# Iterative multi-criterion decision-making methods for multi-objective co-evolution in cooperative and competitive environments

Jaturong Kongmanee
CSE848 Term Project Report
Michigan State University, East Lansing, USA
kongmane@msu.edu

## ABSTRACT

In this project, we propose three iterative multi-criterion decision-making (MCDM) approaches for multi-objective co-evolutionary optimization. The proposed approaches are used to demonstrate how each agent iteratively selects the most preferred strategy/solution from its obtained Pareto-optimal (PO) solution set. The PO solution set is obtained from the previous study, which is the result of multi-objective co-evolutionary algorithms on the Tug of War problem. These proposed iterative MCDM approaches are generic and can be used with other static MCDM methods. To complement and highlight the significance of the proposed approaches, two competitive-competitive and cooperative-cooperative scenarios and four MCDM methods are employed. Two of three proposed approaches are able to demonstrate that a final stable solution for both agents can be choosen.

## CCS CONCEPTS

• **Theory of computation → Design and analysis of algorithms**; *Algorithm design techniques.*

## KEYWORDS

Co-evolutionary algorithm, multi-objective optimization, multi-criterion decision making, cooperative and competitive co-evolution

**ACM Reference Format:**

Jaturong Kongmanee. 2021. Iterative multi-criterion decision-making methods for multi-objective co-evolution in cooperative and competitive environments. In *Proceedings of the Genetic and Evolutionary Computation Conference 2021 (GECCO '21).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

In many practical, real-world problems, multiple conflicting objectives optimization problems are required to be optimized simultaneously. Although, there are different approaches to solve a multi-objective optimization problem. One of the most popular methods, due to its characteristics, is to employ Multi-objective Evolutionary Algorithms (MOEAs). MOEAs are population-based approaches. A population of solutions is evolved to create a new population in each iteration leading to a set of multiple non-dominated (trade-off) solutions, known as Pareto-optimal (PO) solutions, in a single simulation run. This property provides a wide variety of trade-off solutions to the users. Another equally important task to make MOEA procedure complete is to incorporate a multi-criterion decision-making (MCDM) procedure to choose one preferred solution from the obtained PO solutions [3]. A number of studies in MCDM that are based on MOEAs have been discussed [7]. Coello et al. [2] categorize the preference-based MOEAs for assisting MCDM as follows: (i) aspiration point; (ii) weight; (iii) trade-off rate; (iv) utility function; (v) outranking; and (vi) fuzzy logic. Each approach has its advantages and disadvantages. In multi-objective co-evolutionary (MOCoEv) problems, two or more agents/populations are evolved simultaneously. The fitness of a solution in one agent is evaluated based on comparisons with other agents that are also evolving [5, 6]. These comparisons are based on the interactions among these agents, i.e., competitive or cooperative interactions. Due to the nature of the population-based approach, a posterior MCDM is still required to make multi-objective co-evolutionary algorithm complete. Despite a number of static MCDM studies, there has been a limited study of MCDM in a dynamic manner, simulating the arms-race conditions of two evolving agent in selecting an appropriate solutions from its PO solution set in various scenarios.

Thus, this project aims to explore to better understand the insights and issues related to multi-objective co-evolutionary from the perspective of decision-making. In this study, we restrict our study to a two-agent system only with the mixed competing and cooperating scenarios. The obtained PO solutions from MOCoEv algorithm proposed in the previous study [8] were used to simulate choosing a strategy/action of one agent depending on the solution selected by the second agent and the process is carried iteratively in the arms-race condition of two evolving agents. The iterative multi-criterion decision-making methods will be implemented to demonstrate how each agent iteratively selects an appropriate solution from its PO solution set for different scenarios. In all three methods, the process of finding a preferred solution in each agent's iteration can iteratively continue until the identical stable solution is chosen or the predefined number of iterations has elapsed. Three iterative multi-criterion decision-making methods will be carried out here.


*GECCO '21, July 10–14, 2021, Lille, France*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

(1) *a sequential iterative multi-criterion decision-making* method. In this method, each agent chooses the best preferred solution from its obtained PO set one by one iteratively, considering the opponent's selected solution.
(2) *a sequential iterative multi-criterion decision-making worst-case selection* method. As inspired by the work done by Eisenstadt et al. [4]. Instead of choosing the best strategy from its obtained PO set, each agent choose the worst strategy from its obtained PO set. In this method, both agent behave irrationally and has no motivation to win or control the system by always selecting a worst-case strategy.
(3) *a simultaneous iterative multi-criterion decision-making* method. Each agent chooses a preferred solution from its PO set simultaneously knowing only a selected strategy from the previous iteration of the opponent.

Note that the other static decision-making methods from the literature mentioned earlier can also be applied. The results of the proposed iterative decision-making method methods will be visualized to show the interaction between two agents in choosing a preferred strategy in each iteration and the progress of selected strategy. The author believes that this study will lead to a better understanding of multi-objective co-evolutionary optimization in various scenarios.

In the remainder of the report, the two objective multi-objective co-evolutionary algorithms proposed in the previous study and are introduced in the background section. Three proposed sequential iterative decision-making methods in competitive and cooperative scenarios are elaborated in the methodology section. The results of three proposed iterative decision-making methods are presented and the interaction between two agents in choosing a preferred strategy in each iteration and the progress of selected strategy are visualized. Finally, the conclusions, findings, and future studies are highlighted in the conclusion section.

# 2 BACKGROUND

## 2.1 Multi-objective Co-evolutionary Problem

Conceptually, in multi-objective co-evolutionary (MOCoEv) problems, two or more agents/populations are evolved alternately. Each agent has its decision variables and a set of objectives and constraints. The fitness of a solution in one agent is evaluated based on comparisons with other agents that are also evolving [5, 6]. These comparisons are based on the interactions among these agents, i.e., competitive or cooperative interactions.

In the previous study [8], the multi-objective co-evolutionary problem was formulated as:

Problem $P_1$:

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

Problem $P_2$:

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

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

From the overall idea of the multi-objective co-evolutionary algorithm, one population is evolved iteratively for a specified iteration at a time, while another population is frozen. For example, P1 is evolved for $\tau_1$ generations while P2 is frozen. Then, P2 is evolved for $\tau_2$ generations with P1 frozen. The process continues up to $T$ cycles. As briefly described earlier, at every generation, each population member needs members from the other population to evaluate its objective and constraint functions. To do this, every population member is paired with every member of the other population. Then the aggregate function is applied in order to obtain a representative objective vector for a population member. According to the previous study, the mean aggregation is used to obtain the mean function value along with each objective (as a fitness value for each population). The proposed mean aggregation function from the previous study is described in 3 and 4.

