Multi-objective Coevolution and Decision-making for Cooperative and Competitive Environments

Anirudh Suresh, Jaturong Kongmanee, Kalyanmoy Deb, Vishnu Boddeti
Michigan State University
East Lansing, USA
{suresha2, kongmane, kdeb, vishnu}@msu.edu

***Abstract*—Co-evolutionary algorithms involve two co-evolving populations, each having its own set of objectives and constraints, and interact with each other during function evaluation. Co-evolutionary algorithms are of great interest in cooperative and competing games and search tasks in which multiple agents having different interests are in play. Despite a number of single-objective co-evolutionary studies, there has been limited interest in multi-objective co-evolutionary algorithms. A recent study has revealed that in addition to the challenges associated with the development of an efficient algorithm, a proper understanding of the conflicting objectives within a single population and their interaction among objectives of the second population becomes extremely difficult to comprehend. In this paper, we extend the previous proof-of-principle multi-objective co-evolutionary (MoCoEv) study in three important directions. First, we enhance MoCoEv's ability to handle mixed cooperating and conflicting scenarios among different players. Second, we propose an iterative multi-criterion decision-making (MCDM) approach to demonstrate how, in an arms-race type scenario, the most appropriate solution can be selected from the obtained Pareto-optimal solution set iteratively. Third, we extend the previous MoCoEv algorithm with a many-objective evolutionary algorithm (NSGA-III) to make them applicable to three or more objectives for each player. These three developments reveal better insights about the intricate issues related to multiple objectives and decision-making for co-evolutionary optimization and take MoCoEv a step closer to solving more complex multi-player problems.**

***Index Terms*—Co-evolutionary algorithm, multi-objective optimization,multi-criterion decision making, cooperative and competitive co-evolution.**

I. Introduction

Multi-agent games and search problems involve two or more agents (or players). Computational methods to arrive at suitable strategy for an agent to solve such tasks intricately depends on the strategies of other agents. In terms of search problems involving multiple agents, a solution of one agent requires solutions of other agents before its objectives and associated constraints can be evaluated. Thus, an evolution of a population of solutions of one agent intricately depend on the evolution of solutions of other populations. One of the ways to tackle such problems is the use of co-evolutionary algorithms [1]. We restrict our study here on a two-agent system involving co-evolution of two interacting populations.

When a single objective (or criterion) is used to evolve a population for each agent, the task is referred to as "single-objective co-evolutionary (SoCoEv) algorithms". The objectives are usually different for different agents. While a number of such studies exist [2], [3], consideration of multiple conflicting objectives for each agent is an important aspect, but has not been studied much. Some recent studies have focused on obtaining objective trade-off solutions for multi-objective games (MOGs) and have proposed solution concepts based on rationality from game theory [4]–[7]. In these methods, worst case performance, called anti-optimal fronts, are used to evaluate and evolve rational strategies. These anti-optimal fronts are later used for multi-objective decision making (MCDM) [8]. Eisenstadt et al. proposed a co-evolutionary method to solve for rational strategies in an MOG. Żychowski et al. [9] proposed a memetic co-evolutionary method to solve MOGs with expensive function evaluations using the nadir point (worst case) as an aggregation method. Though these methods show promising results, multi-objective co-evolutionary methods have not been understood enough from the perspective of optimization or decision making.

In an earlier study [10], we have referred to them as multi-objective co-evolutionary (MoCoEv) algorithms. In MoCoEv problems, each agent's target is to find a set of Pareto-optimal (PO) solutions, making a trade-off between two or more conflicting objectives associated with the agent's own objectives. Due to linking of multiple populations, the PO solutions of one agent will intricately depend on the PO solutions of other populations, thereby making the search for such PO solutions and a clear understanding of their interactions a challenge.

In this paper, we investigate and extend our earlier MoCoEv algorithm, developed for two objectives for each agent using NSGA-II [11], in three directions. We restrict our study for two agents only. First, we extend MoCoEv algorithm to tackle mixed competing and cooperating scenarios. The previous study considered competing scenario among objectives of both populations. If one agent minimizes two objectives (f_1 and f_2), the second agent maximizes the same two objectives or optimizes other objectives that produces completely different PO solution set for the the first agent, thereby creating a competing scenario. Here, we consider a competing-cooperating scenario in which the second agent may be interested in minimizing f_1 (the same goal as the first agent, or a function that requires similar PO solutions of the first agent as companion PO solutions of the second agent), but performs in a conflicting sense for f_2. The competing or cooperating scenarios will make a change in the PO solutions. Next, we propose an iterative decision-making strategy in which agents choose a strategy one after the other in an iterative manner but strategies are chosen from PO solutions obtained from the MoCoEv task. This will simulate the arms race scenario or iterative strategies usually involved in an attacker-defender system. Third, we extend MoCoEv's ability to handle more than two objectives for each agent by integrating an evolutionary many-objective optimization algorithm (NSGA-III [12]) within MoCoEv framework. Each of these enhancements are demonstrated on a Tug of War problem introduced in another study [4].

In the remainder of the paper, the two-objective MoCoEv algorithm is introduced in brief in Section II. Three proposed extensions of the MoCoEv algorithm and its working are elaborated in Section III. Results of three extensions are presented in Section IV. Finally, conclusions and certain future studies are highlighted in Section V.

II. Existing Multi-objective Co-evolutionary Algorithm (MoCoEv)

A typical multi-objective co-evolutionary problem is can be defined as:

Problem P_1:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \left(f_1^{(1)}(\mathbf{x},\mathbf{y}),\ldots,f_{M_1}^{(1)}(\mathbf{x},\mathbf{y})\right), \\ s.t. \quad & g_j^{(1)}(\mathbf{x},\mathbf{y}) \leq 0,\ j=1,\ldots,J_1, \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)},\ i=1,\ldots,n_1. \end{aligned} \tag{1}$$

Problem P_2:

$$\begin{aligned} \min_{\mathbf{y}} \quad & \left(f_1^{(2)}(\mathbf{x},\mathbf{y}),\ldots,f_{M_2}^{(2)}(\mathbf{x},\mathbf{y})\right), \\ s.t. \quad & g_j^{(2)}(\mathbf{x},\mathbf{y}) \leq 0,\ j=1,\ldots,J_2, \\ & y_j^{(L)} \leq y_j \leq y_j^{(U)},\ j=1,\ldots,n_2. \end{aligned} \tag{2}$$

where, problems P_1 and P_2 deal with variable vectors $\mathbf{x}$ and $\mathbf{y}$ of sizes n_1 and n_2, respectively. The objective and constraint functions for P_i is given as $\mathbf{f}^{(i)}(\mathbf{x},\mathbf{y})$ of size M_i and $\mathbf{g}^{(i)}(\mathbf{x},\mathbf{y}) \leq \mathbf{0}$ of size J_i.

