competitive fitness	One	Real	None
[59]	CGA	Target solutions	Two	Real	None
[67]	NNCA	Target solutions	Two	Real	None

2) Nondominated Neighbor Coevolutionary Algorithm:
A nondominated neighbor CA (NNCA) which adopts a coevolutionary framework for multiobjective optimization was proposed in [67]. NNCA works with two populations of nondominated values, one of elite solutions having the best crowding distance values, and another one that contains the remaining nondominated solutions, where elite solutions are meant to guide the search. The elite population is aimed to be in the least crowded regions of the fitness landscape, so that they are given a higher chance to select more individuals to form part of their population. With this, NNCA tries to guide the search to the most promising regions of the search space. To avoid premature convergence, NNCA uses a size boundary for the elite population. This mechanism removes individuals from the elite population when the limit size is reached. NNCA proposes a coevolutionary mechanism based on the way teams behave in human society. Elite individuals cooperate among them using symmetrical operators to breed, leading the other individuals to more promising areas by means of dissymmetrical operators to generate new solutions. This way, NNCA provides an improvement in terms of both convergence and spread of solutions.

NNCA also adopts a nondominated neighbor-based selection technique and a size guarantee mechanism, which work as follows. Elite individuals are given more opportunities to choose new individuals for their species so that those promising regions to which they belong are better explored. This scheme also improves the distribution of solutions, since elite individuals are those with the best distribution. To control the size of the elite population, when its actual size is smaller than the limit, dominated individuals or nonelite individuals are selected to form part of the elite species, so that the size of the elite population remains to the limit at every stage of the algorithm. Also, different crossover operators are simultaneously applied to produce new solutions, with the objective of improving diversity. Since there is no way of knowing ahead of time the ideal operator for each MOP, this method is meant to deal with the different difficulties presented in MOPs. NNCA was compared against seven state-of-the-art MOEAs when solving twenty MOPs (see [67] for further details about the test problems adopted). NNCA was found to be very competitive in all cases.

Nevertheless, further comparisons with other CMOEAs are required in order to obtain more insights about the advantages and possible limitations of this approach with respect to other coevolutionary schemes.

Table II summarizes the competitive CMOEAs we have just described.

VI. COMPETITIVE-COOPERATIVE CMOEAS

There are studies that indicate that a balance of cooperation and competition is needed to avoid stagnation and poor stable states in MOEAs [68]. The competitive-cooperative coevolution framework is meant to allow the decomposition process of the MOP to be self-adaptable, instead of being a static process just performed at the early stages of the algorithm. This new coevolutionary model is designed to attain good solutions and promote diversity along with an enhanced effective and efficient performance. In the same way as cooperative coevolutionary approaches do, members from different species work together to solve a problem. Each of the species is evolved in an independent way, with no restrictions of the encoding of the solutions neither for the election of the MOEA used as optimizer. Cooperative species are also involved in a competition for representing more parts of the problem. This competitive methodology unchains potential arms races among the species, who look toward improving their contribution to the overall performance of the whole ecosystem. This competition also allows the discovery of interdependencies among the components, since it provides an environment in which interdependent components end up within the same groups. Ideal problem decompositions emerge by means of the coevolutionary pressure instead of being a parameter predefined by the user. The competitive-cooperation coevolutionary scheme provides a mechanism to use diversity preservation techniques of both approaches: competitive and CC.

In the cooperative framework, the evolution of different species generates higher diversity among the different species. However, this feature does not extend to the internal behavior of each species. Also, the competition created by the species' necessity of gaining more resources, guides the way diversity behaves. Additionally, the competitive process improves diversity, since the optimization of each problem component is not restricted to only one species [69]. The species' competition provides an extra round of optimization for each part of the problem, which enhances the extent of the search with low computational requirements. The approaches described next are representative of this way of incorporating coevolutionary concepts within multiobjective optimization.

A. Coevolutionary Multiobjective Evolutionary Algorithm

A coevolutionary approach which adopts the competitive-cooperative framework for multiobjective evolutionary

optimization (CO-MOEA) is presented in [70]. CO-MOEA focuses the search efforts on the most promising regions that arise along the evolutionary process. This approach adopts a mechanism that divides the decision variables of the problem according to the importance that each variable has over the solution to the MOP. For this sake, the algorithm uses an analysis of the nondominated solutions obtained so far, with the goal of identifying which regions of the search space are the most promising. The overall algorithm is divided into four stages, that is to say that the total number of allowable cycles of the algorithm is divided by four. The algorithm works as follows.