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

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

## 2.2 Tug of War (ToW) Problem

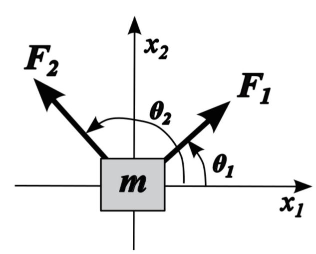



**Figure 1: The game setting of the ToW problem**

This two-player game setting consists of a mass $m$ placed on a horizontal friction-less plane. The players minimize or maximize the Cartesian coordinates of the mass's position along the $x_1$ and $x_2$ axis by applying forces $F_1$ and $F_2$ at the chosen angle $\theta_1$ and $\theta_2$ respectively. The visualization of the game setting is shown in Figure 1. Acceleration along axes are defined as: $\ddot{x_1} = F_1\cos(\theta_1) + F_2\cos(\theta_2)$ and $\ddot{x_2} = F_1\sin(\theta_1) + F_2\sin(\theta_2)$. The final positions of the mass are: $x_1$ and $x_2$. To simplify the problem, the forces $F1 = F2 = 1N$, mass $m = 1kg$, and initial conditions $x_1(0) = x_22(0) = 0$ and $\dot{x}_1(0) = \dot{x}_2(0) = 0$. Each player has the control only on one angle $\theta_1$ or $\theta_2$. The objective functions are defined as: $x_1(\theta_1, \theta_2) = \cos(\theta_1) + \cos(\theta_2)$ and $x_2(\theta_1, \theta_2) = \sin(\theta_1) + \sin(\theta_2)$, where $x = \theta_1$ and $y = \theta_2$ (both in the range [0, 360]).

#### 2.2.1 *Competitive-Competitive scenario.*

In this competitive scenario, P1 minimizes two objectives $x_1$ and $x_2$, while P2 maximizes these two objectives. Thus, the minimization objectives for agent P1 and maximization objective for agent P2 are given as follows:

$$\begin{aligned} &\text{P1}: f_1^{(1)}(\theta_1, \theta_2) = x_1(\theta_1, \theta_2), \quad f_2^{(1)}(\theta_1, \theta_2) = x_2(\theta_1, \theta_2), \\ &\text{P2}: f_1^{(2)}(\theta_1, \theta_2) = -x_1(\theta_1, \theta_2), f_2^{(2)}(\theta_1, \theta_2) = -x_2(\theta_1, \theta_2). \end{aligned} \tag{5}$$

#### 2.2.2 *Cooperative-Cooperative scenario.*

In the case of cooperative-cooperative, both agents P1 and P2 minimize the same two objectives $x_1$ and $x_2$. Thus, he optimization problems are formulated as as follows:

$$\begin{aligned} &\text{P1}: f_1^{(1)}(\theta_1, \theta_2) = x_1(\theta_1, \theta_2), f_2^{(1)}(\theta_1, \theta_2) = x_2(\theta_1, \theta_2), \\ &\text{P2}: f_1^{(2)}(\theta_1, \theta_2) = x_1(\theta_1, \theta_2), f_2^{(2)}(\theta_1, \theta_2) = x_2(\theta_1, \theta_2). \end{aligned} \tag{6}$$

## 3 METHODOLOGY

To better understand and simulate in an arm-race scenario of a multi-agent system, in which one agent's selected strategy must change based on the strategy selected by another agent. The obtained PO set of two agents of a ToW problem, from the previous study's MOCoEv algorithm of Suresh et al. [8], is used here to simulate how each agent iteratively selects the preferred solution from its obtained Pareto-optimal solution set in an arm-race scenario. Four MCDM techniques used here: (i) $l_2$ weighted-metric method from a supplied aspiration point, (ii) weighted-sum method for a given weight vector, (iii) achievement scalarization function (ASF) [9] method for a given weight vector, and (iv) Tchebysheff method [9] for a given weight vector. Note that the ASF and Tchebysheff methods are applied with the default setting of pymoo framework [1]. The three proposed iterative MCDM methods are described below.

### 3.1 Sequential Iterative Decision-Making

For a given solution $\theta_1^{(t)} \in \text{PO}^{(1)}$ of P1 at iteration $t$, the most preferred solution $\theta_2^{(t)}$ from P2's PO set can be selected at iteration $t$ using one of the four MCDM techniques implemented above. The most preferred solution $\theta_2^{(t)}$ is selected by computing the objective vectors $\mathbf{f}^{(2)}(\theta_1^{(t)}, \theta_2)$ vectors for which $\theta_2 \in \text{PO}^{(2)}$ using selected one MCDM technique out of the four mentioned. In the next iteration $(t + 1)$, this preferred $\theta_2^{(t)}$ is kept fixed, and the selected MCDM technique is repeated to find the most preferred solution $\theta_1^{(t+1)} \in \text{PO}^{(1)}$ by checking each $\theta_1$ solution while keeping $\theta_2^{(t)}$ fixed. Next, we keep $\theta_1^{(t+1)}$ fixed and find the most preferred solution, $\theta_2^{(t+1)}$, for the second agent P2. The process of finding preferred agent-wise solutions continues iteratively until the predefined number of iterations has elapsed or converged to a single stable for each agent. The stable solution for the chosen MCDM technique is found if the process converges to a single solution for each agent.

To begin the iterative MCDM process, the best pair for all combinations $\theta_1 - \theta_2$ from both PO sets is selected using the selected MCDM technique. For example, the pair $(\theta_1, \theta_2)$ found, $\theta_2$ is used as the respective $\theta_2^{(0)}$ and $\theta_1$ is used as $\theta_1^{(1)}$. This results in the same front for both population P1 and P2 in the iteration 1 (as will be shown in the results section).

### 3.2 Sequential Iterative Decision-Making Worst-Case Selection

Similar to the previous method in beginning the iterative process, all combinations from both PO sets are considered, and the worst pair $(\theta_1, \theta_2)$ is considered instead of the best pair by using the selected MCDM technique. Then the iterative MCDM process can begin, where $\theta_2$ is used as the respective $\theta_2^{(0)}$ and $\theta_1$ is used as $\theta_1^{(1)}$. As expected, this results in the same front for both population P1 and P2 in iteration 1.