A MoCoEv algorithm consists of two interacting populations, each involving their own decision variables. This can be viewed as two evolutionary algorithms running in parallel where one population is evolved at a time. Each population selfishly optimizes its own objectives with no control over the other population's variables. If the objectives of one population conflict with each other, each population aims to obtain an efficient trade-off solution set with best possible variables which other population's evolution will allow, thereby making the multi-objective co-evolutionary algorithm development a difficult and challenging task.

The populations are evolved one at a time in an iterative manner. P1 is evolved for τ_1 generations with P2 frozen. Then, P2 is evolved for τ_2 generations with P1 frozen. This cycle continues up to T cycles. Each population's evolution process progresses typical to any multi-objective evolutionary algorithm. N_i population members are initialized at random and genetic operators are applied at every generation to improve its own population. Survival operation is performed based on objective trade-off values and constraints. In this method, NSGA-II's procedure was followed for this purpose, but other EMO procedure can be followed as well.

At every generation, each population member needs a member from the other population to evaluate any objective or constraints. To circumvent this issue, every population member is paired with every member of the other population and aggregation functions are used to obtain a representative objective vector for a population member. Once all the pairings are evaluated, several aggregation methods can be applied. One of the proposed aggregation methods is mean aggregation fitness, where the mean function value along each objective ($i = 1, 2$) is assigned as fitness for each population, as described in 3 and 4.

$$F_i^{(1)}(\mathbf{x}^{(k)}) = \frac{1}{N_2}\sum_{l=1}^{N_2} f_i^{(1)}(\mathbf{x}^{(k)},\mathbf{y}^{(l)}), \tag{3}$$

$$F_i^{(2)}(\mathbf{y}^{(l)}) = \frac{1}{N_1}\sum_{k=1}^{N_1} f_i^{(2)}(\mathbf{x}^{(k)},\mathbf{y}^{(l)}). \tag{4}$$

Other aggregation functions, such as the best and the worst objective value can also be used.

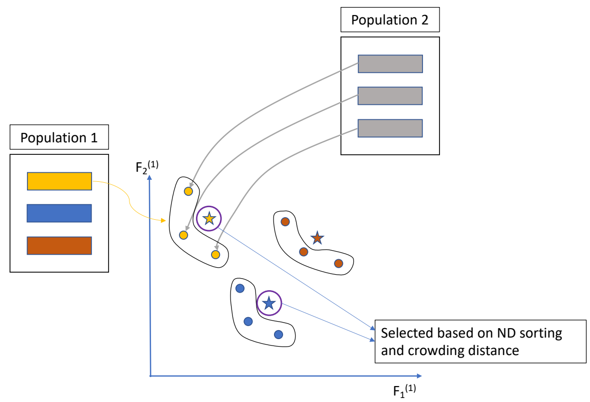


Fig. 1: Co-evolutionary NSGA-II using aggregate fitness.

These aggregate fitness values can be used akin to a conventional MOEA's fitness values. NSGA-II based evaluation procedure is shown in Figure 1. Here, points of each color are members of P1 evaluated against all members of P2. The stars represent the aggregate fitness that is used as a representative vector for non-dominated (ND) sorting and crowding distance based selection. For the minimization problem, the yellow and blue stars form the ND front.

III. Proposed Extensions of MoCoEv

Previous study was restricted to two objectives per population and only competing objectives between the two populations were considered. Moreover, decision-making with aggregation concepts were used. Here, we propose necessary extensions to evaluate the MoCoEv algorithm in a more comprehensive manner.

A. Mixed Cooperative-Competitive Scenarios

In a truly multi-objective problem, objectives are usually in conflict, meaning that the optimal solution for one objective is not usually the optimal solution for any other conflicting objectives. This will cause a trade-off scenario among the objectives, thereby leading the problem to have multiple PO solutions [13], [14].

A careful thinking of our proposed MoCoEv algorithm reveals that no change in the algorithm is required, except to set the objectives accordingly. Since MoCoEv is set to minimize all objectives, the maximization objectives can be multiplied by -1 and set as objectives. The rest of the algorithm should take care of competing and cooperative scenarios and produce resulting PO solutions. Care should be taken to interprete the results.

B. Iterative Decision-making Among Two Populations

In the original study, we proposed two decision-making principles: (i) pseudo-weight approach and (ii) minimum sensitivity approach. In both approaches, a single preferred solution from each population is chosen based on two principles: (i) the solution having the closest calculated pseudo-weight vector from its own PO front to a supplied weight vector, or (ii) the solution having smallest dispersion of PO solutions due to all PO solutions of the second population. While many other such static decision-making principles can be implemented for the MoCoEv results, here we propose a sequential iterative multi-criterion decision-making (MCDM) method, which can be played in a dynamic manner. Similarly, a simultaneous iterative MCDM method can also be applied.

In certain multi-agent system, the choice of a solution by one agent must depend on the solution chosen by the second agent and this needs to be carried on iteratively. Such an iterative decision-making (DM) is applicable for arms-race type of problems or in problems in which each agent gets multiple chances to control the system. We argue that in such situations, while the choice by each player must change based on the solution or strategy chosen by the second agent, the choice can be restricted to only the obtained PO sets ($\mathrm{PO}^{(1)}$ and $\mathrm{PO}^{(2)}$) found by the MoCoEv algorithm. We describe the procedure below.

