# A Non-cooperative Game for Faster Convergence in Cooperative Coevolution for Multi-objective Optimization

Luis Miguel Antonio and Carlos A. Coello Coello
Computer Science Department
CINVESTAV-IPN (Evolutionary Computation Group)
Av. IPN No. 2508, Col. San Pedro Zacatenco, Mexico City .07300, Mexico.
lmiguel@computacion.cs.cinvestav.mx, ccoello@cs.cinvestav.mx

***Abstract*—Cooperative coevolution is an approach for evolving solutions from different populations which are evaluated based on how well they perform together. The advantage of cooperative coevolutionary algorithms is the decomposition of the problem which allows us to learn different parts of the problem instead of the whole problem at once. However, previous research within the field of global optimization has shown that cooperative coevolutionary algorithms are biased towards equilibrium states. Since studies concerning cooperative coevolutionary algorithms used for solving multi-objective optimization problems were initiated, no attention has been paid to this issue. In this paper, we show empirical evidence of the existence of these problems within the multi-objective optimization field and present a novel cooperative coevolution framework which, through the use of the concept of Nash equilibrium, alleviates some of those optimization-related pathologies present in cooperative coevolutionary algorithms. We compare our proposed algorithm with respect to two algorithms that make use of the cooperative coevolutionary model to multi-objective optimization, NSCCGA (that makes use of Potter's coevolutionary model) and GCEA (a game theory based coevolutionary algorithm). The computational effort required by each algorithm (measured in terms of the number of fitness function evaluations) is also analyzed. Our preliminary results indicate that the proposed framework clearly outperforms the results of the aforementioned algorithms when using the Deb-Thiele-Laumanns-Zitzler (DTLZ) and the Zitzler-Deb-Thiele (ZDT) test suites.**

## I. Introduction

Coevolutionary algorithms (CAs) are extensions of traditional evolutionary algorithms (EAs). The main difference between CAs and EAs is the adaptive nature of the fitness evaluation for the members of coevolutionary systems. Individuals are assigned with fitness values based on the interactions that they have with other individuals from other species. Coevolution then refers to a reciprocal evolutionary change between species that interact with each other. Such models have been historically categorized as competitive or cooperative. In this work, our interest is in the second category, the so-called cooperative coevolutionary algorithms (CCAs), which are able to exploit the compositional nature of problems. CCAs use individuals which belong to different species that represent a component of a larger problem. Each of these components receive a fitness value based on how well it performs in conjunction with individuals from other species. The first framework of cooperative coevolution (CC) utilized within evolutionary algorithms was originally introduced by Potter and De Jong [1], in their *Cooperative Coevolutionary Genetic Algorithm* (CCGA). They found that CCGA has a significant improvement in performance over the traditional GA when optimizing functions with 30 decision variables. This framework uses a divide-and-conquer approach to split the decision variables into subpopulations of smaller size, so that each of these subpopulations is optimized with a separate EA. The main idea is to decompose a problem into several subcomponents and evolve these subcomponents cooperatively for a predefined number of *cycles*.

Since the cooperative coevolutionary framework can be extended in a relatively easy way to multi-objective optimization, a number of approaches have been proposed which incorporate it to improve the performance of multi-objective EAs [2], [3]. However, CCAs have some issues among which premature convergence to equilibrium states stands out [4], [5]. This pathology emerges since in CCAs, each species population is searching in its subcomponent of the problem, and that projection is constantly changing. So, it is highly likely for CCAs to get misleading information that leads them to prefer individuals in one population which do well with most of the individuals from other populations, even if these combinations are not optimal. This work treats this phenomenon from the multi-objective cooperative coevolutionary point of view and proposes a novel cooperative coevolutionary framework that makes use of the concept of Nash equilibrium to alleviate the *relative overgeneralization* [4] present in coevolution.

The remainder of this paper is organized as follows. In Section II, we provide some basic concepts related to multi-objective optimization, which are included to make the paper self-contained. The previous related work is discussed in Section III. Section IV is devoted to present our proposed approach and the experiments carried out. Finally, our conclusions and some possible paths for future work are drawn in Section V.

## II. The Multi-objective Optimization Problem

The Multi-objective Optimization Problem (MOP) can be defined as the problem of finding a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. Hence, the term "optimize" means finding such a solution which would give the values of all the objective functions acceptable to the decision maker [6]. In the following definitions we are assuming, without loss of generality, the minimization of all

the objectives. Mathematically, we can write MOPs as:

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

subject to:

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

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

where $k$ is the number of objective functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $gi, hj : \mathbb{R}^n \to \mathbb{R}, i = 1, ..., m, j = 1, ..., p$ are the constraint functions of the problem and $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ the vector of decision variables. We thus wish to determine from the set $\Omega$ (where $\Omega$ is the feasible region) of all the vectors that satisfy (2) and (3) to the vector $\vec{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ of solutions that are *Pareto optimal*. To describe the concept of optimality that we will adopt, we need to introduce a few additional definitions.

**Pareto Optimality:** We say that a vector of decision variables $\vec{x}^* \in \Omega$ (where $\Omega$ is the feasible region) is **Pareto Optimal** with respect to $\Omega$ if and only if $\forall \vec{x} \in \Omega \wedge \forall i \in \{1, \ldots, k\}$:

$$f_i(\vec{x}) = f_i(\vec{x}^*) \vee \nexists i \in \{1, \ldots, k\} : f_i(\vec{x}) < f_i(\vec{x}^*) \quad (4)$$

**Pareto Dominance:** A vector $\vec{u} = [u_1, \ldots, u_k]^T$ is said to **dominate** another vector $\vec{v} = [v_1, \ldots, v_k]^T$ (denoted by $\vec{u} \preceq \vec{v}$) if and only if $\vec{u}$ is partially less than $\vec{v}$, i.e.,:

$$\forall i \in \{1, \ldots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \ldots, k\} : u_i < v_i \quad (5)$$

**Pareto Optimal Set:** For a given MOP $\vec{f}(\vec{x})$, the **Pareto Optimal Set** $P^*$ is defined by:

$$P^* := \{\vec{x} \in \Omega \mid \nexists \vec{x'} \in \Omega, \vec{f}(\vec{x'}) \preceq \vec{f}(\vec{x})\} \quad (6)$$

**Pareto Front:** For a given MOP $\vec{f}(\vec{x})$ and its Pareto optimal set $P^*$, the **Pareto Front** $PF^*$ is defined by:

$$PF^* := \{\vec{f}(\vec{x}) \mid \vec{x} \in P^*\} \quad (7)$$

When plotted in objective space, the non-dominated vectors are collectively known as the **Pareto front**.