To continue the iterative MCDM process, for an obtained $\theta_1^{(1)} \in \text{PO}^{(1)}$ of P1 at iteration $t = 1$, the worst preferred solution $\theta_2^{(t=1)}$ from P2's PO set can be selected by computing the objective vectors $\mathbf{f}^{(2)}(\theta_1^{(t=1)}, \theta_2)$ vectors for which $\theta_2 \in \text{PO}^{(2)}$ using the selected MCDM technique for iteration $t = 1$. In the next iteration $(t = 2)$, this preferred $\theta_2^{(1)}$ is kept fixed, and the selected MCDM technique is repeated to find the worst preferred solution $\theta_1^{(t=2)} \in \text{PO}^{(1)}$ by checking each $\theta_1$ solution while keeping $\theta_2^{(t=1)}$ fixed. Next, we keep $\theta_1^{(t=2)}$ fixed and find the worst preferred solution, $\theta_2^{(t=2)}$, for the second agent P2. The process of finding a stable solution for the chosen MCDM technique continues iteratively until a stable solution for each agent is found or the number of iteration reaches the maximum predefined number.

### 3.3 Simultaneous Iterative Decision-Making

In this method, each population P1 and P2 must select the most preferred solution from its own PO set by using the selected solution from the previous iteration of its opponent. Similar to the two previous methods, the selected MCDM technique is used to select the most preferred solution. To start the process, a solution $\theta_2^{(0)} \in \text{PO}^{(2)}$ of P2 is selected randomly for P1, and a solution $\theta_1^{(0)} \in \text{PO}^{(1)}$ of P1 is also selected at random for P2. Note that this results in a different front for P1 and P2 in the first iteration, which is different from the first two proposed methods. For a given solution $\theta_1^{(0)} \in \text{PO}^{(1)}$ of P1 at iteration $t = 0$, the most preferred solution $\theta_2^{(t=1)}$ for iteration $t = 1$ is selected by computing the objective vectors $\mathbf{f}^{(2)}(\theta_1^{(0)}, \theta_2)$ vectors for which $\theta_2 \in \text{PO}^{(2)}$ using the selected MCDM technique. Simultaneously, for a given solution $\theta_2^{(0)} \in \text{PO}^{(2)}$ of P2 at iteration $t = 0$, the most preferred solution $\theta_1^{(t=1)}$ is selected by computing the objective vectors $\mathbf{f}^{(1)}(\theta_1, \theta_2^{(0)})$ vectors for which $\theta_1 \in \text{PO}^{(1)}$ using the same selected MCDM technique. Similar to the termination condition of the first two proposed methods, this process of finding a single solution for each agent continues simultaneously for a fixed number of predefined iteration or converges to a single stable for each agent.

# 4 RESULTS

In this section, we consider competitive-competitive and cooperative-cooperative scenarios. We assume that PO sets $Z^{(1)}$ of population P1, and $Z^{(2)}$ of population P2 are obtained before the three proposed iterative MCDM methods are employed. The proposed iterative MCDM methods are applied to select solutions from these two sets. The aim of this section is to demonstrate how each agent iteratively selects the most preferred solution out of its PO set, given the solution provided by its opponent. All four MCDM techniques are used for all three proposed iterative MCDM methods.

## 4.1 Competitive-Competitive Scenario

### 4.1.1 *Sequential Iterative Decision-Making.*

(1) **Aspiration Points Technique** In the aspiration points approach, each agent selects a solution closet to its supplied aspiration point at the beginning of the decision-making process. For case 1, the following aspiration points $(-1, -1)$ and $(1, 1)$ are selected for P1 and P2 respectively. At the first iteration ($t = 1$), we start with $\theta_2^{(0)} = 0$ deg. obtained from the minimum euclidean distance from P1's given aspiration point. Keeping this $\theta_2^{(0)}$ fixed, we find $\theta_1^{(1)} = 207.4$ deg. closet to P1's aspiration point shown in Fig 2a as "P1 iter=1" with a star. For keeping this $\theta_1^{(1)}$ fixed, the respective variation of $\theta_2$ from $Z^{(2)}$ causes the $f_1^{(2)}$-$f_2^{(2)}$ distribution shown as the red triangles in the Figure 2. The respective $\theta_2^{(1)} = 38.4$ deg. is the closet point to P2's aspiration point $(1, 1)$, marked as "P2 iter=1". This first iteration of the process is completed and the process continues.
Interestingly, the two preferred points are closer together when the number of iterations increase. In this way, as shown in Figure 2c, the two preferred points marked as "P1 iter=3" and "P2 iter=3" correspond to the same $\theta_1$ and $\theta_2$ values, indicating a convergence and further iterations will not change these solutions. In Case 2, another pair of aspiration point for two agents: (-0.75, -0.55), and (0.4, 0.8) are selected respectively. The progress of iterative decision-making for both cases are shown in Table 1.

(2) **Weighted-sum Technique** In this approach, each agent selects a solution from a PO set using the minimum weight-sum of objectives for the given weight vector. In case 1, the vectors (0.75, 0.25) and (0.25, 0.75) are selected for P1 and P2 respectively. The selected solutions for each agent are shown in Figure 3a and 3b and Table 1. In Case 2, another pair of weight vectors for two agents: (0.4, 0.6), and (0.8, 0.2) are selected respectively. Surprisingly, for two cases, only two iterations are required to converge.

(3) **Achievement Scalarization Function (ASF) Technique** In this approach, we utilize the pymoo framework by supplying the weight vector to obtain the minimum value of the ASF metric. In case 1, the vectors (0.75, 0.25) and (0.25, 0.75) are selected for P1 and P2 respectively. The selected solutions for each agent are shown in Table 1. In Case 2, another pair of weight vectors for two agents: (0.4, 0.6), and (0.8, 0.2) are selected respectively. For two cases, only one iteration is required to converge.

(4) **Tchebysheff Technique.** Similarly to the ASF technique, we also leverage the pymoo framework. In case 1, the vectors (0.75, 0.25) and (0.25, 0.75) are selected for P1 and P2 respectively. The selected solutions based on Tchebysheff for each agent are shown in Table 1. In Case 2, another pair of weight vectors for two agents: (0.4, 0.6), and (0.8, 0.2) are selected respectively. Same as the ASF technique, only one iteration is required to converge for two cases.