For a given solution (say, $\mathbf{x}^{(t)} \in \mathrm{PO}^{(1)}$ of the P1 at iteration t), the most preferred solution $\mathbf{y}^{(t)}$ of P2's PO set can be chosen at iteration t using any MCDM technique. Here, we use two such techniques: (i) ℓ_2 weighted-metric method from a supplied aspiration point, and (ii) weighted-sum method for a given weight vector. The objective vectors $\mathbf{f}^{(2)}$ are computed for $(\mathbf{x}^{(t)}, \mathbf{y})$ vectors for which $\mathbf{y} \in \mathrm{PO}^{(2)}$ and the MCDM technique is used to choose the most preferred $\mathbf{y}$. In the next iteration $(t+1)$, this preferred $\mathbf{y}^{(t)}$ is kept fixed (as if, the solution $\mathbf{y}^{(t)}$ is announced by agent 2), and the MCDM technique is repeated to find the most preferred solution $\mathbf{x}^{(t+1)} \in \mathrm{PO}^{(1)}$ by checking each $\mathbf{x}$ solutions and keeping $\mathbf{y}^{(t)}$ fixed. Next, we can keep $\mathbf{x}^{(t+1)}$ fixed and find the most appropriate solution ($\mathbf{y}^{(t+1)}$) for the second population. The process of finding preferred agent-wise solutions iteratively can continue until the process converges to a single stable solution for each population or a fixed number of iterations have elapsed. If the process converges to single solution for each population, the respective solution will be the *stable* solution for the chosen MCDM technique.

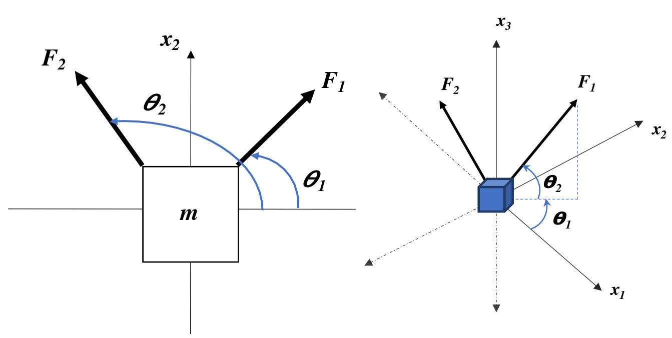
To start the above iterative MCDM method at iteration 1, the knowledge of $\mathbf{y}^{(0)}$ is needed. To simplify matter, we can choose a random solution from P2's PO set, or $\mathbf{y}^{(0)} = \mathrm{random}(\mathrm{PO}^{(2)})$. Alternatively, the best pair for all combinations of $\mathbf{x}$-$\mathbf{y}$ from both PO sets can be identified for the chosen MCDM technique and use the respective $\mathbf{y}$ as $\mathbf{y}^{(0)}$.

C. Three-objective Coevolutionary Algorithm

We extend MoCoEv algorithm to handle three or more ($M \geq 3$) objectives by incorporating NSGA-III's survival procedure [12]. The algorithm is similar to the two objective case except the survival operator. Instead of the crowding distance metric from NSGA-II, we use the reference-direction based survival operator of NSGA-III. Other niche preserving operators (such as that used in MOEA/D [15]) tailored to a coevolutionary scenario can also be incorporated here. The rest of the MoCoEv algorithm can stay the same. This process is expected to generate a well-distributed set of PO solutions on M-objective space for both populations.

IV. RESULTS

We demonstrate the efficacy of our proposed methods on the Tug of War (ToW) problem [4], which is a highly intuitive differential game. This two-player game consists of a mass m placed on a horizontal friction-less plane, as shown in Figure 2a. The players apply forces F_1 and F_2 at their respective chosen angles θ_1 and θ_2, respectively, aiming to minimize or maximize the Cartesian coordinates of the final position of the mass, x_1 and x_2. Acceleration along axes are: $\ddot{x_1} = F_1\cos(\theta_1) + F_2\cos(\theta_2)$ and $\ddot{x_2} = F_1\sin(\theta_1) + F_2\sin(\theta_2)$. Final position of the mass are: $x_1(t) = x_1(0) + \dot{x}_1(0)t + \frac{1}{2}\ddot{x}_1 t^2$ and $x_2(t) = x_2(0) + \dot{x}_2(0)t + \frac{1}{2}\ddot{x}_2 t^2$. The problem is simplified by assuming the forces $F_1 = F_2 = 1N$, mass $m = 1kg$ and initial conditions $x_1(0) = x_2(0) = 0$ and $\dot{x}_1(0) = \dot{x}_2(0) = 0$. Each player only controls one of the angles θ_1 and θ_2. The position after $t_f = \sqrt{2}$ sec is used to construct the objective functions with $x = \theta_1$ and $y = \theta_2$, both in the range $[0, 360]$: $x_1(x, y) = \cos(x) + \cos(y)$ and $x_2(x, y) = \sin(x) + \sin(y)$.



(a) Populations 1 and 2 evolve variables θ_1 and θ_2, respectively.

(b) Each population evolves variable pairs (θ_1 and θ_2) to define the force vector F_i.

Fig. 2: Tug of War problem in 2D and 3D

Thus, according to our MoCoEv problem formulation, the minimization objectives for both populations are given as follows:

$$\begin{aligned} &\text{P1}: f_1^{(1)}(x,y) = x_1(x,y), \quad f_2^{(1)}(x,y) = x_2(x,y),\\ &\text{P2}: f_1^{(2)}(x,y) = -x_1(x,y), f_2^{(2)}(x,y) = -x_2(x,y). \end{aligned} \tag{5}$$

A. Cooperative and Competitive Scenarios

With two objectives for each population for the TOW problem, there can be three scenarios possible. We describe them and the obtained results in the following subsections.

1) Competitive-Competitive scenario: Here, while P1 minimizes both objectives (m_x and m_y), P2 maximizes them. This competitive scenario described in Equation 5 was solved in the previous study [10] and are reproduced here for completeness. In the simulations, we used the population size is N_1 = 50 for P_1 and N_2 = 50 for P_2, SBX crossover operator [16] with $\eta_c = 20$ and $p_c = 0.5$, and polynomial mutation with $\eta_m = 50$, $p_m = 0.5$ for both populations. The total number of generations for each population is $T = 50$, and frequency of each iteration $\tau_1 = \tau_2 = 1$. The mean aggregation fitness function is used for both populations. The gray points in Figures 3a and 3b show all final coordinates (or, objective vectors) of the mass for all combinations of $\mathbf{x}$ and $\mathbf{y}$ for two obtained PO sets $Z^{(1)} = \text{PO}^{(1)}$ and $Z^{(2)} = \text{PO}^{(2)}$.