## III. Cooperative Coevolution

In nature, coevolution is the process of reciprocal genetic change in one species, or group, in response to another. The original framework of cooperative coevolution (CC) utilized within evolutionary algorithms was originally introduced by Potter and De Jong [1]. This framework uses a divide-and-conquer approach to split the decision variables into subpopulations of smaller size, so that each of these subpopulations is optimized with a separate EA. The original CC framework can be summarized as follows:

1) Decompose an objective vector into $m$ low dimensional subcomponents.
2) Set $j = 1$ to start a new cycle.
3) Optimize the $j$-th subcomponent with a certain EA for a predefined number of fitness evaluations (FEs).
4) If $j < m$ then $j++$, and go to Step 3.
5) Stop if the stopping criteria are satisfied; otherwise go to Step 2 for the next cycle.

A cycle consists of one complete evolution of all species and the main idea is to decompose a high-dimensional problem into several low-dimensional subcomponents and evolve these subcomponents cooperatively for a predefined number of cycles. CC has shown to be a good framework for solving optimization problems [7], [8]. Since the cooperative coevolutionary framework has been extended to multi-objective optimization, a number of approaches have been proposed incorporating the CCA framework to improve the performance of multi-objective EAs. This is evidenced by MOCCGA [2], which integrates cooperative coevolution with Fonseca and Fleming's MOGA [9]. MOCCGA uses a dominance rank for individuals, in which a count of the number of individuals dominating others is the fitness criterion. In MOCCGA, the objectives are evaluated twice for each individual both with the best ranked individuals from each subpopulation, as well as with randomly selected individuals. This follows the approach described by Potter and De Jong, which aims to decrease the premature convergence observed on some test problems adopted with the original CC framework. In MOCCGA, the subcomponents are ranked only within the same subpopulation. It is important to mention that the number of evaluations adopted by the authors is not reported anywhere in the paper. Another approach is presented in [3], where a cooperative coevolutionary algorithm for multi-objective optimization is presented. This algorithm subdivides the decision variable space and determines which portions of the decision variables intervals are being used and discards portions of the intervals that it deems that are not being used by the search process. It also subdivides intervals so that separate sub-populations can operate on the portions of these intervals which contribute to the search. Sub-populations which are not making contributions are eliminated from the search. Parmee and Watson [10] proposed a collaborative multi-objective optimization scheme for the preliminary design of airframes. Here, they use one population to optimize each of the objective functions of the problem. The method utilizes individual genetic algorithms (GAs) for the optimization of each objective in order to reduce the problem to a number of concurrent co-evolutionary tasks specific to the overall design domain. Iorio and Li used NSGA-II [11] to evolve each species in their NSCCGA [12]. This Algorithm uses Potter and De Jong's scheme [1] where there are as many species as decision variables in the problem and each of them are evolved using NSGA-II. Individuals from each population are evaluated forming collaborations with randomly selected components from the best non-domination levels in the previous generation's sub-populations. Another coevolutionary approach is found in [13]. Here, a game theory based coevolutionary algorithm (GCEA) which makes use of the concept of Nash equilibrium [14] is presented. They used a

game matrix, and as an optimal solution of the game, the equilibrium state of this coevolutionary algorithm is to be found. Payoff of the game, for each population $G_i$, is calculated from the differences between two objective functions as follows:

$$G_1(v_i, v_i') = f_1(x) - f_2(x') \tag{8}$$
$$G_2(v_i, v_i') = f_2(x') - f_1(x) \tag{9}$$

From these payoffs, the fitness of each player is:

$$F_i = 100 * \frac{G_1(v_i, v_i')}{\alpha} \tag{10}$$
$$F_i' = 100 * \frac{G_2(v_i, v_i')}{\alpha} \tag{11}$$

where $\alpha$ is constant to normalize the fitness of $F_i$ or $F_i'$. From these settings, GCEA works as follows: first two populations are randomly generated. Then, the Player selected in the first population ($x$) plays with that from the second population ($x'$) and then he is paid off using (8). In the same manner, the Player in the second population is paid off using (9). The fitness of each player $F_n$ and $F_n'$ is updated using (10) and (11). This process is executed for all individuals of each population one by one. Each population is regenerated separately using genetic algorithms. The main drawback of this approach is that it works only for two-dimensional problems and the way the Nash equilibrium is used is very naive and static. There exist even more examples of the use of cooperative coevolution as a framework, but most of them work in a similar way as the ones we have described in this section.

In spite of the success of these approaches, there is evidence that shows how CCAs frequently perform poorly on what seem to be relatively simple problems, giving poor results or collapsing prematurely due to the loss in population diversity [4]. Here, we claim that cooperative coevolutionary systems are more attracted to points of robust resting balance than to ideal collaboration. CCAs have been shown to gravitate towards equilibrium states, regardless of whether or not such states correspond to an optimal solution for a given problem. *Relative overgeneralization* is one of the main causes for this behavior [15]. Relative overgeneralization occurs when coevolving populations are attracted to areas of the search space in which there are many strategies that perform well when combined with individuals from the other populations. This tends to generate individuals that have poor performance, which in turn can prevent the evolution of optimal or near-optimal solutions. It has also been studied how, through the process of Darwinian selection, a population of agents can evolve to an Evolutionary Stable Strategy (ESS) as introduced by Maynard Smith in 1982 [16]. He showed how a coevolutionary phenomenon reaches static states and that these states are ESSs in game theory. Also, from a mathematical point of view, coevolution has game theoretical properties and for that reason coevolution finally reaches the stable equilibrium state and this state is thought of as an optimal solution because of the dominance property of the game. Because of these properties, we assume that all these CCAs' pathologies are also present when dealing with MOPs and that there exist ways to alleviate them so that the performance of such algorithms can be improved.

The most logical approach to deal with some of the pathologies present in CCAs is to bias the CCAs toward searching for ideal collaborations. So, our aim is to combine the coevolutionary algorithm with evolutionary game theory in order to create a novel game theory based cooperative coevolutionary framework. So, here we propose to make use of the Nash equilibrium to define ideal collaborations that allow the CC framework to avoid *relative overgeneralization* (instead of looking for equilibrium states as the aforementioned GCEA does). With this, we aim to get a faster and more efficient CCA for multi-objective optimization.