1) At the beginning, CO-MOEA makes an exploration of decision variables space adopting the ranking technique presented in [71] and the adaptive grid proposed in [72]. Thereafter, the algorithm performs an analysis of the current nondominated solutions stored in the adaptive grid to know the level of importance of each of the decision variables of the MOP. Such analysis looks at the values of the decision variables of the current set of best solutions, in order to discover if certain decision variables values are distributed along the limits of the problem or if they are concentrated in a more narrow region. When the values of certain decision variables are distributed all along the allowable range, CO-MOEA maintains the whole interval for each of those decision variables. On the other hand, if the values of the decision variables used a more reduced part of the interval, the algorithm looks for the portions of the search space that can be removed. In other words, the idea is to determine the importance that each part of the search space has over the solution of the problem and the possible divisions (and the number of them) that can be performed to use in a more clever way the available resources. Once the important regions for the search have been determined, they are assigned to different species.

2) At the second stage, the algorithm uses the species created in the first stage to search in all regions of the decision variables space The search process of all species is performed in an independent manner, and the nondominated solutions found so far by each species' population are placed in the adaptive grid in order to *cooperate* and *compete* to form a global set of nondominated solutions that represents the approximation of the Pareto front. Thereafter, species which do not generate nondominated solutions to the global set of solutions are erased and their resources (i.e., individuals) are distributed along the remaining species, providing more resources to those species which contribute the most to the overall approximation of the Pareto front and less to those species that have a lower contribution. In other words, at each cycle, individuals are assigned or removed from species according to their performance, in such a way that each population size is proportional to its performance. Therefore, species *compete* against each other to obtain more individuals for their population.

3) At the third stage, the algorithm checks the performance of each species to decide which are meant to

survive. Thereafter, the process from the second stage is applied over the current populations (i.e., the intervals of the decision variables are divided again and new species are created with the purpose of achieving a better exploration of the most promising regions of the search space). At this stage, a minimum population size for each species is defined and is inflicted for all species at the beginning of this stage.

4) Finally, during the fourth stage, the previously described procedure for the third stage is performed to obtain a more refined search.

CO-MOEA was benchmarked using three test problems taken from the specialized literature (see [70] for further details about the test problems) and was compared with respect to three other MOEAs: 1) the microGA for multiobjective optimization [73]; 2) PAES [74]; and 3) NSGA-II [64]. The authors used a fixed number of generations for each problem and the final results of each algorithm were assessed using four performance indicators: 1) the two SC [40]; 2) SP [75]; 3) GD [76]; and 4) error ratio [77]. This comparative study showed that CO-MOEA had a competitive performance with respect to the other MOEAs with respect to which it was compared.

The main drawback of CO-MOEA is that the number of populations could increase so much that the algorithm would become quite inefficient. Also, it is necessary to decrease the high selection pressure introduced by its elitist scheme, since this may cause premature convergence when a false attractor exists in the population.

B. Competitive and Cooperative Coevolutionary Multiobjective Particle Swarm Optimization Algorithm

A competitive and cooperative coevolutionary approach for multiobjective particle swarm optimization (CCPSO) is presented in [78]. CCPSO is capable of dealing with complex MOPs by adopting a competitive-cooperative framework. This approach proposes the use of a credit assignment method for the competitive-cooperative process; such method is applied both at the species and at the individual level. For the cooperative behavior, CCPSO evaluates the different objective functions of the MOP by forming solutions created with the collaboration of all the species. That is, to say, to assemble a valid vector of decision variables to the problem, representatives from every species are taken to form a possible solution to the MOP. Thereafter, a Pareto ranking technique is applied to perform the fitness assignment to each individual. On the other hand, for the competitive interaction, fitness is computed according to how better is the performance of each species in comparison to the other competing species when forming solutions to the problem. For example, species i selects n competing species to represent a subpart of the problem. If the subpart is being represented by a species which brings a good performance, this species is given a high probability of representing the same part of the problem at later stages of the search. Meanwhile, those species which show a poor performance are penalized with a lower probability of representing that part of the problem in future cycles. The overall procedure of the algorithm is presented next.

CCPSO decomposes the problem in terms of the decision variables by creating subcomponents of the vector of decision variables. Each component is then assigned to a species' population of particles [known as *swarms* in particle swarm optimization (PSO)]. In other words, n species of swarms are created and initialized having the jth decision variable assigned to the jth species. So, to perform the evaluation of a particle, it has to collaborate with a representative from each of the other species in order to assemble a complete solution to the MOP. These representatives from each species are the particles which are considered as the best in their swarm. The external archive of nondominated solutions is then updated after each particle's evaluation and the new best individual is chosen after the evaluation of the whole species. The best particle in each species is chosen by means of a partial order method, where Pareto ranking is considered as the first criterion and a crowding method is adopted as a secondary method for breaking ties. When comparing the particles, the one with the lowest rank is chosen and in case of a tie (i.e., if they are nondominated with respect to each other), the particle having the least crowded value is selected. This approach is meant to increase the distribution of the solutions as well as to improve the convergence of the algorithm.