The solid circles in Figure 3a are shown for a fixed $\mathbf{x}$ and all $\mathbf{y}$ from $Z^{(2)}$. Using the pseudo-weight decision-making approach for a target weight vector of $(0.5, 0.5)$, we obtain a specific $\mathbf{x} = \mathbf{z}^{(1)} = 225.71$ deg., which is used to mark the solid circles. This indicates that when $\mathbf{x}$ is fixed, the variation in $Z^{(2)}$ makes the distribution of $f_1^{(1)}$-$f_2^{(1)}$ as a concave front for P1. For P2, we show the variation of $\mathbf{x}$ on a fixed $\mathbf{y} = 46.05$ deg. (obtained by pseudo-weight approach) in Figure 3b in solid circles. It is clear that the distribution of $f_1^{(2)}$-$f_2^{(2)}$ appears as a concave front for P2 as well.

We also presented in the original study an aggregate representative PO front ($Z^{(1)}$ and $Z^{(2)}$) for each population by computing an average fitness given in Equations 3 and 4. A pseudo-weight approach is then used to select one specific pair ($\mathbf{x}$ and $\mathbf{y}$) with an equal weight vector (0.5, 0.5). The pseudo-weight solutions are marked in a star and the corresponding PO fronts for both populations are shown in blue and red circles, respectively, in Figure 3c. If the two populations are solved independently (not in a co-evolutionary manner), we shall obtain two different PO fronts, as shown in the figure with solid lines. It is clear from the figure that the co-evolutionary solutions are inferior and gets dominated by the respective PO fronts of the individual problems.

2) Cooperative-Cooperative Scenario: In this case, both agents P_1 and P_2 minimizes the same two objectives m_x and m_y. This reduces the overall co-evolutionary problem to a standard two-objective optimization problem, except that two variables are evolved separately. We are expected to achieve the same PO fronts ($Z^{(1)} = Z^{(2)}$) for this case. We ran our modified MoCoEv for the following problem formulation:

$$\begin{aligned} &\text{P1}: f_1^{(1)}(x,y) = x_1(x,y), f_2^{(1)}(x,y) = x_2(x,y),\\ &\text{P2}: f_1^{(2)}(x,y) = x_1(x,y), f_2^{(2)}(x,y) = x_2(x,y). \end{aligned} \tag{6}$$

Figure 4 shows the obtained results. As shown in Figure 4a, we observe that the extent of objective values are identical to the individual optimization case. The aggregate objective values are slightly inferior to the individual fronts due to the averaging of objectives according to Equations 3 and 4. Both individual PO fronts are identical in this case due to the cooperative nature of the two populations. Interestingly, for a fixed $\mathbf{x}$ from P1, the distribution of $f_1^{(1)}$ and $f_2^{(1)}$ for different $\mathbf{y}$ points is now convex, similar in nature to the PO front of P1. Recall that in the competing-competing scenario, we achieved an opposite relationship between Figures 3a and 3b.

The aggregate fronts and the pseudo-weight solutions (points marked with stars) are slightly different for two populations, due to our optimization runs being terminated after a finite number of generations. Ideally, they are expected to be identical.

3) Competitive-Cooperative Scenario: In this scenario, the second agent P_2 is interested in minimizing m_y (the same goal as the first agent P_1) but maximizes m_x (conflicting with the first agent):

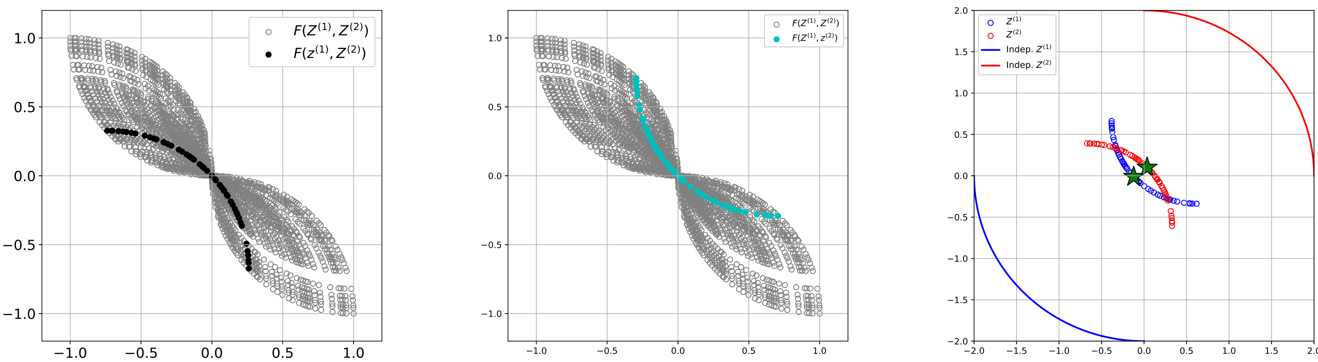
$$\begin{aligned} &\text{P1}: f_1^{(1)}(x,y) = x_1(x,y), \quad f_2^{(1)}(x,y) = x_2(x,y),\\ &\text{P2}: f_1^{(2)}(x,y) = -x_1(x,y), f_2^{(2)}(x,y) = x_2(x,y). \end{aligned} \tag{7}$$

Figure 5 shows the two plots for this scenario. Figure 5a shows all the combinations of $\mathbf{x}$-$\mathbf{y}$ solutions from two PO sets $Z^{(1)}$ and $Z^{(2)}$. For a fixed $\mathbf{x}$ from $Z^{(1)}$, the distribution of $f_1^{(1)}$ and $f_2^{(1)}$ for different $\mathbf{y}$ points is different from the previous two scenarios and is similar in nature to $Z^{(2)}$. A similar observation can also be made from Figure 5a regarding a fixed $\mathbf{y}$ against all $\mathbf{x}$. Figure 5b shows the aggregate PO fronts of both populations and their individual fronts. The respective pseudo-weight solutions for target weight vector of (0.5, 0.5) are marked on this figure.