## IV. Our Proposed Approach

Our approach to cooperative coevolution follows Potter's [1] model of cooperative coevolution where each population contains individuals that represent a particular component (in decision varibles space) of the problem, so that one member from each population is needed in order to assemble a complete solution. Evaluation of an individual from a particular population is performed by joining the individual with collaborating partners from other populations. Aside from evaluation, the populations are evolved independently. An abstract mathematical model for this system comes from evolutionary game theory (EGT). EGT is seen as a way of thinking about evolution at the phenotypic level when the fitness of particular individuals depend on their frequencies in the population [17]. As used by NSCCGA and MOCCGA, the common way to perform the evaluation of each individual is by taking one representative from the other populations that belongs to the best set of solutions found so far. We believe that this way of doing collaboration is the main cause for the aforementioned pathologies, because each species is choosing representatives from other populations having in mind only the performance of the species alone and not as a whole team among all of the species. This kind of interaction makes coevolution to only explore narrow regions of the collaboration space, which suggests that evolution is strongly attracted to certain regions of the search space. However, these regions do not necessarily correspond to (fitness-based) optimal solutions, and coevolution often converges to sub-optimal equilibria. Since in multi-objective optimization, objectives are in conflict, we believe that making use of Nash equilibrium for finding better collaborations is the best option, since it gives a solution of a non-cooperative game. According to Nash, each participant of the game has his own strategy set and objective function. Then, during the game, each player searches for the optimal strategy while other players' strategies are fixed. The game is conducted in this frame and when no player can further improve his criterion, the system is regarded as having reached a state of equilibrium, known as *Nash equilibrium*. This is exactly the way MOPs are managed by EAs, so this idea can

be applied to develop a novel CC framework, which, instead of looking for best collaborations from the point of view of an specific species, will look for ideal collaborations which take into account all of the objectives in order to perform better interactions among species. Our proposed approach is described next.

### A. Description of the proposed approach

Our approach works as follows: at the beginning, it divides the vector of decision variables $\vec{x}$ of dimension $D \in \mathbb{N}$ into $S \in \mathbb{N}$ subcomponents, where $S$ is equal to the number of objective functions in the problem. Each subcomponent is created from a random grouping of decision variables in order to increase the probability of grouping interacting variables in non-separable problems. At the same time, $S$ subpopulations (species) are created, each one with $NP$ individuals, and these $S$ subpopulations are assigned their corresponding decision variables in a random way. This means that to each subpopulation, it corresponds a subcomponent from $S$ which had been already created. Thus, every subpopulation will have a total of $m$ decision variables. At the same time, each species will be assigned with an specific objective function, so that there will be as many species as objective functions and each of them will do the fitness assignment of their individuals according to that, as will be described next. Once the subpopulations are created, the algorithm does a random initialization of all the individuals across all subpopulations. Then, the algorithm performs the *cycles* in which the evolution of each of the subpopulations is done for a given number of *generations*. This will continue until the stopping condition is reached, and at the end, the solutions that are globally non-dominated (i.e., with respect to all the subpopulations), constitute the outcome of the algorithm. The collaboration among the subpopulations takes place in the next way: in the first generation, random collaborations are formed and evaluated, obtaining a random individual from each subpopulation and forming a complete set of solutions to be evaluated in their objective functions. Then, the results from the evaluation are assigned back to the individual under evaluation. After the first generation, the resulting child subpopulations $Q_1$ to $Q_S$ will be evaluated by forming collaborations with individuals from the other species which are in a Nash equilibrium according to their fitness values. As mentioned before, the usual way to perform collaborations is by selecting representatives from other species which perform the best according to their respective subpopulation. However, we believe that this is the main cause for the tendency of CCAs to fall in ESS. So, we propose to find an ideal collaboration by making use of Nash equilibrium. Next, we give a brief description of such concept.

### B. Equilibriums for selection of strategies

The formal description of Nash equilibrium is presented next.

*1) Normal form game:* Let $P = \{1, \ldots, n\}$ be the set of players, $i \in P$, $a_x^i \in \Sigma^i$ be an element of the set of simple plays, and $s_x^i$ be a strategy of player $i$, $s_x^i \in S_i$; let $G = (S_1, \ldots, S_n; u_1, \ldots, u_n)$ be the game in normal form [14] where:

- A strategy is a sequence of actions $s_x^i = a_1^i \ldots a_n^i$.
- A strategy profile is an n-tuple of strategies $(s_1, \ldots, s_n)$; one strategy per player.
- $S_i$ is the set of strategies for the $i_{th}$ player.
- $\{S_1, \ldots, S_n\}$ is the set of all the $S_i$ strategies.
- $\{u_1, \ldots, u_n\}$ is the set of all payoff functions; one per player.
- $u_i(s_1, \ldots, s_n) = r$, where $(s_1, \ldots, s_n) \in S_1 \times \cdots \times S_n, r \in \mathbb{R}$.

*2) Nash equilibrium:* The Nash equilibrium [14] is a widely used mathematical concept, especially in the modelling of non-cooperative games. To identify the strategy profiles that fit the condition of Nash equilibrium, every strategy profile is evaluated with the payoff functions of the players. Then, the chosen profiles are those that, for every player, are the options that produce the lowest loss for each one, individually, in a non-cooperative way. In a more formal way, let $s_1^*, \ldots, s_n^*$ and $s_i^*$ be the non-cooperative player's strategies from $i$ to the $n-1$ other players' strategies. So $(s_1^*, \ldots, s_i^*, \ldots, s_n^*)$ fits the Nash equilibrium condition, if and only if it maximizes the corresponding payoff function:

$$u_i(s_1^*, \ldots, s_i^*, \ldots, s_n^*) \geq u_i(s_1^*, \ldots, s_i, \ldots, s_n^*) \quad \forall i \in P, s_i \in S_i. \tag{12}$$

Every strategy profile is a payoff function valued and is compared with all the others, to determine whether or not it is dominated. Given a strategy profile $x_1$ for each player $i$, the strategy profile is modified by altering the player's current strategy while keeping the strategies of the other $n-1$ players unchanged; if any deviation from $x_1$, evaluated by $u_i$, dominates it, that means that player $i$'s profit is higher by $u_i(x_2)$. So, $x_1$ is dominated by $x_2$'s profile and, therefore, $x_1$ is discarded. All the dominated profiles are discarded and the non-dominated profiles are the ones that fit the Nash equilibrium. Any game in (finite) normal form has at least one strategy profile that fits the Nash equilibrium [14]. Observe that in Nash equilibrium, every player is applying a non-cooperative perspective which turns out to be not as bad for him as the other players' strategies.