Each species has a probability of representing a specific decision variable, such that just two species can compete to represent any decision variable at a time. This probability is initialized with a value $1/L$, where L is the number of decision variables of the MOP being solved, this with the aim of allowing a fair change to each species to represent any of the decision variables. Thereafter, the probability of representing a variable depends on a competition where roulette wheel selection takes place to select individuals from different species to compete. So, the assignment occurs as follows. In the very first cycle, the jth variable is assigned to the jth species. Then, a species is selected to compete using roulette wheel selection, which is performed using the current selection probabilities of representing the decision variable in turn. In order to compete, one representative from each competing species is combined with the representatives of the other different species in order to assemble a solution. Once the solutions are formed, the species providing the best solution is the one which represents the decision variable in the following cycle of the process. The probability of representing the decision variable is increased as the species is more and more specialized in the treatment of such decision variable. Conversely, those species having a poor management of a decision variable have a lower chance to represent it. It is important to note that, as the process goes on, it is possible that one species represents several decision variables at the same time. The best individual in each species is known as the *global-best* in PSO. As in the multiobjective context there is no a single solution being the best but a set of them, the global-best is a particle from the set of nondominated solutions who wins a binary tournament, where a crowding measure is again the second criterion to break ties. With this, CCPSO ensures spread, diversity and convergence of the solutions. CCPSO adopts a limit size for the external archive of nondominated solutions. For this sake, a niche count method is applied when the limit size is reached, so that only nondominated solutions residing in the least crowded regions become members of the archive.

CCPSO was benchmarked against several MOEAs representative of the state-of-the-art, adopting several well-established test problems and performance indicators (see [78] for further details). Results showed that CCPSO had a faster convergence than ordinary MOEAs. Nevertheless, CCPSO showed a poor performance when dealing with multifrontal MOPs. The use of the competitive-cooperative coevolutionary framework allows CCPSO to bring the fast convergence speed of PSO along near-optimal and diverse Pareto fronts in all the adopted MOPs. A sensitivity analysis to analyze the effects of different inertia weight values was performed, showing that CCPSO achieves a better convergence with a smaller inertia weight and that the use of an adaptive inertia scheme as well as higher inertia weights are able to generate better distributed Pareto fronts.

The main drawback of CCPSO is that its high speed of convergence results in a poor performance (e.g., premature convergence) for multifrontal MOPs such as ZDT4.

VII. Pathologies of the Coevolutionary Framework

Since the evaluation of individuals in CMOEAs is based on coevolving individuals, coevolution settings may cause an inaccurate evaluation, leading to problems such as *intransivity*, *disengagement*, and *red queen effect* among others [79]–[81]. Next, we briefly describe these problems.

A. Intransitivity

One problem feature that has received particular interest in the past is that of intransitivity [82]. As described by de Jong [82], a relation R is transitive if $aRb \land bRc$ implies aRc; if this cannot be guaranteed, the relation is then intransitive. The existence of such intransitive relations in a coevolutionary system can lead to *cycling*, i.e., the recurrence of previously visited states of the species. Intransitivity has been viewed as an inherent feature of coevolution that can make CMOEAs unreliable. Indeed, the resulting problem of cycling has been thought of as an obstacle that could prevent coevolution from becoming a reliable problem solving technique. In fact, some researchers believe that intransitivity is a problem that, similarly to the case of the presence of local minima in gradient-descent methods, cannot be completely eliminated [83].

B. Disengagement

Coevolutionary disengagement takes place when one population outperforms another to the extent that individuals from the same species become indistinguishable from one another (in terms of fitness) [84]. When this occurs, coevolved subpopulations become disassociated and selection acts with no specific direction, causing the coevolutionary process to drift and, in many cases, without any possibility of generating acceptable results.

C. Red Queen Effect

CAs can suffer from the so-called *red queen effect*. This happens when, through their interaction, species alter each other's fitness landscapes [85]. Such effect significantly affects the performance of the coevolutionary process, creating fitness ambiguities that cause improvements in the performance of individuals to be considered as undesired changes and viceversa.

VIII. FUTURE RESEARCH PATHS

In this section, we will provide some insights regarding topics that we believe are worth investigating within this area.