B. Iterative Decision-making on Tug-of-War Problem

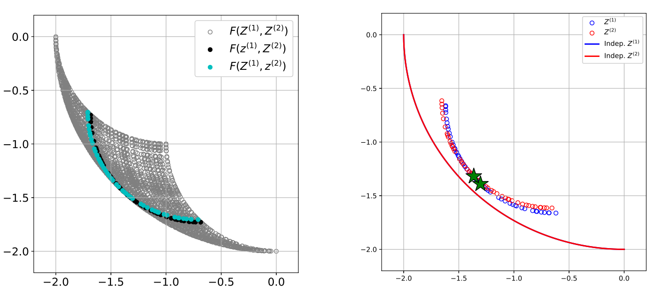
Here, we consider the competing-competing scenario for this study, however, the procedure can be repeated for any other scenarios as well. We assume here that the PO sets $Z^{(1)}$ and $Z^{(2)}$ are known before the decision-making task and we use our proposed iterative DM method to pick solutions from these two sets one by one.

1) Aspiration Points Approach: In this approach, each agent chooses a solution closest to a supplied aspiration point at the start of the decision-making process. For Case 1, we use following aspiration points $(-1,-1)$ and $(1,1)$ for P1 and P2, respectively. At the first iteration ($t = 1$), we start with $\mathbf{y}^{(0)}$ corresponding to the minimum Euclidean distance from P1's given preferred point $(-1,-1)$. For the obtained fronts, $\mathbf{y}^{(0)} = 0$ deg. Keeping this $\mathbf{y}^{(0)}$ fixed, we find the best $\mathbf{x}^{(1)} = 207.4$ deg. that makes the respective objective vector closest to P1's preference point. Figure 6a shows this



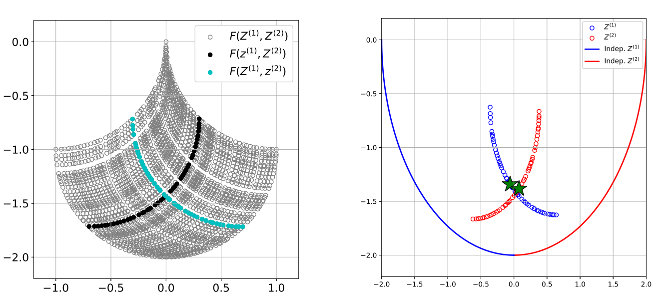
(a) P1. (b) P2. (c) Aggregate fronts for both populations.

Fig. 3: Competing-competing scenario for TOW problem.



(a) P1 and 2. (b) Aggregate fronts for both populations.

Fig. 4: Cooperative-cooperative scenario for TOW problem.



(a) P1 and 2. (b) Aggregate fronts for both populations.

Fig. 5: Cooperative-competitive scenario for TOW problem.

point as "P1 iter=1" with a star. For keeping this $\mathbf{x}^{(1)}$ fixed, the respective variation of $\mathbf{y}$ from $Z^{(2)}$ causes the $f_1^{(2)}$-$f_2^{(2)}$ distribution, as shown in the figure in red triangles. The closest point to P2's aspiration point (1, 1) is the point marked as "P2 iter=1" in the figure. The respective $\mathbf{y}^{(1)} = 38.4$ deg. This completes the first iteration of the proposed decision-making procedure.

In the second iteration, we keep $\mathbf{y}^{(1)} = 38.4$ deg. fixed and find the respective distribution of $f_1^{(1)}$-$f_2^{(1)}$ points with $Z^{(1)}$ points in Figure 6b, shown in blue circles. The closest $\mathbf{x}^{(2)}$ to P1's aspiration point $(-1, -1)$ is shown as "P1 iter=2" in the figure. Similarly, given P1's most preferred point, P2's most preferred is found from the red triangles. The preferred point is marked as "P2 iter=2" in the figure. It is interested to note that these two preferred points are closer together than they were in Iteration 1. When one more iteration is executed, as shown in Figure 6c, the two preferred points marked as "P1 iter=3" and "P2 iter=3" correspond to the same θ_1 and θ_2 values, thereby indicating a convergence has occurred and further iterations will not change these solutions. Table I shows the progress of iterative decision-making for this case, indicating that the iterative process terminates after three iteration.

TABLE I: $\theta_1^{(t)}$ and $\theta_2^{(t)}$ values selected at each iteration of proposed decision-making method. All angles are in degrees.

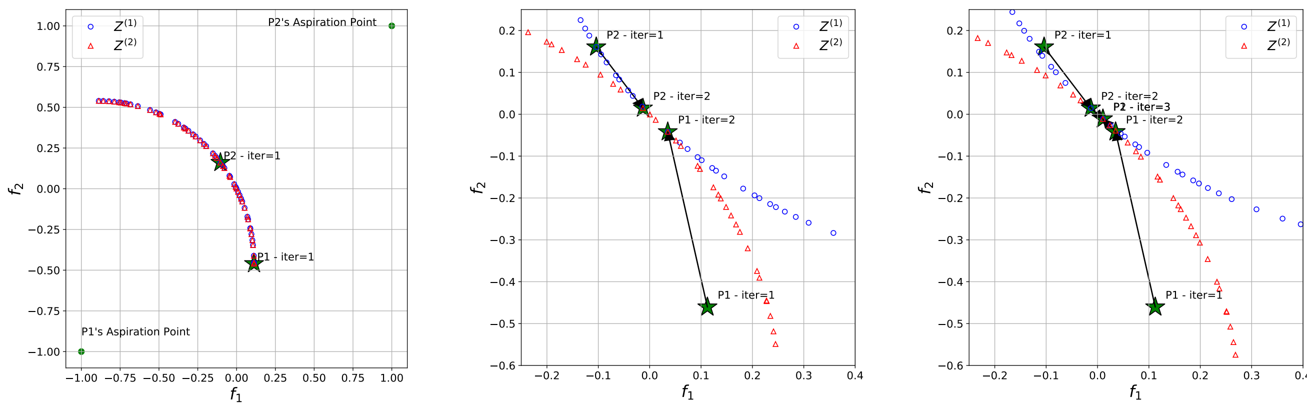
Iter	Aspiration Method							
	Case 1				Case 2			
	P1's Decision		P2's Decision		P1's Decision		P2's Decision	
t	$\theta_1^{(t)}$	$\theta_2^{(t-1)}$	$\theta_1^{(t)}$	$\theta_2^{(t)}$	$\theta_1^{(t)}$	$\theta_2^{(t-1)}$	$\theta_1^{(t)}$	$\theta_2^{(t)}$
1	207.4	0.0	207.4	38.4	244.8	90.0	244.8	66.5
2	221.5	38.4	221.5	42.7	233.0	66.5	233.0	56.7
3	223.5	42.7	223.5	42.7	226.9	56.7	226.9	54.7
4	–	–	–	–	226.3	54.7	226.3	54.7