**Table 1: $\theta_1^{(t)}$ and $\theta_2^{(t)}$ values selected at each iteration of the sequential iterative decision-making method for competitive-competitive scenario. 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 |

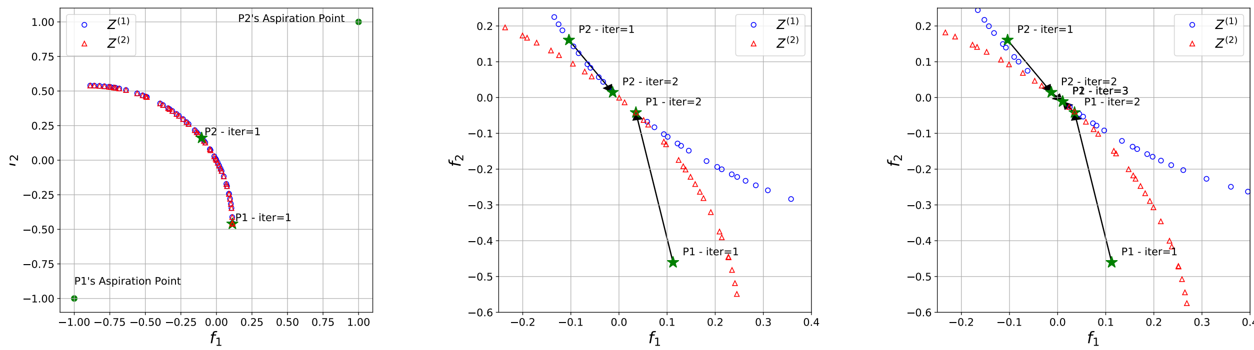
| Iter | Weighted-sum Method | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Case 1 | | | | Case 2 | | | |
| | P1's Decision | | P2's Decision | | P1's Decision | | P2's Decision | |
| $t$ | $\theta_1^{(t)}$ | $\theta_2^{(t-1)}$ | $\theta_1^{(t)}$ | $\theta_2^{(t)}$ | $\theta_1^{(t)}$ | $\theta_2^{(t-1)}$ | $\theta_1^{(t)}$ | $\theta_2^{(t)}$ |
| 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 |

| Iter | Achievement Scalarization Function (ASF) 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 | 189.3 | 8.3 | 189.3 | 8.3 | 189.3 | 8.3 | 189.3 | 8.3 |

| Iter | Tchebysheff 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 | 270.0 | 90.0 | 270.0 | 90.0 | 270.0 | 90.0 | 270.0 | 90.0 |

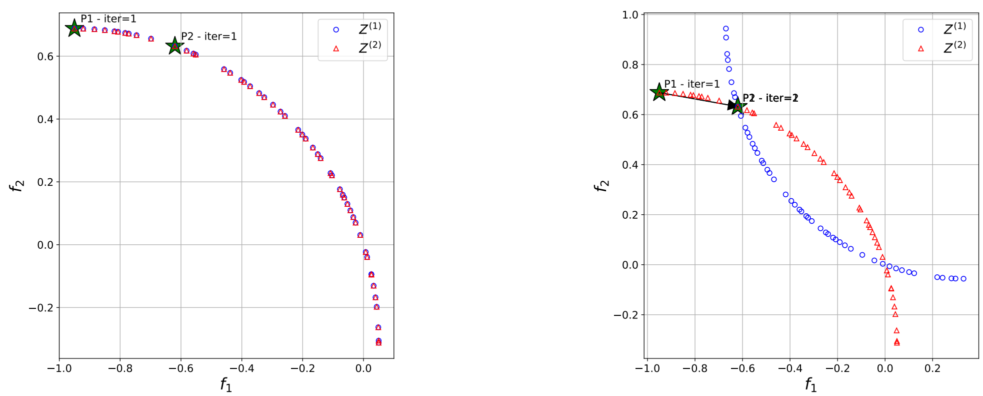
### 4.1.2 *Sequential Iterative Decision-Making Worst-Case Selection.*

For the aspiration point-based MCDM technique, in case 1, the following aspiration points (1,1) and (1,1) are selected for P1 and P2 respectively. Starting from the first iteration, we observe that the label "P1 iter=1" marked as a star is the farthest away from the P1's aspiration point. Similar to P2, the label "P2 iter=1" marked as a star is the farthest away from the P2's aspiration point as well. However, it is interesting that within two iterations they converge to the same $\theta_1$ and $\theta_2$ as shown in Figures 4a and 4b for case 1 of the aspiration point (see case 2 in Figure 8). In the case of the weighted-sum technique, the weight vectors (0.75, 0.25) and (0.25, 0.75) are selected for P1 and P2 respectively for case 1. In case 2,



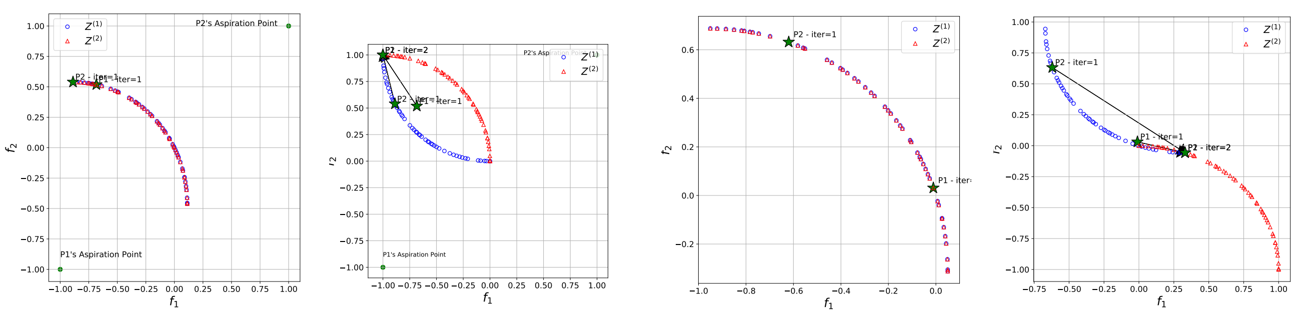

(a) Iteration 1. (b) Iteration 2. (c) Iteration 3.