With all this in mind, we propose to make collaborations by finding the best unions according to the previous objective values found so far. For this purpose, we create all the combinations from the function of the individual being evaluated and the posible values given by the rest of the other populations. The way combinations are formed is depicted in Figure 1, in which we show an example of a three-dimensional MOP using 2 individuals for each population. Once we have this, we consider that for our purpose, each objective function value can be used as an strategy. With this, we can build a utility function from each species point of view, in order to create a non-cooperative game where each player is an species, and its interest is focused on its specific objective function. For this purpose, we adopted SPEA2's fitness assignment

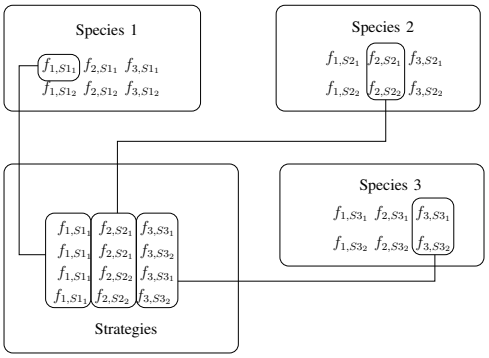


Fig. 1. Strategies creation for one individual in a CCA with 3 Species (objective functions) and 2 individuals in each subpopulation.

strategy [18] within each species. Fitness assignment operates as follows: having an individual $i$ which belongs to the $Q_j$ species being evolved, we will create all the combinations of the objectives from the current species with the objectives of the other species. Having these composed objective vectors we will evaluate each of these combinations according to the current species population. Thus, at the end we will have a set of strategies from which we will find the Nash equilibrium according to the fitness values that will be used as the utility functions values. So, the individuals which bring a Nash equilibrium will be selected from the subpopulations, $P_1$ to $P_S$, of the previous generation in order to collaborate. This will allow us to evaluate the new individual. The collaboration procedure is shown in Figure 2. The way Nash strategy profiles are obtained is shown in Algorithm 1.

**Algorithm 1** Nash equilibrium algorithm for selection of strategies for CC collaborations.

**Input:** Input each strategy profile and its payoff value
**for** all $x = (x_1, \ldots, x_m)$ strategy profiles **do**
  **for** all player $i = (1, \ldots, n)$ **do**
    **if** $x$ is labeled as non-dominated **then**
      Do the derivations in $x$ for player $i$
      **if** $x$ is dominated by at least one derivation of $i$ **then**
        labeled $x$ as dominated {move to the next strategy profile}
      **end if**
    **else**
      {move to the next strategy profile}
    **end if**
  **end for**
**end for**

The algorithm iterates until some termination condition is fulfilled (usually when a certain predefined number of cycles is reached). At the end, we get the non-dominated solutions

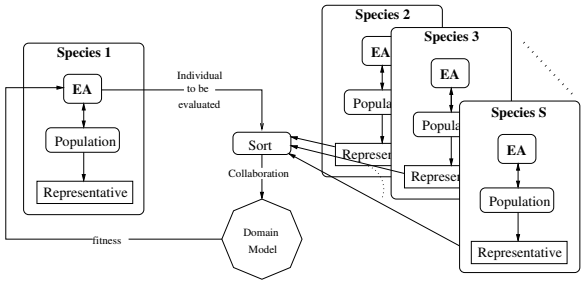


Fig. 2. Cooperative coevolutionary collaboration architecture from the perspective of species number 1. Here, we assume that we have $S$ species, where the representative of each species for collaboration is the one that fits a Nash equilibrium.

from the non-dominated individuals of each subpopulations, in order to obtain a final set of solutions for the problem being solved. A summary of the way in which our approach works is presented in Algorithm 2.

**Algorithm 2** Cooperative Coevolutionary Framework

**Input:** $NP$, $Cycles$, $Gmax$, $NumEsp$
**Output:** $SolutionSet$
  $Pobs \leftarrow Populations(NP, NumEsp)$
  $InitializeSpecies(Pobs)$
  **for** $j \leftarrow 1$ **to** $Cycles$ **do**
    **for** $i \leftarrow 1$ **to** $NumEsp$ **do**
      **for** $k \leftarrow 1$ **to** $Gmax$ **do**
        $MOEA(Pobs[i])$
      **end for**
    **end for**
  **end for**
  $SolutionSet \leftarrow ObtainNonDominatedSet(Pobs)$
  **return** $SolutionSet$

### C. Experimental Studies

For the purposes of this study, we adopted the Zitzler-Deb-Thiele (ZDT) [19] and the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suites [20] with two and three objectives, respectively.

### D. Methodology

Since the main objective of this work is to evaluate the performance of our approach in terms of efficiency when solving MOPs, we will analyze our results with respect to those of the NSCCGA [12] and GCEA [13]. For this sake, we established a predefined number of function evaluations that the algorithms can use, to analyze how they behave with the same resources. For measuring the results we adopted the hypervolume performance indicator [21]. The hypervolume is obtained by computing the volume (in objective function space) of the non-dominated set of solutions $Q$ that minimize a MOP. For every solution $i \in Q$, a hypercube $v_i$ is generated with a reference point $W$ and the solution $i$ as its diagonal corner of the hypercube. The reference point $W$ can be generated by building a vector of worst possible objective

function values. Then, the hypervolume (HV) is computed as the union of all the found hypercubes as follows:

$$HV = volume\left(\bigcup_{i=1}^{|Q|} v_i\right) \tag{13}$$

The aim of this study is to identify which of the MOEAs being compared is able to get closer to the true Pareto front using the same number of objective function evaluations.

### *E. Parameterization*

The parameters of each MOEA used in our study were chosen in such a way that we could do a fair comparison among them. Although all the algorithms used for our comparative study are CCAs, the specific nature of each of them requires different parameters. Thus, for our approach and for GCEA there will be as many species as objective functions, whereas for NSCCGA there are as many species as decision variables. For all the algorithms we used a small population size of 16 individuals for each subpopulation (species). This was done, in order to set an environment where the aforementioned pathologies arise. For the ZDT test suite, we used 67 cycles and the number of species for NSCCGA was set to 30, since that is the number of decision variables for these problems. The exception is ZDT4 where the number of decision variables is 10. For this problem, we used 200 cycles. For our approach (NashCC) and GCEA, the number of species was set to 2, because the ZDT problems have 2 objectives. The number of cycles was set to 1000. For the DTLZ test suite, we could only compare results against CCNSGA since GCEA is not able to scale to more than two objectives. The number of species for NSCCGA was set to 12, since the DTLZ problems use by default that number of decision variables. In this case, 1250 cycles were used. For our proposed NashCC, the number of species was set to 3, since these problems have 3 objectives. The number of cycles was set to 5000. For all algorithms and problems, the distribution indexes for the SBX and polynomial-based mutation operators [11], were set as: $\eta_c = 20$ and $\eta_m = 20$, respectively. The crossover probability was set to $p_c = 0.9$ and the mutation probability was set to $p_m = 0.01$. Finally, we used just one generation for each species per cycle for all approaches. All of this in order to use the same number of function evaluations in all CCAs and to allow for a fair comparison of results.