Iter	Weighted-sum Method							
	Case 1				Case 2			
	P1's Decision		P2's Decision		P1's Decision		P2's Decision	
t	$\theta_1^{(t)}$	$\theta_2^{(t-1)}$	$\theta_1^{(t)}$	$\theta_2^{(t)}$	$\theta_1^{(t)}$	$\theta_2^{(t-1)}$	$\theta_1^{(t)}$	$\theta_2^{(t)}$
1	198.2	90.0	198.2	70.7	236.7	0.0	236.7	12.6
2	198.2	70.7	198.2	70.7	236.7	12.6	236.7	12.6

In Case 2, we repeat the aspiration point method for another pair of aspiration points for two populations: (-0.75, -0.55), and (0.4, 0.8), respectively. This requires four iterations for both populations' decision-making to converge to an identical stable solution. The progress is shown in Table I.

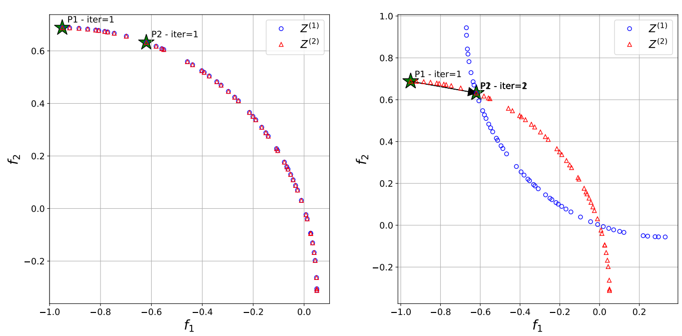
2) Weighted-sum Approach: Next, we use the well-known weight-sum approach in which a solution from a PO set is chosen using the minimum weight-sum of objectives for a given weight vector. In Case 1, we use vectors $(0.75, 0.25)$ and $(0.25, 0.75)$ for P1 and P2, respectively. Figures 7a and 7b show the respective chosen solutions for each population and Table I presents the θ_1 and θ_2 values. The process converges to a set of stable weights in just two iterations.

In Case 2, two different weight vectors $(0.4, 0.6)$ and $(0.8, 0.2)$, for P1 and P2, respectively, are considered. Table



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3.

Fig. 6: Aspiration point based iterative decision-making method – Case 1.



(a) Iteration 1. (b) Iteration 2.

Fig. 7: Weighted-sum based iterative DM method – Case 1.

I show the progress and respective chosen solutions for each population. It is interesting that two iterations are enough for this case as well.

The iterative decision-making procedure proposed here for multi-objective co-evolutionary optimization is generic and can be aided with other MCDM approaches (such as, Tchebyshev or ASF metrics). Whether the problem allows an iterative decision-making process in which each player changes its solution like a two-player game or requires a solution to be chosen upfront for implementation, the above iterative procedure can be played and the final stable solution can be chosen. For the chosen MCDM approach, when the agents reach the stable solutions, they will have no further motivation to change their solutions.

C. Three-objective Co-evolutionary Optimization

Next, we apply our extended MoCoEv algorithm for a three-objective problem by extending the 2-D Tug of War game to have three-dimensional forces. The players aim to minimize or maximize the final Cartesian coordinates of a mass m by applying forces F_1 and F_2, as shown in Figure 2b. The i-th player decides on the angles $\theta_1^{(i)}$ – the angle the force F_i makes from the x-axis while projected onto the x-y plane, and $\theta_2^{(i)}$ – the angle made by the force when measured from the x-y-plane. The first angle is in the range $[0, 360]$ degrees and the second angle lies in $[-90, 90]$ degrees. Similar to the two-dimensional case, the forces are resolved along each axes and final positions are computed. Acceleration and forces are considered constant with $F_1 = F_2 = 1N$ and mass $m = 1$. The problem is further simplified by assuming initial conditions $x_1(0) = x_2(0) = x_3(0) = 0$, $\dot{x}_1(0) = \dot{x}_2(0) = \dot{x}_3(0) = 0$, and total time $t_f = \sqrt{2}$. Hence, objective functions are formulated as follows:

$$x_1(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}) = \cos\theta_2^{(1)} \cos\theta_1^{(1)} + \cos\theta_2^{(2)} \cos\theta_1^{(2)} \quad (8)$$

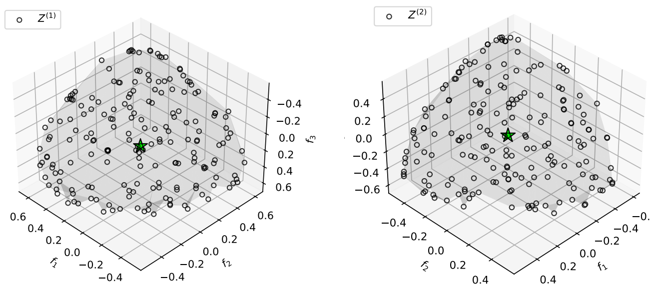
$$x_2(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}) = \cos\theta_2^{(1)} \sin\theta_1^{(1)} + \cos\theta_2^{(2)} \sin\theta_1^{(2)} \quad (9)$$

$$x_3(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}) = \sin\theta_2^{(1)} + \sin\theta_2^{(2)} \quad (10)$$

Ideally, in a pure evolution of combined $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$, the minimization solutions of (x_1, x_2, x_3) should satisfy $\theta_1^{(1)} \in [180, 270]$ deg and $\theta_2^{(1)} \in [0, -90]$ deg. The respective $\boldsymbol{\theta}^{(2)}$ also would be in the same range. The final position of the mass would be uniformly distributed in asall negative octant at a radius of 2 units. Similarly, for a maximization of (m_x, m_y, m_z), the final positions would be evenly distributed in the all positive octant at a radius of 2 units with $\theta_1^{(2)} \in [0, 90]$ and $\theta_2^{(2)} \in [0, 90]$ and $\boldsymbol{\theta}^{(1)}$ will also be in the same range. However, due to co-evolutionary nature of the problem, in a competitive setting, these final positions can never be achieved. Apart from finding the locations for the competitive setting, we also evaluate several combinations of cooperative and competitive environments along with a fully cooperative environment.