**Figure 2: Aspiration point based used in the sequential iterative DM method for competitive-competitive scenario – Case 1.**




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

**Figure 3: Weighted-sum based used in the sequential iterative DM method for competitive-competitive scenario – Case 1.**




(a) Iteration 1. – Case 1 (b) Iteration 2. – Case 1 (c) Iteration 1. – Case 1 (d) Iteration 2. – Case 1

**Figure 4: Results of sequential iterative decision-making worst-case selection method for competitive-competitive scenario, (a) and (b) are from aspiration point approach, (c) and (d) are from weighted-sum approach**

another pair of weight vectors for two agents: (0.4, 0.6), and (0.8, 0.2) are selected respectively. Instead of using a minimum weighted sum of objectives for the given weight vector, each agent selects a solution from a PO set using the maximum weighted sum of objectives. Surprisingly, two agents take only two iterations to converge to a stable solution, in which both agents have no motivation to

**Table 2: $\theta_1^{(t)}$ and $\theta_2^{(t)}$ values selected at each iteration of the sequential iterative decision-making worst-case selection method for competitive-competitive scenario. All angles are in degrees.**

| Iter | Aspiration Method | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Case 1 | | | | Case 2 | | | |
| | P1's Decision | | P2's Decision | | P1's Decision | | P2's Decision | |
| $t$ | $\theta_1^{(t)}$ | $\theta_2^{(t-1)}$ | $\theta_1^{(t-1)}$ | $\theta_2^{(t)}$ | $\theta_1^{(t)}$ | $\theta_2^{(t-1)}$ | $\theta_1^{(t)}$ | $\theta_2^{(t)}$ |
| 1 | 207.4 | 0.0 | 207.4 | 90.0 | 244.7 | 90.0 | 244.7 | 0.0 |
| 2 | 180.0 | 90.0 | 180.0 | 90.0 | 270.0 | 0.0 | 270.0 | 0.0 |

| 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 | 198.1 | 90.0 | 198.1 | 70.7 | 236.6 | 0.0 | 236.6 | 12.6 |
| 2 | 270.0 | 70.7 | 270.01 | 70.7 | 180.0 | 12.6 | 180.0 | 12.6 |

| Iter | Achievement Scalarization Function (ASF) 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 | 270.0 | 8.3 | 270.0 | 0.0 | 270.0 | 8.3 | 270.0 | 0.0 |
| 2 | 270.0 | 0.0 | 270.0 | 0.0 | 270.0 | 0.0 | 270.0 | 0.0 |

| Iter | Tchebysheff 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 | 180.0 | 79.7 | 180.0 | 90.0 | 180.0 | 79.7 | 180.0 | 90.0 |
| 2 | 180.0 | 90.0 | 180.0 | 90.0 | 180.0 | 90.0 | 180.0 | 90.0 |

further change their solution. The process of selecting a solution for each agent one by one is shown in Figures 4c and 4d for case 1 of the weighted-sum (see case 2 in Figure 8). Similar to the aspiration point and weighted-sum techniques, ASF and Tchebysheff Technique applied only need two iterations to arrive at a stable solution for the competitive-competitive scenario (see plots in Figure 9 of ASF and Tchebysheff). Table 2 shows the progress of all $\theta$ values selected at each iteration.

#### 4.1.3 *Simultaneous iterative Decision-Making*.

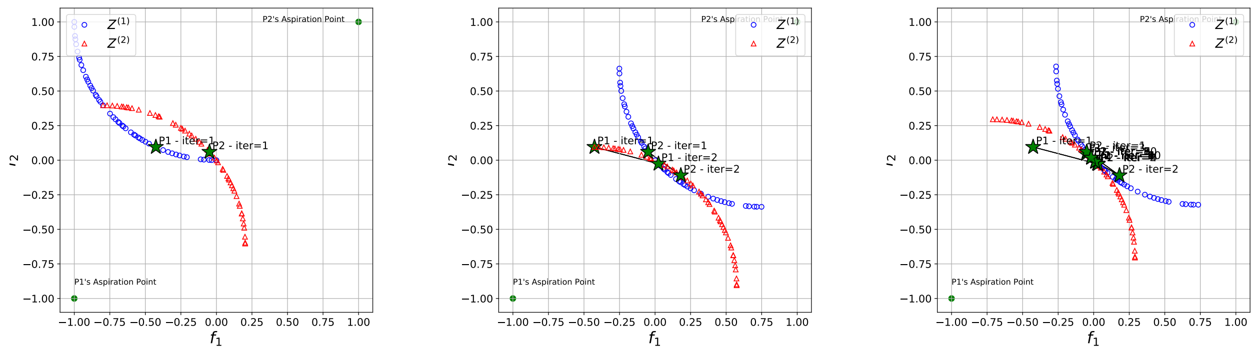
In this method, we only consider the competing-competing scenario and aspiration point MCDM technique for this study. However, the procedure can be repeated for other scenarios and static MCDM techniques as well. Note that we restrict the strategies for each agent only from the PO sets obtained from the previous study and must be known before the decision-making process.

**Table 3: $\theta_1^{(t)}$ and $\theta_2^{(t)}$ values selected at each iteration of the simultaneous iterative decision-making method for competitive-competitive scenario. All angles are in degrees.**

| Iter | Aspiration Method | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Case 1 | | | | Case 2 | | | |
| | P1's Decision | | P2's Decision | | P1's Decision | | P2's Decision | |
| $t$ | $\theta_1^{(t)}$ | $\theta_2^{(t-1)}$ | $\theta_1^{(t-1)}$ | $\theta_2^{(t)}$ | $\theta_1^{(t)}$ | $\theta_2^{(t-1)}$ | $\theta_1^{(t-1)}$ | $\theta_2^{(t)}$ |
| 1 | 221.5 | 35.9 | 185.8 | 29.2 | 211.8 | 27.5 | 185.8 | 32.6 |
| 2 | 217.0 | 29.2 | 221.5 | 42.6 | 214.3 | 32.6 | 211.8 | 46.2 |
| 3 | 223.5 | 42.6 | 217.0 | 41.5 | 221.5 | 46.2 | 214.3 | 47.3 |
| 4 | 223.5 | 41.5 | 223.5 | 42.6 | 221.5 | 47.3 | 221.5 | 51.4 |
| 5 | 223.5 | 42.6 | 223.5 | 42.6 | 223.5 | 51.4 | 221.4 | 51.4 |
| 6 | – | – | – | – | 223.5 | 51.4 | 223.5 | 52.6 |
| 7 | – | – | – | – | 224.7 | 52.6 | 223.5 | 52.6 |
| 8 | – | – | – | – | 224.7 | 52.6 | 224.7 | 52.6 |