A. Coevolution Based on Indicators

Most MOEAs in current use have selection mechanisms based on Pareto optimality. Nevertheless, it is well-known that Pareto-based MOEAs cannot properly deal with MOPs having four or more objective functions (the so-called *many-objective optimization problems* [86]). Alternative selection methods have been proposed to alleviate this problem. Among such proposals, the use of selection mechanisms or density estimators based on performance indicators has been a popular choice [87]. When adopting indicator-based selection, the idea is to identify the solutions that contribute the most to the improvement of an indicator measure (e.g., the hypervolume). This sort of selection mechanism could also be used within either competitive or cooperative CMOEAs. For example, at the time of selecting individuals from other species to perform evaluation of an individual, the usual approach is a mechanism based on Pareto optimality (as evidenced by [19]), but the use of a performance indicator is also possible. CMOEAs based on a performance indicator could probably be more suitable for large-scale many-objective optimization which, by the way, is a research area that has remained practically unexplored until now. It is worth indicating that the use of indicator-based selection has mainly focused on maintaining diversity rather than on selecting solutions, but their interaction with coevolution has not been studied until now. To the best of the authors' knowledge, the only work in this direction has been recently presented in [18], but there is still a wide range of possibilities left to explore.

B. Coevolutionary Memetic MOEAs

With the aim of making MOEAs more efficient for real-world applications, local search techniques have been applied to create more effective and efficient MOEAs. Several local search techniques have been proposed along with domain specific methods [5], [88]. To the best of the authors' knowledge, nobody has explored so far the idea of combining coevolutionary frameworks with MOEAs incorporating local search techniques (this would be a coevolutionary memetic MOEA). Balancing the global search performed by the CMOEA with the local search mechanism would certainly pose several challenging issues.

C. Parallelizing CMOEAs

We believe that another promising research area is the parallelization of coevolutionary MOEAs (e.g., using GPUs [89]), since this would allow them to deal with computationally expensive applications (e.g., in aeronautical engineering [90]). This could also extend the use of CMOEAs to genetic programming, in which, to the best of the authors' knowledge, they have not been used yet.

D. Coevolution of Preferences

We believe that the concept of coevolving a family of decision-maker preferences together with a population of candidate solutions still has many research paths left to explore, as evidenced by Wang *et al.* [43]. Preference-based approaches are suitable for the generation of tradeoff surfaces in objective subspaces of interest to the decision maker. Therefore, coevolving a family of preferences simultaneously with the population of candidate solutions is a promising concept for solving many-objective problems.

IX. CONCLUSION

This paper has provided a review of the state-of-the-art regarding adaptations of CAs to handle MOPs, including a general introduction to this sort of algorithmic framework. In our review, the main coevolutionary paradigms in current use in combination with MOEAs have been analyzed and briefly discussed. Additionally, the approaches reviewed in this paper have been placed in a taxonomy proposed by us to classify the different Coevolutionary MOEAs currently available in the specialized literature.

In the final part of this paper, we have described some research paths that have not been explored so far and that we believe that could be worth exploring in the next few years.

REFERENCES

[1] J. Paredis, "Coevolutionary computation," *Artif. Life*, vol. 2, no. 4, pp. 355–375, Aug. 1995.

[2] 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, Mar. 2000.

[3] C. D. Rosin and R. K. Belew, "New methods for competitive coevolution," *Evol. Comput.*, vol. 5, no. 1, pp. 1–29, Mar. 1997.

[4] S. G. Ficici, "Multiobjective optimization and coevolution," in *Multiobjective Problem Solving From Nature* (Natural Computing Series). Heidelberg, Germany: Springer, 2008, pp. 31–52.

[5] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York, NY, USA: Kluwer Acad., May 2002.

[6] L. M. Antonio and C. A. Coello Coello, "A non-cooperative game for faster convergence in cooperative coevolution for multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Sendai, Japan, May 2015, pp. 109–116.

[7] P. R. Ehrlich and P. H. Raven, "Butterflies and plants: A study in coevolution," *Evolution*, vol. 18, no. 4, pp. 586–608, 1964.

[8] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in *Proc. Int. Conf. Evol. Comput. 3rd Conf. Parallel Problem Solving Nat. Parallel Problem Solving Nat. (PPSN III)*, Jerusalem, Israel, 1994, pp. 249–257.

[9] R. R. N. Mohammad and Z. Kobti, "A new strategy to detect variable interactions in large scale global optimization," in *Proc. IEEE Symp. Swarm Intell. (SIS)*, Orlando, FL, USA, Dec. 2014, pp. 1–8.

[10] Z. Yang, K. Tang, and X. Yao, "Differential evolution for high-dimensional function optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, vol. 1. Singapore, Sep. 2007, pp. 3523–3530.

[11] M. N. Omidvar, X. Li, Z. Yang, and X. Yao, "Cooperative co-evolution for large scale optimization through more frequent random grouping," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, vol. 1. Barcelona, Spain, Sep. 2010, pp. 1–8.

[12] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Inf. Sci.*, vol. 178, no. 15, pp. 2985–2999, Aug. 2008.