In our simulations, we use a population size of $N_1 = N_2 = 192$, SBX crossover operator [16] with $\eta_c = 20$ and $p_c = 0.5$, and polynomial mutation with $\eta_m = 50$, $p_m = 0.5$ and $T = 150$ generations for each population. Frequency of each iteration $\tau_1 = \tau_2 = 1$ is used here. Das and Dennis method is used for generating reference directions with 18 divisions. We use pseudo-weight method to choose a solution from the PO front similar to the 2D case [10]. All simulations are performed with mean fitness aggregation. Similar study can be performed with alternate aggregation functions as well.

1) All Competitive Environment: We first consider a fully competitive environment, where P1 minimizes all three Cartesian coordinates, while P2 maximizes them.



(a) P1: f_i lies in $[-0.4, 0.6]$. (b) P2: f_i lies in $[-0.6, 0.4]$.

Fig. 8: Mean aggregation fitness PO solutions of two populations for all competitive case.

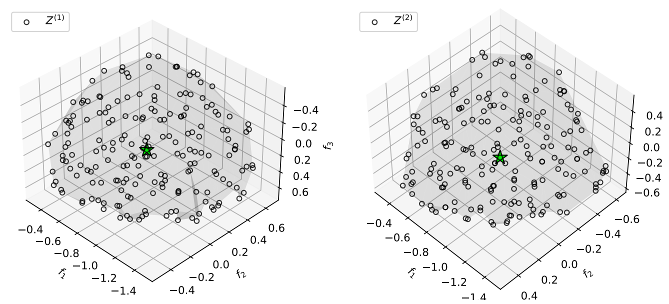
PO front of mean aggregate fitness is shown in Figure 8. We clearly see that the achieved solution is in compromise between the two populations. PO front of each population makes a clear trade-off in each objective. Similar to the 2D case, the two PO fronts (surfaces here instead of curves) overlap each other close to the origin. Both surfaces form a convex front with respect to their objectives. Figure 11a shows the final achieved $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$ values. P1 achieves $\theta_1^{(1)}$ values roughly bounded between $[180, 270]$ and $\theta_2^{(1)}$ values roughly bounded between $[-90, 0]$, whereas P2 achieves $\theta_1^{(2)}$ and $\theta_2^{(2)}$ values bounded roughly between $[0, 90]$ and $[0, 90]$, respectively, as expected.

Decision making based on pseudo-weights with a target wright vector of $([\frac{1}{3}, \frac{1}{3}, \frac{1}{3}])$ yielded $\boldsymbol{\theta}^{(1)} = [225.44, -33.59]$ for P1 and $\boldsymbol{\theta}^{(2)} = [44.74, 37.18]$ for P2. These points are marked with green stars in Figure 8, which can be finally selected as two compromise solutions for both populations. The proposed iterative DM approach can also be used for this problem.

2) Cooperative-Competitive Environments: Next, we consider two cases for combined cooperative-competitive environments. First, we consider a scenario where two objectives (f_2 and f_3) are competitive and objective f_3 is cooperative between the two populations. The PO front of aggregate fitness of this cooperative-competitive-competitive case is shown in Figure 9. The PO front clearly shows that there is no competition between the two population in x_1 direction, leading to identical range in f_1. Both populations achieve minimum values ($-x_1$ direction) while x_2 and x_3 show a compromise. Figure 11b shows the final achieved $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$ values. For P1, the obtained $\theta_1^{(1)}$ values are bounded in the same range as in the previous case (as P1 functions are not different form previous case), but for P2, the obtained $\theta_1^{(2)}$ values are now bounded roughly in $[90, 180]$ deg and $\theta_2^{(2)}$ values are bounded roughly in $[0, 90]$ deg, due to the minimization of $f_1^{(2)}$.

We apply the pseudo-weight approach with a target of $([\frac{1}{3}, \frac{1}{3}, \frac{1}{3}])$ to choose a single solution from the mean aggregation PO front. The process finds the following ND solution: $\boldsymbol{\theta}^{(1)} = [228.15, -34.75]$ deg for P1 and $\boldsymbol{\theta}^{(2)} = [134.14, 34.02]$ deg for P2. These points are marked on the PO fronts with green stars. They seem to lie on the middle of the PO fronts, ensuring equal priority for all three objectives. Importantly, a cooperation of f_1 between two populations increases $\theta_1^{(2)}$ by around 90 deg.

As the second case, we consider a cooperative-cooperative-competitive environment where m_x and m_y are cooperative and m_z is competitive between the two populations. We do not show the PO fronts here for brevity, but the obtained solutions are shown in Figure 11c.

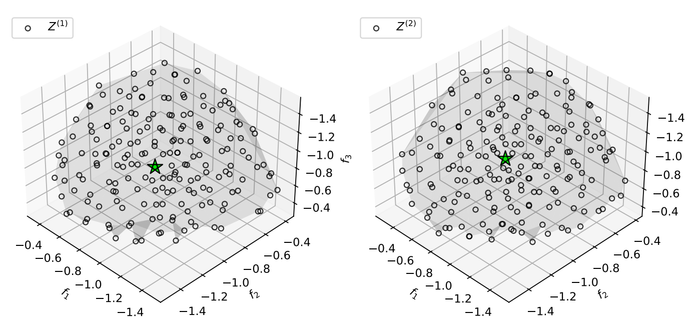


(a) P1: f_1 lies in $[-1.4, -0.4]$. (b) P2: f_1 lies in $[-1.4, -0.4]$.

Fig. 9: Mean aggregation fitness PO solutions of two populations for cooperative-competitive-competitive case.