### *F. Analysis of results*

In our experiments, we obtained the hypervolume value over the 25 independents runs performed. Tables I and II show results of the hypervolume measure for the ZDT and DTLZ test suites, respectively, as well as the reference points used for each of the problems. To ease the analysis of the results in these tables, the cells containing the best hypervolume value for each problem have a grey colored background.

From Figures 3 to 14, we plot the results of the median of the 25 runs. These plots are shown for the ZDT and the DTLZ test problems, respectively. We can observe that, using the same number of function evaluations, our proposed NashCC is able to get closer than NSCCGA and GCEA to the Pareto front in all the problems. It is clear that NashCC is much faster than the other two algorithms in terms of number of evaluations. As can be observed, NSCCGA and GCEA have premature converge in most problems, which makes them to fall in false fronts in the case of multi-frontal problems such as ZDT4. This confirms that the tendency of CCAs to fall into ESS is also present when dealing with MOPs and that looking for good collaborations is an effective way to alleviate this problem.

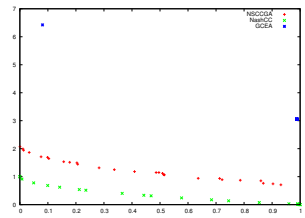


Fig. 3. Plot of CCAs for ZDT1.

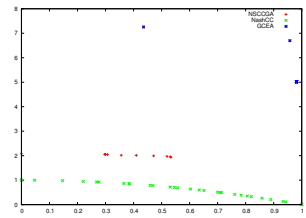


Fig. 4. Plot of CCAs for ZDT2.

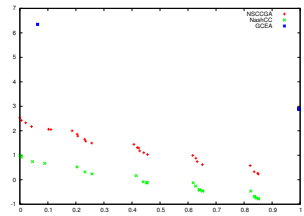


Fig. 5. Plot of CCAs for ZDT3.

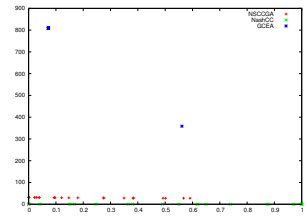


Fig. 6. Plot of CCAs for ZDT4.

## V. Conclusions and Future Work

This paper proposes a new cooperative coevolutionary framework for solving MOPs using a novel cooperation strategy based on the Nash equilibrium. With this, we presented

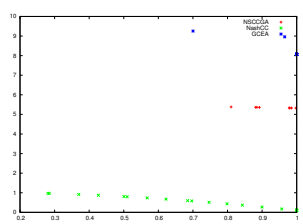


Fig. 7. Plot of CCAs for ZDT6.

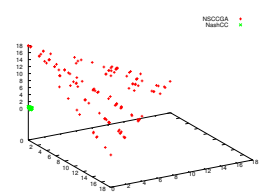


Fig. 8. Plot of CCAs for DTLZ1.

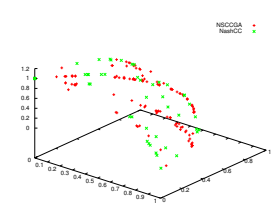


Fig. 9. Plot of CCAs for DTLZ2.

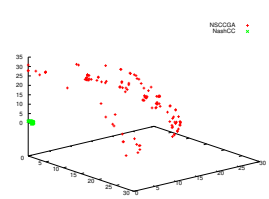


Fig. 10. Plot of CCAs for DTLZ3.