(1) **Success in Convergent Case**
In this aspiration point technique, each agent must choose the solution that is closest to its predefined aspiration point at the beginning of the decision-making process. In case 1, the following aspiration points(1,1) and (1,1) are selected for P1 and P2 respectively. At the first iteration, the random $\theta_2^{(0)}$ = 35.9 degrees is given to P1, and the random $\theta_1^{(0)}$ = 185.8 degrees is given to P2. The closest point to each of the agent's aspiration points is marked as a star. As the process continues iteratively, we observe that these two preferred points are closer together than they were in the previous iteration. However, at iteration 4 and 5 of case 1, P1 selected the same most preferred solution $\theta_1$ for different given $\theta_2$ from P2. This phenomenon demonstrates how the choice by each agent must change based on the solution selected by the second agent in the arm-race problem. For case 2, another pair of aspiration points for two agents: (-0.75, -0.55), and (0.4, 0.8) are selected respectively. However, a similar phenomenon also occurs for the P2 at iteration 7 and 8, where P2 also changes its most preferred solution, given different $\theta_1$ from P1. The progress of iterative decision-making for both cases is shown in Table 3.

(2) **Failure in Convergent Case**
In the case of failure in convergence, the following aspiration points(1,1) and (1,1) are selected for P1 and P2 respectively for case 1. In case 2, another pair of aspiration points for two agents: (-0.75, -0.55),and (0.4, 0.8) are selected respectively. After running experiments with many random $\theta_1^{(0)}$ and $\theta_2^{(0)}$, we observe that if the first iteration conforms to this shape of the $f1 - f2$ distribution, two agents do not agree to arrive at the final stable solutions. Some pieces of evidence are picked to be shown in Figure 5 and the rest are shown in Figure 16 in the appendix section. However, it intuitively is hard to figure out what could be the reason behind it. Further analysis is required to understand this phenomenon.




**(a) Iteration 1. – Case 1** **(b) Iteration 2. – Case 1** **(c) Iteration 10. – Case 1**

**Figure 5: Failed to converge - aspiration point based used in simultaneous iterative DM method for competitive-competitive scenario – Case 1.**

**Table 4: $\theta_1^{(t)}$ and $\theta_2^{(t)}$ values selected at each iteration of the sequential iterative decision-making method for cooperative-cooperative scenario. 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 | 179.9 | 269.9 | 179.9 | 269.9 | 179.9 | 269.9 | 179.9 | 269.9 |

| 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 | 198.1 | 98.4 | 198.2 | 252.3 | 235.0 | 236.2 | 235.0 | 193.7 |
| 2 | 198.1 | 252.3 | 198.1 | 252.3 | 235.0 | 193.7 | 235.0 | 193.7 |

| Iter | Achievement Scalarization Function (ASF) 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 | 179.9 | 246.2 | 179.9 | 269.9 | 226.8 | 246.2 | 226.8 | 180.0 |
| 2 | 179.9 | 269.9 | 179.9 | 269.9 | 270.0 | 180.0 | 270.0 | 180.0 |

| Iter | Tchebysheff 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 | 247.6 | 257.1 | 247.6 | 180.0 | 179.9 | 257.1 | 179.9 | 269.9 |
| 2 | 270.0 | 180.0 | 270.0 | 180.0 | 179.9 | 269.9 | 179.9 | 269.9 |

**Table 5: $\theta_1^{(t)}$ and $\theta_2^{(t)}$ values selected at each iteration of the sequential iterative decision-making worst-case selection method for cooperative-cooperative scenario. All angles are in degrees.**

| Iter | Aspiration Method | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Case 1 | | | | Case 2 | | | |
| | P1's Decision | | P2's Decision | | P1's Decision | | P2's Decision | |
| $t$ | $\theta_1^{(t)}$ | $\theta_2^{(t-1)}$ | $\theta_1^{(t-1)}$ | $\theta_2^{(t)}$ | $\theta_1^{(t)}$ | $\theta_2^{(t-1)}$ | $\theta_1^{(t)}$ | $\theta_2^{(t)}$ |
| 1 | 270.0 | 180.0 | 270.0 | 269.9 | 270.0 | 180.0 | 270.0 | 269.9 |
| 2 | 270.0 | 269.9 | 270.0 | 269.9 | 270.0 | 269.9 | 270.0 | 269.9 |

| 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 | 198.1 | 198.4 | 198.1 | 180.0 | 235.0 | 236.6 | 235.0 | 269.9 |
| 2 | 270.0 | 180.0 | 270.01 | 180.0 | 179.9 | 269.9 | 179.9 | 269.9 |

| Iter | Achievement Scalarization Function (ASF) 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 | 270.0 | 246.2 | 270.0 | 269.9 | 270.0 | 246.2 | 270.0 | 269.9 |
| 2 | 270.0 | 269. | 270.0 | 269.9 | 270.0 | 269. | 270.0 | 269.9 |

| Iter | Tchebysheff 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 | 179.9 | 257.1 | 179.9 | 269.9 | 270.0 | 257.1 | 270.0 | 180.0 |
| 2 | 179.9 | 269.9 | 179.9 | 269.9 | 179.9 | 180.0 | 179.9 | 180.0 |

## 4.2 Cooperative-Cooperative Scenario

### *4.2.1 Sequential Iterative Decision-Making.*

For the aspiration point MCDM technique, the aspiration points for P1 and P2, and for both cases are (-1, -1) - (1, 1) and (-0.75, -0.55) - (0.4, 0.8) as used in the competitive-competitive scenarios. All three weighted-based methods: weighted-sum, ASF, and Tchebysheff are also have the same weight vectors as used in the competitive-competitive scenarios. In case 1, the vectors (0.75, 0.25) and (0.25, 0.75) are selected for P1 and P2 respectively. In Case 2, another pair of weight vectors for two agents: (0.4, 0.6), and (0.8, 0.2) are selected respectively. We observe that the process requires only two iterations maximum to converge to a stable solution for four MCDM techniques. Table 4 shows the progress of values selected at each iteration

#### 4.2.2 *Sequential Iterative Decision-Making worst-case selection*.

In this method for the cooperative scenario, all aspiration points and the weight vectors for both cases are the same as used in the Sequential Iterative Decision-Making for the cooperative scenario. It is interesting for this method that only two iterations are enough for both agents to converge to a set of stable solutions. Even though both agents behave irrationally by selecting the worst solutions one by one. The progress and respective chosen solutions for each agent are shown in Table 5.