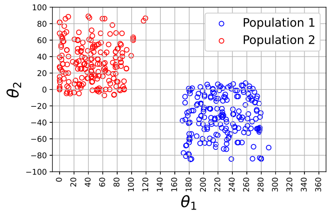
3) All Cooperative Environment: Finally, we evaluate the proposed method on a completely cooperative environment. Here, both populations are aiming to minimize the coordinates of the final position of the mass. Since there is no competition here, the populations need to help each other to achieve their respective optima. The mean aggregate fitness PO front is shown in Figure 10. We can clearly see that both the PO fronts are entirely in $[-x_1, -x_2, -x_3]$ octant. Similar to 2D Tug of War case, the two PO fronts are nearly identical to each other, supporting the argument above. Both populations yield similar solutions: $\theta_1^{(i)}$ values bounded between $[180, 270]$ deg and $\theta_2^{(i)}$ values bounded between $[-90, 0]$ deg, as shown in Figure 11d.

The pseudo-weight solution with equal importance results in a solution for P1 and P2 as $[227.78, -32.46]$ deg and $[229.02, -36.92]$ deg, respectively. They are almost identical, cooperating with each other to trying the take the mass towards the negative octant. These points are marked on Figure 10 with green stars.

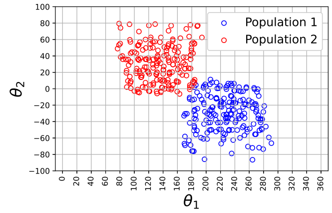


(a) P1: f_i lies in $[-1.4, -0.4]$. (b) P2: f_i lies in $[-1.4, -0.4]$.

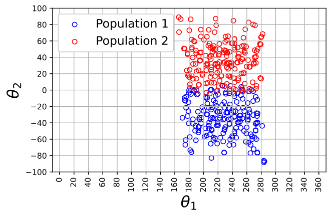
Fig. 10: Mean aggregation fitness PO solutions of two populations for all cooperative case.



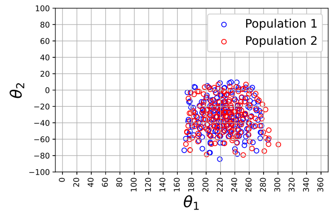
(a) All competitive scenario.



(b) Cooperative-competitive-competitive scenario.



(c) Cooperative-cooperative-competitive scenario.



(d) All cooperative scenario.

Fig. 11: $\boldsymbol{\theta}$ values of Populations 1 and 2 for 3-D TOW problem.

It is also important to note that though this is a two-variable problem, NSGA-III based multi-objective co-evolution does not yield well distributed PO fronts. We believe this is due to the pathology of co-evolution itself. At every evolution step, the opponent has evolved and fitness values have changed from previous generation. Previous fitness values close to the reference line that were chosen may not lie close to the same reference line now. In this ever-changing fitness landscape and finite evolution steps, the points do not converge to their respective reference lines. Further analysis of the convergence of many-objective co-evolutionary methods is left for future studies.

V. Conclusions and Future Studies

In this paper, we have extended a previously proposed multi-objective co-evolutionary (MoCoEv) algorithm in three directions to make the approach more generic and applicable in practice. First, we have proposed to handle mixed competing and cooperative scenarios of objectives between two agents, which may occur in a real-world scenario in which agents may agree with each other on some objectives but differ on others. Second, we have proposed an iterative decision-making procedure in which agents restrict their choice from the obtained PO solutions, but chooses one's preference iteratively by knowing the solution chosen by the other. This simulates the arms race or a game-playing scenario and the goal is to arrive at a stable solution from which no agent is interested in switching to any other solution. Third, we have extended the MoCoEv approach to handle more then two objectives by replacing the niche-preservation operator with a recently-proposed many-objective EA (NSGA-III). Results on different versions of the Tug of War problem show the efficacy of each extension with illustrative figures and tables.

The originally proposed MoCoEv is now ready for application to different multi-agent scenarios (i.e., two or more objectives, mixed competing-cooperative scenarios, and static or iterative decision-making strategies). The study now must be extended to solve more complex multi-objective multi-agent problems, such as generative adversarial network (GAN) design and other game and control system problems.

Acknowledgements

Authors acknowledge a Facebook Research Award for conducting this study.

References

[1] J. Paredis, "Coevolutionary constraint satisfaction," in *Parallel Problem Solving from Nature III (PPSN-III)*, 1994, pp. 46–55.

[2] H. J. C. Barbosa, "A genetic algorithm for min-max problems," in *Proceedings of the First International Conference on Evolutionary Computation and Its Application (EvCA'96)*, 1996, pp. 99–109.

[3] K. O. Stanley and R. Miikkulainen, "Competitive coevolution through evolutionary complexification," *Journal of artificial intelligence research*, vol. 21, pp. 63–100, 2004.

[4] E. Eisenstadt, A. Moshaiov, and G. Avigad, "Co-evolution of strategies for multi-objective games under postponed objective preferences," in *2015 IEEE conference on computational intelligence and games (CIG)*. IEEE, 2015, pp. 461–468.

[5] E. Eisenstadt-Matalon, G. Avigad, A. Moshaiov, and J. Branke, "Rationalizable strategies in multi-objective games under undecided objective preferences," 2016.

[6] E. Eisenstadt-Matalon and A. Moshaiov, "Mutual rationalizability in vector-payoff games," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2019, pp. 593–604.

[7] E. Matalon-Eisenstadt, A. Moshaiov, and G. Avigad, "The competing travelling salespersons problem under multi-criteria," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2016, pp. 463–472.

[8] E. Eisenstadt and A. Moshaiov, "Decision-making in non-cooperative games with conflicting self-objectives," *Journal of Multi-Criteria Decision Analysis*, vol. 25, no. 5-6, pp. 130–141, 2018.

[9] A. Żychowski, A. Gupta, J. Mańdziuk, and Y. S. Ong, "Addressing expensive multi-objective games with postponed preference articulation via memetic co-evolution," *Knowledge-Based Systems*, vol. 154, pp. 17–31, 2018.

[10] A. Suresh, K. Deb, and V. Naresh, "Towards multi-objective co-evolutionary problem solving."

[11] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[12] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints," *IEEE transactions on evolutionary computation*, vol. 18, no. 4, pp. 577–601, 2013.

[13] K. Deb, *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley, 2001.

[14] C. A. C. Coello, D. A. VanVeldhuizen, and G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Boston, MA: Kluwer, 2002.

[15] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

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