TABLE I
HYPERVOLUME VALUES FOR THE ZDT TEST SUITE

| | ZDT1 | | | ZDT2 | | | ZDT3 | | | ZDT4 | | | ZDT6 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | NashCC | CCNSGA | CCGT | NashCC | CCNSGA | CCGT | NashCC | CCNSGA | CCGT | NashCC | CCNSGA | CCGT | NashCC | CCNSGA | CCGT |
| Ref. Points | (1 8) | | | (1 9) | | | (1 9) | | | (1 900) | | | (1 10) | | |
| Best | 7.6285 | 7.0870 | 5.4413 | 8.3108 | 7.8108 | 2.5263 | 9.0247 | 8.4213 | 4.2544 | 899.6179 | 890.5287 | 532.1361 | 6.7281 | 3.7310 | 0.7373 |
| | 7.6269 | 7.0843 | 4.0366 | 8.3106 | 7.7170 | 2.5197 | 9.0188 | 8.4140 | 3.9618 | 899.6039 | 889.9144 | 517.7009 | 6.7246 | 3.7002 | 0.4183 |
| | 7.6268 | 7.0667 | 3.3556 | 8.3089 | 7.7149 | 2.4991 | 9.0185 | 8.3796 | 3.7482 | 899.5833 | 886.6908 | 484.3817 | 6.7196 | 3.4643 | 0.4033 |
| | 7.6264 | 7.0632 | 3.0911 | 8.3082 | 7.6785 | 2.4911 | 9.0146 | 8.3700 | 3.7173 | 899.4977 | 885.5299 | 454.5236 | 6.7187 | 3.3439 | 0.3968 |
| | 7.6256 | 7.0305 | 3.0564 | 8.3070 | 7.6688 | 2.4911 | 9.0141 | 8.3097 | 3.7161 | 899.4590 | 884.8965 | 443.1219 | 6.7186 | 3.3223 | 0.3579 |
| | 7.6253 | 6.9974 | 2.9899 | 8.3053 | 7.6135 | 2.4712 | 9.0137 | 8.2885 | 3.6335 | 899.4424 | 884.8232 | 425.7738 | 6.7181 | 3.3147 | 0.3444 |
| | 7.6244 | 6.9758 | 2.9129 | 8.3012 | 7.6082 | 2.0796 | 9.0119 | 8.2882 | 3.5062 | 899.4239 | 884.8077 | 420.1604 | 6.7163 | 3.3110 | 0.3347 |
| | 7.6241 | 6.9534 | 2.9070 | 8.3010 | 7.5968 | 2.0732 | 9.0112 | 8.2664 | 3.4547 | 899.2720 | 884.7901 | 418.4893 | 6.7094 | 3.2766 | 0.3298 |
| | 7.6239 | 6.9454 | 2.6500 | 8.2991 | 7.5676 | 1.8883 | 9.0047 | 8.2413 | 3.4528 | 899.2685 | 884.3186 | 415.4923 | 6.7057 | 3.2499 | 0.3197 |
| | 7.6233 | 6.9338 | 2.5859 | 8.2987 | 7.5648 | 1.8712 | 9.0040 | 8.1980 | 3.4151 | 899.2629 | 880.8902 | 413.5572 | 6.7055 | 3.2174 | 0.3132 |
| | 7.6228 | 6.9297 | 2.5426 | 8.2982 | 7.5565 | 1.7314 | 9.0035 | 8.1663 | 3.2693 | 899.2607 | 879.9972 | 411.7965 | 6.6999 | 3.1605 | 0.2874 |
| | 7.6226 | 6.9118 | 2.5214 | 8.2964 | 7.5375 | 1.7263 | 9.0001 | 8.1254 | 3.2510 | 899.2540 | 878.8118 | 410.1775 | 6.6987 | 3.0051 | 0.2680 |
| Median | 7.6225 | 6.9105 | 2.4754 | 8.2944 | 7.5193 | 1.6522 | 8.9982 | 8.0892 | 3.2009 | 899.2539 | 878.0266 | 375.7467 | 6.6984 | 2.9893 | 0.2642 |
| | 7.6212 | 6.9095 | 2.2863 | 8.2943 | 7.4892 | 1.6327 | 8.9924 | 8.0647 | 3.0842 | 899.2509 | 877.0857 | 366.9912 | 6.6962 | 2.9829 | 0.2334 |
| | 7.6210 | 6.8809 | 2.1792 | 8.2940 | 7.4726 | 1.6086 | 8.9717 | 8.0465 | 2.9459 | 899.2400 | 877.0416 | 366.0214 | 6.6891 | 2.8825 | 0.2130 |
| | 7.6198 | 6.8771 | 2.0903 | 8.2934 | 7.4721 | 1.5961 | 8.9706 | 7.9992 | 2.9135 | 899.2394 | 876.6987 | 338.2662 | 6.6840 | 2.8581 | 0.1778 |
| | 7.6188 | 6.8707 | 2.0473 | 8.2934 | 7.4207 | 1.5661 | 8.9697 | 7.9949 | 2.8526 | 899.2325 | 876.6007 | 336.7565 | 6.6812 | 2.7485 | 0.1693 |
| | 7.6185 | 6.8668 | 2.0357 | 8.2921 | 7.3748 | 1.5278 | 8.9682 | 7.9943 | 2.8100 | 899.0523 | 876.0462 | 333.4452 | 6.6667 | 1.9087 | 0.0928 |
| | 7.6183 | 6.8500 | 2.0352 | 8.2920 | 7.3586 | 1.4419 | 8.9649 | 7.9875 | 2.7137 | 899.0446 | 873.8957 | 330.1884 | 6.6563 | 1.9012 | 0.0798 |
| | 7.6158 | 6.8195 | 2.0239 | 8.2908 | 7.3559 | 1.4396 | 8.9584 | 7.9737 | 2.6660 | 899.0429 | 873.8497 | 326.3380 | 6.6543 | 1.8504 | 0.0503 |
| | 7.6153 | 6.8105 | 2.0183 | 8.2904 | 7.3497 | 1.3578 | 8.9512 | 7.9686 | 2.6425 | 899.0254 | 872.5329 | 325.2272 | 6.6456 | 1.7674 | 0.0367 |
| | 7.6152 | 6.7614 | 1.9923 | 8.2893 | 7.2987 | 1.3367 | 8.9507 | 7.9513 | 2.5738 | 898.8857 | 871.4517 | 322.9623 | 6.6377 | 1.7088 | 0.0343 |
| | 7.6138 | 6.7308 | 1.9235 | 8.2888 | 7.2484 | 1.1220 | 8.9482 | 7.9195 | 2.5155 | 898.8794 | 870.6939 | 306.6790 | 6.6339 | 0.9597 | 0.0243 |
| | 7.6112 | 6.6191 | 1.8730 | 8.2830 | 7.2318 | 1.0963 | 8.9120 | 7.8683 | 2.4955 | 898.8418 | 866.3119 | 290.3387 | 6.6194 | 0.9265 | 0.0150 |
| Worst | 7.5758 | 6.6184 | 1.4873 | 8.2790 | 7.0012 | 1.0383 | 8.2848 | 7.8366 | 2.4719 | 898.2626 | 857.3100 | 281.6815 | 6.5977 | 0.8791 | 0.0059 |
| Average | 7.6195 | 6.9002 | 2.5823 | 8.2968 | 7.4891 | 1.7914 | 8.9592 | 8.1265 | 3.1707 | 899.2079 | 878.5418 | 385.9181 | 6.6849 | 2.6584 | 0.2363 |
| STD | 0.0102 | 0.1273 | 0.8232 | 0.0084 | 0.1833 | 0.4819 | 0.1435 | 0.1794 | 0.5050 | 0.2949 | 7.6306 | 68.3190 | 0.0363 | 0.8878 | 0.1727 |