The three iterative decision-making procedures for multi-objective co-evolutionary optimization proposed here are generic and can be used with other static multi-criterion decision-making exits in the literature. These three proposed iterative multi-criterion decision-making procedures can be used to simulate the two-player games, where each player selects a solution one by one. Only the simultaneous sequential iterative decision-making method can't find the final stable solution for both in some cases, while the other first two sequential iterative decision-making are able to find the stable solution for both players in both competitive-competitive and cooperative-cooperative scenarios.

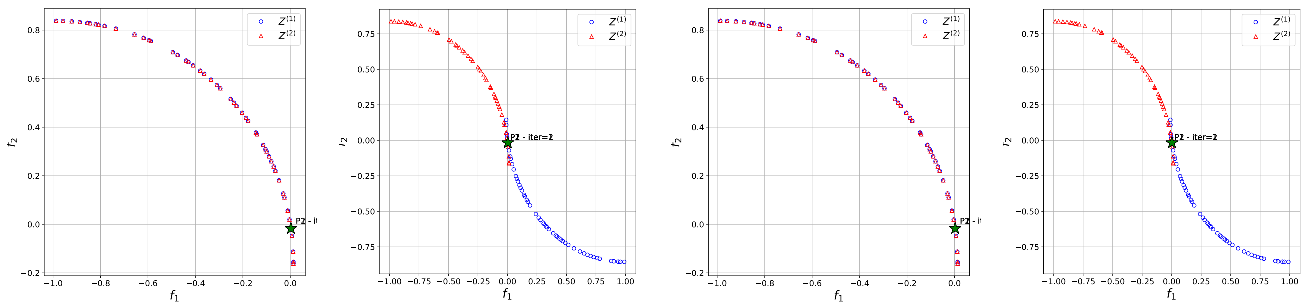
## 5 CONCLUSION

Three iterative decision-making procedures are proposed which are the following: (i) sequential iterative decision-making, (ii) sequential iterative decision-making worst-case selection, and (iii) simultaneous iterative decision-making. Each agent only restricts its strategies from the obtained PO solutions and selects a preference strategy iteratively based on only a strategy selected by its opponent. These MCDM procedures were used to simulates the game-playing scenario in competitive and cooperative environments between two agents. The results of the two proposed methods show that each agent can arrive at a stable solution where each agent is not interested in changing their strategy; however, this may not apply to complex problems. Further analysis is required to better understand the underlying mechanism. The limitation of the proposed methods is that they can't apply to the dynamic settings, where the set of solutions/strategies has to be re-optimized after each agent take an action.

## REFERENCES

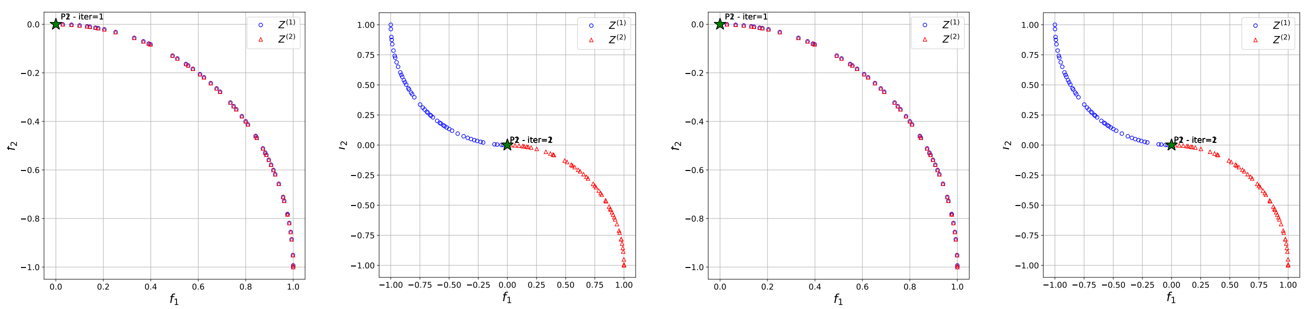
[1] J. Blank and K. Deb. 2020. Pymoo: Multi-Objective Optimization in Python. *IEEE Access* 8 (2020), 89497–89509.
[2] Carlos A Coello Coello, Gary B Lamont, David A Van Veldhuizen, et al. 2007. *Evolutionary algorithms for solving multi-objective problems*. Vol. 5. Springer.
[3] Kalyanmoy Deb. 2011. Multi-objective optimisation using evolutionary algorithms: an introduction. In *Multi-objective evolutionary optimisation for product design and manufacturing*. Springer, 3–34.
[4] Erella Eisenstadt and Amiram Moshaiov. 2018. Decision-making in non-cooperative games with conflicting self-objectives. *Journal of Multi-Criteria Decision Analysis* 25, 5-6 (2018), 130–141.
[5] Chi-Keong Goh and Kay Chen Tan. 2008. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 13, 1 (2008), 103–127.
[6] Chi Keong Goh, Kay Chen Tan, DS Liu, and Swee Chiang Chiam. 2010. A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design. *European Journal of Operational Research* 202, 1 (2010), 42–54.
[7] Kaisa Miettinen. 2012. *Nonlinear multiobjective optimization*. Vol. 12. Springer Science & Business Media.
[8] Anirudh Suresh, Kalyanmoy Deb, and Vishnu Naresh. [n.d.]. Towards Multi-objective Co-evolutionary Problem Solving. ([n. d.]).
[9] Andrzej P Wierzbicki. 1980. The use of reference objectives in multiobjective optimization. In *Multiple criteria decision making theory and application*. Springer, 468–486.