[13] A. Zamuda, J. Brest, B. Boskovic, and V. Zumer, "Large scale global optimization using differential evolution with self-adaptation and cooperative co-evolution," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, 2008, pp. 3718–3725.

[14] Z. Yang, K. Tang, and X. Yao, "Multilevel cooperative coevolution for large scale optimization," in *Proc. IEEE Congr. Evol. Comput. World Congr. Comput. Intell. (CES)*, Hong Kong, 2008, pp. 1663–1670.

[15] H. J. C. Barbosa and A. M. S. Barreto, "An interactive genetic algorithm with co-evolution of weights for multiobjective problems," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, San Francisco, CA, USA, Jul. 2001, pp. 7–11.

[16] N. Keerativuttiumrong, N. Chaiyaratana, and V. Varavithya, "Multi-objective co-operative co-evolutionary genetic algorithm," in *Parallel Problem Solving From Nature—PPSN VII* (LNCS 2439), J. J. M. Guervós, P. Adamidis, H.-G. Beyer, J.-L. Fernández-Villacañas, and H.-P. Schwefel, Eds. Heidelberg, Germany: Springer-Verlag, Sep. 2002, pp. 288–297.

[17] L. M. Antonio and C. A. Coello Coello, "Use of cooperative coevolution for solving large scale multiobjective optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Cancún, México, Jun. 2013, pp. 2758–2765.

[18] L. M. Antonio and C. A. Coello Coello, "Indicator-based cooperative coevolution for multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, Jul. 2016, pp. 991–998.

[19] A. W. Iorio and X. Li, "A cooperative coevolutionary multiobjective algorithm using non-dominated sorting," in *Proc. Genet. Evol. Comput. Conf. Part I (GECCO)*, vol. 3102. Seattle, WA, USA, Jun. 2004, pp. 537–548.

[20] K. Maneeratana, K. Boonlong, and N. Chaiyaratana, "Multi-objective optimisation by co-operative co-evolution," in *Parallel Problem Solving From Nature—PPSN VIII*. (LNCS 3242). Heidelberg, Germany: Springer-Verlag, Sep. 2004, pp. 772–781.

[21] T. G. Tan, H. K. Lau, and J. Teo, "Cooperative coevolution for Pareto multiobjective optimization: An empirical study using SPEA2," in *Proc. IEEE Region 10 Conf. TENCON*, Taipei, Taiwan, Oct./Nov. 2007, pp. 1–4.

[22] Y.-J. Zheng and S.-Y. Chen, "Cooperative particle swarm optimization for multiobjective transportation planning," *Appl. Intell.*, vol. 39, no. 1, pp. 202–216, 2013, doi: 10.1007/s10489-012-0405-5.

[23] B. Dorronsoro, G. Danoy, P. Bouvry, and A. J. Nebro, "Multi-objective cooperative coevolutionary evolutionary algorithms for continuous and combinatorial optimization," in *Intelligent Decision Systems in Large-Scale Distributed Environments* (Studies in Computational Intelligence), vol. 362, P. Bouvry, H. González-Vélez, and J. Kołodziej, Eds. Heidelberg, Germany: Springer, 2011, pp. 49–74.

[24] T. Tan and J. Teo, "Improving the performance of multiobjective evolutionary optimization algorithms using coevolutionary learning," in *Nature-Inspired Algorithms for Optimisation* (Studies in Computational Intelligence), vol. 193, R. Chiong, Ed. Heidelberg, Germany: Springer, 2009, pp. 457–487.

[25] T. G. Tan, H. K. Lau, and J. Teo, "Cooperative versus competitive coevolution for Pareto multiobjective optimization," in *Bio-Inspired Computational Intelligence and Applications* (LNCS 4688), K. Li, M. Fei, G. Irwin, and S. Ma, Eds. Heidelberg, Germany: Springer, 2007, pp. 63–72.

[26] K. C. Tan, Y. J. Yang, and C. K. Goh, "A distributed cooperative coevolutionary algorithm for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 527–549, Oct. 2006.

[27] T. G. Tan, J. Teo, and H. K. Lau, "Performance scalability of a cooperative coevolution multiobjective evolutionary algorithm," in *Proc. Int. Conf. Comput. Intell. Security*, Harbin, China, Dec. 2007, pp. 119–123.

[28] K. C. Tan, Y. H. Chew, T. H. Lee, and Y. J. Yang, "A cooperative coevolutionary algorithm for multiobjective optimization," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, vol. 1. Washington, DC, USA, Oct. 2003, pp. 390–395.

[29] W. Zhao, S. Alam, and H. A. Abbass, "MOCCA-II: A multi-objective co-operative co-evolutionary algorithm," *Appl. Soft Comput.*, vol. 23, pp. 407–416, Oct. 2014.