TABLE II
HYPERVOLUME VALUES FOR THE DTLZ TEST SUITE

| | DTLZ1 | | DTLZ2 | | DTLZ3 | | DTLZ4 | | DTLZ5 | | DTLZ6 | | DTLZ7 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | NashCC | CCNSGA | NashCC | CCNSGA | NashCC | CCNSGA | NashCC | CCNSGA | NashCC | CCNSGA | NashCC | CCNSGA | NashCC | CCNSGA |
| Ref. Points | (35 35 35) | | (1.2 1.2 1.2) | | (50 50 50) | | (1.2 1.2 1.2) | | (1.2 1.2 1.2) | | (7 7 7) | | (7 7 7) | |
| Best | 42876.2811 | 42861.9638 | 1.0987 | 1.0976 | 124999.3152 | 124975.9658 | 1.0628 | 0.7717 | 0.7180 | 0.7027 | 334.8787 | 323.5990 | 498.3214 | 469.9119 |
| | 42874.8800 | 42847.0050 | 1.0971 | 1.0928 | 124999.1670 | 124865.7086 | 1.0294 | 0.7716 | 0.7172 | 0.7019 | 334.8272 | 319.1156 | 498.0460 | 449.7765 |
| | 42874.8800 | 42819.1787 | 1.0963 | 1.0926 | 124997.6336 | 124715.7241 | 0.9992 | 0.7691 | 0.7169 | 0.7018 | 334.7990 | 315.1843 | 497.8648 | 430.2499 |
| | 42874.8431 | 42804.0798 | 1.0958 | 1.0925 | 124994.5279 | 124655.8189 | 0.9992 | 0.7685 | 0.7169 | 0.6986 | 334.7064 | 311.0600 | 497.5445 | 417.8822 |
| | 42874.8248 | 42742.8616 | 1.0956 | 1.0922 | 124994.3927 | 124623.6335 | 0.9992 | 0.7681 | 0.7169 | 0.6977 | 334.6832 | 310.9483 | 497.3881 | 415.6720 |
| | 42874.7272 | 42707.8119 | 1.0924 | 1.0896 | 124993.7485 | 124565.9743 | 0.9991 | 0.7671 | 0.7165 | 0.6969 | 334.6208 | 309.8232 | 497.3302 | 400.1097 |
| | 42874.5948 | 42596.9558 | 1.0912 | 1.0886 | 124991.6194 | 124355.8553 | 0.9991 | 0.7646 | 0.7164 | 0.6967 | 334.5886 | 304.8691 | 497.1576 | 398.6594 |
| | 42874.5795 | 42525.9315 | 1.0897 | 1.0877 | 124990.9612 | 123498.8561 | 0.9591 | 0.2880 | 0.7164 | 0.6962 | 334.5622 | 302.2213 | 496.9787 | 392.3261 |
| | 42873.8784 | 42501.7135 | 1.0890 | 1.0864 | 124990.5698 | 123367.9406 | 0.7627 | 0.2880 | 0.7161 | 0.6962 | 334.4871 | 294.6850 | 496.4218 | 390.9044 |
| | 42873.8354 | 42490.1736 | 1.0885 | 1.0858 | 124990.2125 | 122602.1798 | 0.7627 | 0.2880 | 0.7155 | 0.6944 | 334.4834 | 286.5837 | 496.2432 | 388.2298 |
| | 42872.9553 | 42440.4895 | 1.0858 | 1.0857 | 124989.1850 | 122528.6792 | 0.7627 | 0.2880 | 0.7153 | 0.6934 | 334.4430 | 286.5616 | 496.2147 | 383.3258 |
| | 42872.1874 | 42395.1893 | 1.0855 | 1.0817 | 124988.6998 | 122354.9181 | 0.7621 | 0.2880 | 0.7149 | 0.6929 | 334.3872 | 278.2876 | 494.8938 | 382.6247 |
| Median | 42872.1758 | 42252.9449 | 1.0850 | 1.0815 | 124988.2743 | 122249.1144 | 0.7621 | 0.2880 | 0.7148 | 0.6917 | 334.3602 | 271.4169 | 494.8744 | 382.3463 |
| | 42872.0777 | 42251.2821 | 1.0843 | 1.0801 | 124986.7070 | 121018.5017 | 0.7621 | 0.2880 | 0.7146 | 0.6912 | 334.3265 | 260.2275 | 494.8650 | 381.6034 |
| | 42872.0042 | 42240.1745 | 1.0841 | 1.0798 | 124984.5723 | 120204.4051 | 0.7584 | 0.2880 | 0.7138 | 0.6911 | 334.1917 | 256.5397 | 494.8056 | 379.4380 |
| | 42871.5896 | 42136.6736 | 1.0829 | 1.0772 | 124981.2659 | 120090.9469 | 0.7584 | 0.2880 | 0.7133 | 0.6907 | 334.1842 | 255.3988 | 494.7458 | 378.7837 |
| | 42871.4296 | 42041.5760 | 1.0821 | 1.0759 | 124979.7506 | 118446.6267 | 0.7584 | 0.2880 | 0.7130 | 0.6903 | 334.1492 | 252.3305 | 494.7171 | 373.3755 |
| | 42870.9092 | 42000.2356 | 1.0812 | 1.0747 | 124978.5341 | 111578.2296 | 0.7578 | 0.2880 | 0.7126 | 0.6892 | 334.0760 | 249.2532 | 494.6616 | 369.2247 |
| | 42870.6630 | 41934.0698 | 1.0807 | 1.0702 | 124977.9124 | 111355.0424 | 0.7578 | 0.2880 | 0.7121 | 0.6884 | 334.0601 | 243.9649 | 494.6347 | 368.6373 |
| | 42867.1791 | 41635.6971 | 1.0795 | 1.0674 | 124964.3519 | 110351.6186 | 0.7578 | 0.2880 | 0.7121 | 0.6876 | 334.0141 | 243.1744 | 494.5831 | 364.3375 |
| | 42859.6049 | 41503.9476 | 1.0768 | 1.0611 | 124959.8377 | 110256.5113 | 0.7461 | 0.2880 | 0.7121 | 0.6842 | 334.0050 | 241.0774 | 494.5474 | 361.4942 |
| | 42858.2116 | 41479.6057 | 1.0746 | 1.0605 | 124950.1114 | 95709.7965 | 0.7461 | 0.2880 | 0.7111 | 0.6827 | 333.9971 | 235.3766 | 494.4941 | 353.6972 |
| | 42853.2824 | 40617.4194 | 1.0725 | 1.0592 | 124943.0481 | 89891.5783 | 0.7461 | 0.2880 | 0.7108 | 0.6773 | 333.9491 | 221.5874 | 494.2884 | 352.4184 |
| | 42845.1465 | 38400.6563 | 1.0718 | 1.0562 | 124913.3725 | 73378.8013 | 0.7461 | 0.2880 | 0.7092 | 0.6765 | 333.9080 | 202.3386 | 494.0993 | 335.0071 |
| Worst | 42780.3090 | 32070.8457 | 1.0703 | 1.0483 | 124894.8008 | 55886.0385 | 0.7451 | 0.2843 | 0.7077 | 0.6760 | 333.0167 | 128.6456 | 494.0689 | 327.5270 |
| Average | 42866.28 | 41691.94 | 1.09 | 1.08 | 124976.90 | 113901.14 | 0.84 | 0.42 | 0.71 | 0.69 | 334.31 | 268.17 | 495.75 | 385.90 |
| STD | 19.4845 | 2210.2745 | 0.0083 | 0.0133 | 26.4582 | 17546.1981 | 0.1202 | 0.2204 | 0.0027 | 0.0076 | 0.4015 | 44.3675 | 1.4275 | 32.6220 |

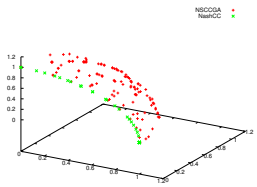


Fig. 11. Plot of CCAs for DTLZ4.

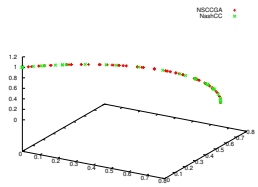


Fig. 12. Plot of CCAs for DTLZ5.

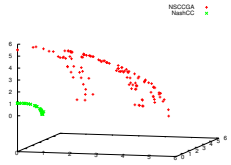


Fig. 13. Plot of CCAs for DTLZ6.

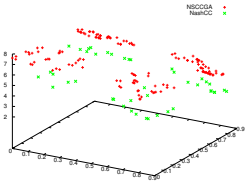


Fig. 14. Plot of CCAs for DTLZ7.

a novel cooperative coevolutionary MOEA, called NashCC, which was shown to be able to successfully deal with the ZDT and DTLZ test functions. We have studied the convergence rate of our proposed NashCC with respect to that of NSCCGA and GCEA. The results confirmed that our proposed approach outperforms the other two CCAs and that the collaboration framework has a great impact in CCAs. We confirm that the tendency of CCAs to fall into ESS is also present when dealing with MOPs and that looking for ideal collaborations is a good way to alleviate this problem. As part of our future work we aim to reduce the computational cost of the Nash equilibrium method required by our algorithm in order to be able to scale it to more objectives.

## Acknowledgments

The first author acknowledges support from CONACyT through a scholarship to pursue graduate studies at Computer Science Department of CINVESTAV-IPN. The second author gratefully acknowledges support from CONACyT project no. 221551.

## References

[1] M. A. Potter and K. A. D. Jong, "A Cooperative Coevolutionary Approach to Function Optimization," in *Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature*, ser. PPSN III. London, UK, UK: Springer-Verlag, 1994, pp. 249–257.

[2] N. Keerativuttiumrong, N. Chaiyaratana, and V. Varavithya, "Multi-objective Co-operative Co-evolutionary Genetic Algorithm," in *Parallel Problem Solving from Nature—PPSN VII*, J. J. Merelo Guervós, P. Adamidis, H.-G. Beyer, J.-L. F.-V. nas, and H.-P. Schwefel, Eds. Granada, Spain: Springer-Verlag. Lecture Notes in Computer Science No. 2439, September 2002, pp. 288–297.

[3] C. A. Coello Coello and M. Reyes Sierra, "A Coevolutionary Multi-Objective Evolutionary Algorithm," in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, vol. 1. Canberra, Australia: IEEE Press, December 2003, pp. 482–489.

[4] R. P. Wiegand, "An analysis of cooperative coevolutionary algorithms," Ph.D. dissertation, Fairfax, VA, USA, 2004, aAI3108645.

[5] L. Panait, R. Wiegand, and S. Luke, "A sensitivity analysis of a cooperative coevolutionary algorithm biased for optimization," in *Genetic and Evolutionary Computation - GECCO 2004*, ser. Lecture Notes in Computer Science, K. Deb, Ed. Springer Berlin Heidelberg, 2004, vol. 3102, pp. 573–584.

[6] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Kluwer Academic Publishers, May 2002, iSBN 0-3064-6762-3.

[7] X. Y. Zhenyu Yang, Ke Tang, "Differential evolution for high-dimensional function optimization," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, vol. 1, sept. 2007.

[8] X. Y. Mohammad Nabi, Zhenyu Yang, "Cooperative co-evolution for large scale optimization through more frequent random grouping," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, vol. 1, sept. 2010, pp. 1– 8 Vol.1.

[9] C. M. Fonseca and P. J. Fleming, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization," in *Proceedings of the Fifth International Conference on Genetic Algorithms*, S. Forrest, Ed., University of Illinois at Urbana-Champaign. San Mateo, California: Morgan Kauffman Publishers, 1993, pp. 416–423.

[10] I. Parmee and A. H. Watson, "Preliminary Airframe Design using Co-Evolutionary Multiobjective Genetic Algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, Eds., vol. 2. Orlando, Florida, USA: Morgan Kaufmann, 13-17 July 1999, pp. 1657–1665.

[11] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II," Indian Institute of Technology, Kanpur, India, KanGAL report 200001, 2000.

[12] A. W. Iorio and X. Li, "A Cooperative Coevolutionary Multiobjective Algorithm Using Non-dominated Sorting," in *Genetic and Evolutionary Computation–GECCO 2004. Proceedings of the Genetic and Evolutionary Computation Conference. Part I*, K. D. et al., Ed. Seattle, Washington, USA: Springer-Verlag, Lecture Notes in Computer Science Vol. 3102, June 2004, pp. 537–548.

[13] K.-B. Sim, D.-W. Lee, and J.-Y. Kim, "Game theory based coevolutionary algorithm: A new computational coevolutionary approach," *INTERNATIONAL JOURNAL OF CONTROL AUTOMATION AND SYSTEMS*, vol. 2, pp. 463–474, 2004.

[14] J. Nash, "Non-cooperative games," *The annals of Mathematics*, vol. 54, no. 2, pp. 286–295, 1951.

[15] L. Panait, R. Wiegand, and S. Luke, "A visual demonstration of convergence properties of cooperative coevolution," in *Parallel Problem Solving from Nature - PPSN VIII*, ser. Lecture Notes in Computer Science, X. Yao, E. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervs, J. A. Bullinaria, J. E. Rowe, P. Tino, A. Kabn, and H.-P. Schwefel, Eds. Springer Berlin Heidelberg, 2004, vol. 3242, pp. 892–901.

[16] J. Maynard-smith and G. R. PRICE, "The Logic of Animal Conflict," *Nature*, vol. 246, pp. 15–18, 1973.

[17] T. L. Vincent and J. S. Brown, *Evolutionary game theory, natural selection, and Darwinian dynamics*. Cambridge, New York: Cambridge University Press, 2005, autre tirage : 2007. [Online]. Available: http://opac.inria.fr/record=b1120057

[18] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, Tech. Rep. 103, May 2001.

[19] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, Tech. Rep. 70, December 1999.

[20] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Tech. Rep., 2001.

[21] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, November 1999.