[30] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, A. Abraham, L. Jain, and R. Goldberg, Eds. London, U.K.: Springer, 2005, pp. 105–145.

[31] M. Gong, H. Li, E. Luo, J. Liu, and J. Liu, "A multiobjective cooperative coevolutionary algorithm for hyperspectral sparse unmixing," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 234–248, Apr. 2017.

[32] H. Wang, L. Jiao, and X. Yao, "Two_arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 524–541, Aug. 2015.

[33] M. Li, S. Yang, and X. Liu, "Pareto or non-Pareto: Bi-criterion evolution in multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 645–665, Oct. 2016.

[34] J. Wang, W. Zhang, and J. Zhang, "Cooperative differential evolution with multiple populations for multiobjective optimization," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2848–2861, Dec. 2016.

[35] C. A. Coello Coello and N. C. Cortés, "Solving multiobjective optimization problems using an artificial immune system," *Genet. Program. Evol. Mach.*, vol. 6, no. 2, pp. 163–190, Jun. 2005.

[36] Q. Zhang *et al.*, "Multiobjective optimization test instances for the CEC 2009 special session and competition," School Comput. Sci. Electron. Eng., Univ. Essex, Colchester, U.K., and Nanyang Technol. Univ., Singapore, Tech. Rep. CES-487, pp. 1–30, 2008.

[37] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

[38] J. Nash, "Non-cooperative games," *Ann. Math.*, vol. 54, no. 2, pp. 286–295, 1951.

[39] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Proc. Evol. Methods Design Optim. Control Appl. Ind. Problems (EUROGEN)*, Athens, Greece, 2002, pp. 95–100.

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

[41] K.-B. Sim, D.-W. Lee, and J.-Y. Kim, "Game theory based coevolutionary algorithm: A new computational coevolutionary approach," *Int. J. Control Autom. Syst.*, vol. 2, no. 4, pp. 463–474, 2004.

[42] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired co-evolutionary algorithms using weight vectors," *Eur. J. Oper. Res.*, vol. 243, no. 2, pp. 423–441, 2015.

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

[44] W. D. Hillis, "Co-evolving parasites improve simulated evolution as an optimization procedure," *Phys. D Nonlin. Phenomena*, vol. 42, nos. 1–3, pp. 228–234, 1990.

[45] M. Laumanns, G. Rudolph, and H.-P. Schwefel, "A spatial predator-prey approach to multi-objective optimization: A preliminary study," in *Parallel Problem Solving From Nature—PPSN V*, A. E. Eiben, M. Schoenauer, and H.-P. Schwefel, Eds. Heidelberg, Germany: Springer-Verlag, 1998, pp. 241–249.

[46] X. Li, "A real-coded predator-prey genetic algorithm for multiobjective optimization," in *Proc. 2nd Int. Conf. Evol. Multi Criterion Optim. (EMO)*, vol. 2632. Faro, Portugal, Apr. 2003, pp. 207–221.

[47] R. Drężewski and L. Siwik, "Co-evolutionary multi-agent system with predator-prey mechanism for multi-objective optimization," in *Proc. 8th Int. Conf. Adapt. Nat. Comput. Algorithms Part I (ICANNGA)*, vol. 4431. Warsaw, Poland, Apr. 2007, pp. 67–76.

[48] S. Chowdhury, R. Moral, and G. Dulikravich, "Predator-prey evolutionary multi-objective optimization algorithm: Performance and improvements," in *Proc. 7th Int. Conf. Eng. Design Optim. (ASMO U.K./ISSMO)*, Bath, U.K., Jul. 2008, pp. 1–10.

[49] K. Deb *et al.*, "Investigating predator-prey algorithms for multi-objective optimization," Kanpur Genet. Algorithms Lab., Kanpur India, KanGAL Tech. Rep. 2005010, 2005.

[50] K. Schmitt, J. Mehnen, and T. Michelitsch, "Using predators and preys in evolution strategies," in *Proc. 7th Annu. Conf. Genet. Evol. Comput.*, Washington, DC, USA, 2005, pp. 827–828.

[51] M. de A. Costa e Silva, L. dos S. Coelho, and L. Lebensztajn, "Multiobjective biogeography-based optimization based on predator-prey approach," *IEEE Trans. Magn.*, vol. 48, no. 2, pp. 951–954, Feb. 2012.

[52] S. Chowdhury, G. S. Dulikravich, and R. J. Moral, "Modified predator-prey algorithm for constrained and unconstrained multi-objective optimisation," *Int. J. Math. Model. Numer. Optim.*, vol. 1, nos. 1–2, pp. 1–38, 2009.

[53] C. Grimme, J. Lepping, and A. Papaspyrou, "The parallel predator-prey model: A step towards practical application," in *Parallel Problem Solving From Nature—PPSN X* (LNCS 5199), G. Rudolph, T. Jansen, S. Lucas, C. Poloni, and N. Beume, Eds. Heidelberg, Germany: Springer, 2008, pp. 681–690.

[54] C. Grimme and K. Schmitt, "Inside a predator-prey model for multi-objective optimization: A second study," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, vol. 1. Seattle, WA, USA, Jul. 2006, pp. 707–714.

[55] M. P. Kleeman and G. B. Lamont, "Coevolutionary multi-objective EAs: The next frontier?" in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, Jul. 2006, pp. 6190–6199.

[56] T. Tan and J. Teo, "Evolving opposition-based Pareto solutions: Multiobjective optimization using competitive coevolution," in *Oppositional Concepts in Computational Intelligence* (Studies in Computational Intelligence), vol. 155, H. Tizhoosh and M. Ventresca, Eds. Heidelberg, Germany: Springer, 2008, pp. 161–206.

[57] T. G. Tan, J. Teo, and H. K. Lau, "Competitive coevolution with k-random opponents for Pareto multiobjective optimization," in *Proc. 3rd Int. Conf. Nat. Comput. (ICNC)*, vol. 4. Haikou, China, Aug. 2007, pp. 63–67.

[58] I. C. Parmee and A. H. Watson, "Preliminary airframe design using co-evolutionary multiobjective genetic algorithms," in *Proc. Genet. Evol. Comput. Conf.*, vol. 2. Orlando, FL, USA, Jul. 1999, pp. 1657–1665.

[59] J. D. Lohn, W. F. Kraus, and G. L. Haith, "Comparing a coevolutionary genetic algorithm for multiobjective optimization," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 2. Honolulu, HI, USA, 2002, pp. 1157–1162.

[60] J.-G. Liu, "Competitive coevolutionary genetic algorithms for multiobjective optimization problems," in *Proc. Int. Conf. Artif. Intell. Comput. Intell. (AICI)*, vol. 3. Shanghai, China, Nov. 2009, pp. 594–597.

[61] J. Liu and W. Wu, "Multiobjective optimization with competitive coevolutionary genetic algorithms," in *Proc. 29th Chin. Control Conf. (CCC)*, Beijing, China, Jul. 2010, pp. 5264–5267.

[62] J. Lee and D. Kim, "A method of genetic algorithm based multiobjective optimization via cooperative coevolution," *J. Mech. Sci. Technol.*, vol. 20, no. 12, pp. 2115–2123, 2006, doi: 10.1007/BF02916328.

[63] I. R. Meneghini, F. G. Guimarães, and A. Gaspar-Cunha, "Competitive coevolutionary algorithm for robust multi-objective optimization: The worst case minimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, Jul. 2016, pp. 586–593.

[64] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," Indian Inst. Technol., Kanpur, India, KanGAL Tech. Rep. 200001, 2000.

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

[66] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1994.

[67] C. Mu, L. Jiao, Y. Liu, and Y. Li, "Multiobjective nondominated neighbor coevolutionary algorithm with elite population," *Soft Comput.*, vol. 19, no. 5, pp. 1329–1349, 2015, doi: 10.1007/s00500-014-1346-1.

[68] S. G. Ficici and J. B. Pollack, "Challenges in coevolutionary learning: Arms-race dynamics, openendedness, and mediocre stable states," in *Proc. 6th Int. Conf. Artif. Life (ALIFE)*, 1998, pp. 238–247.

[69] C. A. Coello Coello and M. R. Sierra, "A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm," in *Proc. 3rd Mexican Int. Conf. Artif. Intell. (MICAI)*, vol. 2972. Mexico City, Mexico, Apr. 2004, pp. 688–697.

[70] C. A. Coello Coello and M. R. Sierra, "A coevolutionary multi-objective evolutionary algorithm," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 1. Canberra, ACT, Australia, Dec. 2003, pp. 482–489.

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

[72] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, 2000.

[73] C. A. Coello Coello and G. T. Pulido, "A micro-genetic algorithm for multiobjective optimization," in *Proc. 1st Int. Conf. Evol. Multi Criterion Optim.*, Zürich, Switzerland, 2001, pp. 126–140.

[74] J. D. Knowles and D. W. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation," in *Proc. Congr. Evol. Comput.*, Washington, DC, USA, Jul. 1999, pp. 98–105.

[75] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," M.S. thesis, Dept. Aeronautics Astronautics, Massachusetts Inst. Technol., Cambridge, MA, USA, May 1995.

[76] D. A. V. Veldhuizen and G. B. Lamont, "On measuring multiobjective evolutionary algorithm performance," in *Proc. Congr. Evol. Comput.*, vol. 1. Piscataway, NJ, USA, Jul. 2000, pp. 204–211.

[77] G. B. Lamont and D. A. V. Veldhuizen, "Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations," Ph.D. dissertation, Dept. Elect. Comput. Eng., Graduate School Eng., Air Force Inst. Technol., Wright-Patterson AFB, OH, USA, May 1999.

[78] C. K. Goh, K. C. Tan, D. S. Liu, and S. C. Chiam, "A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design," *Eur. J. Oper. Res.*, vol. 202, no. 1, pp. 42–54, 2010.

[79] R. A. Watson and J. B. Pollack, "Coevolutionary dynamics in a minimal substrate," in *Proc. 3rd Annu. Conf. Genet. Evol. Comput. (GECCO)*, San Francisco, CA, USA, 2001, pp. 702–709.

[80] S. Luke and R. P. Wiegand, "When coevolutionary algorithms exhibit evolutionary dynamics," in *Proc. Genet. Evol. Comput. Conf. Workshop Program*, 2002, pp. 236–241.

[81] L. Panait, R. P. Wiegand, and S. Luke, "A sensitivity analysis of a cooperative coevolutionary algorithm biased for optimization," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, vol. 3102. Seattle, WA, USA, Jun. 2004, pp. 573–584.

[82] E. D. de Jong, "Intransitivity in coevolution," in *Parallel Problem Solving from Nature—PPSN VIII* (LNCS 3242). Heidelberg, Germany: Springer-Verlag, Sep. 2004, pp. 843–851.

[83] S. Nolfi and D. Floreano, "Coevolving predator and prey robots: Do 'arms races' arise in artificial evolution?" *Artif. Life*, vol. 4, no. 4, pp. 311–335, Oct. 1998.

[84] J. Cartlidge and S. Bullock, "Combating coevolutionary disengagement by reducing parasite virulence," *Evol. Comput.*, vol. 12, no. 2, pp. 193–222, Jun. 2004, doi: 10.1162/106365604773955148.

[85] D. Cliff and G. F. Miller, "Tracking the red queen: Measurements of adaptive progress in co-evolutionary simulations," in *Advances in Artificial Life—ACAL 1995* (Lecture Notes in Artificial Intelligence), vol. 929. Heidelberg, Germany: Springer, 1995, pp. 200–218.

[86] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE Congr. Evol. Comput. World Congr. Comput. Intell. (CEC)*, Hong Kong, Jun. 2008, pp. 2419–2426.

[87] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving From Nature—PPSN VIII* (LNCS 3242), X. Yao *et al.*, Eds. Birmingham, U.K.: Springer-Verlag, Sep. 2004, pp. 832–842.

[88] J.-S. Pangc and D. Ralph, Eds., "Special issue on nonlinear programming variational inequalities, and stochastic programming," *Math. Program.*, vol. 117, no. 1, pp. 1–4, Mar. 2009.

[89] M. L. Wong and G. Cui, "Data mining using parallel multi-objective evolutionary algorithms on graphics processing units," in *Massively Parallel Evolutionary Computation on GPGPUs*, S. Tsutsui and P. Collet, Eds. Heidelberg, Germany: Springer, 2013, pp. 287–307.

[90] A. Arias-Montaño, C. A. Coello Coello, and E. Mezura-Montes, "Multiobjective evolutionary algorithms in aeronautical and aerospace engineering," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 662–694, Oct. 2012.

Luis Miguel Antonio received the bachelor's degree in computer systems from the Instituto Politécnico Nacional, Escuela Superior de Cómputo, Mexico City, México, in 2011, and the master's degree in computer science from the Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Mexico City, in 2013, where he is currently pursuing the Ph.D. degree.

His current research interests include evolutionary multiobjective optimization, large-scale multiobjective optimization, and machine learning.

Carlos A. Coello Coello (M'98–SM'04–F'11) received the Ph.D. degree in computer science from Tulane University, New Orleans, LA, USA, in 1996.

He is currently a Professor (with distinction, CINVESTAV-3F Researcher) with the Computer Science Department, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Mexico City, Mexico. He has authored and co-authored over 450 technical papers and book chapters. He has also co-authored the book entitled *Evolutionary Algorithms for Solving Multi-Objective Problems* (Second edition, Springer, 2007). His publications currently report over 35 900 citations in *Google Scholar* with an H-index of 76). His current research interests include evolutionary multiobjective optimization and constraint-handling techniques for evolutionary algorithms.

Dr. Coello Coello was a recipient of the 2007 National Research Award from the Mexican Academy of Sciences in the area of exact sciences and the Medal to the Scientific Merit 2009 granted by Mexico City's congress. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and serves in the editorial board of 12 other international journals. He is a member of ACM, Sigma Xi, and the Mexican Academy of Science.