# A APPENDIX



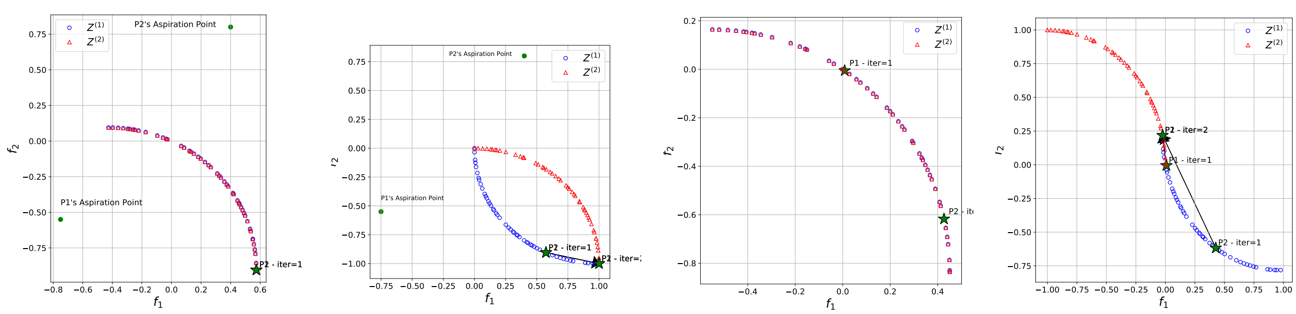

(a) Iteration 1. – Case 1 (b) Iteration 2. – Case 1 (c) Iteration 1. – Case 2 (d) Iteration 2. – Case 2

**Figure 6: ASF based used in sequential iterative decision-making method for competitive-competitive scenario**



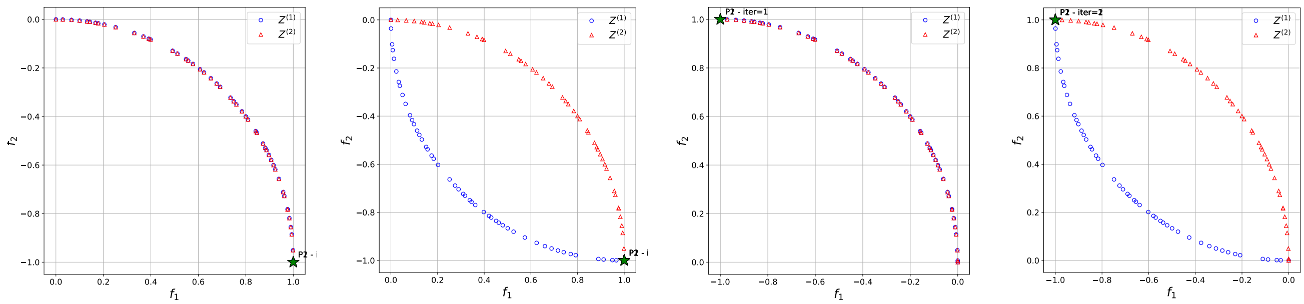

(a) Iteration 1. – Case 1 (b) Iteration 2. – Case 1 (c) Iteration 1. – Case 2 (d) Iteration 2. – Case 2

**Figure 7: Tchebysheff based used in sequential iterative decision-making method for competitive-competitive scenario**



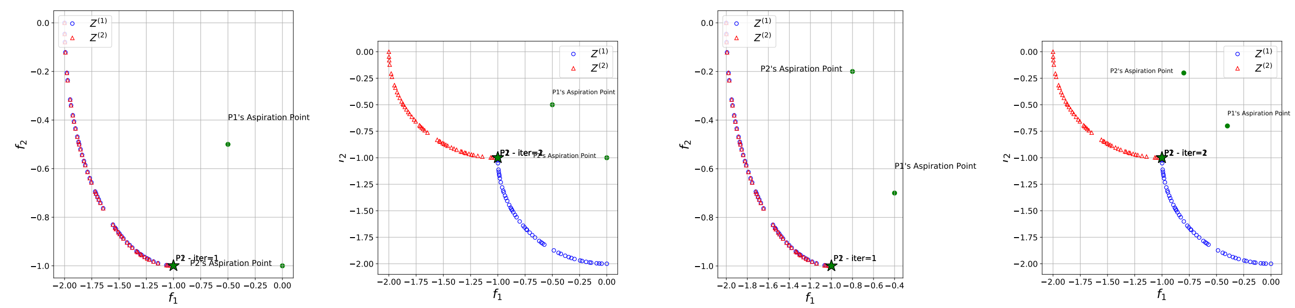

(a) Iteration 1. – Case 2 (b) Iteration 2. – Case 2 (c) Iteration 1. – Case 2 (d) Iteration 2. – Case 2

**Figure 8: Aspiration points based used in sequential iterative decision-making worst-case selection method for competitive-competitive scenario – Case 2**



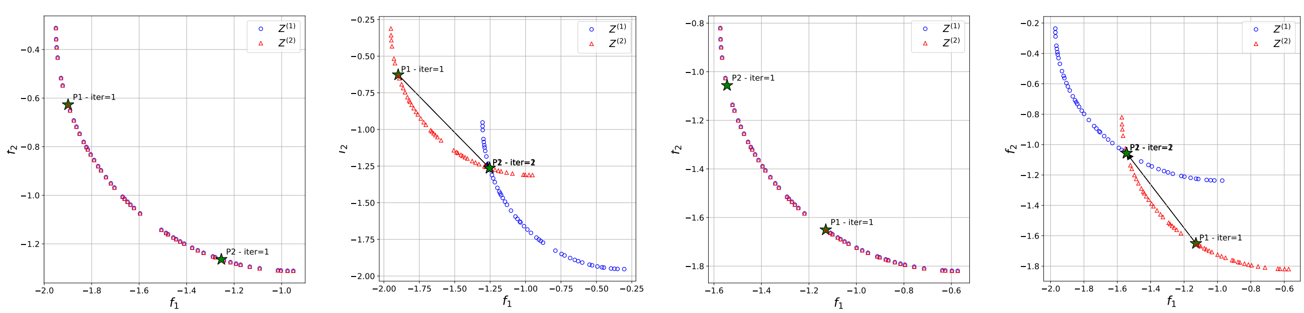

**(a) Iteration 1. – Case 1** **(b) Iteration 2. – Case 1** **(c) Iteration 1. – Case 1** **(d) Iteration 2. – Case 1**

**Figure 9: ASF and Tchebysheff based sequential iterative decision-making worst-case selection method for competitive-competitive scenario**



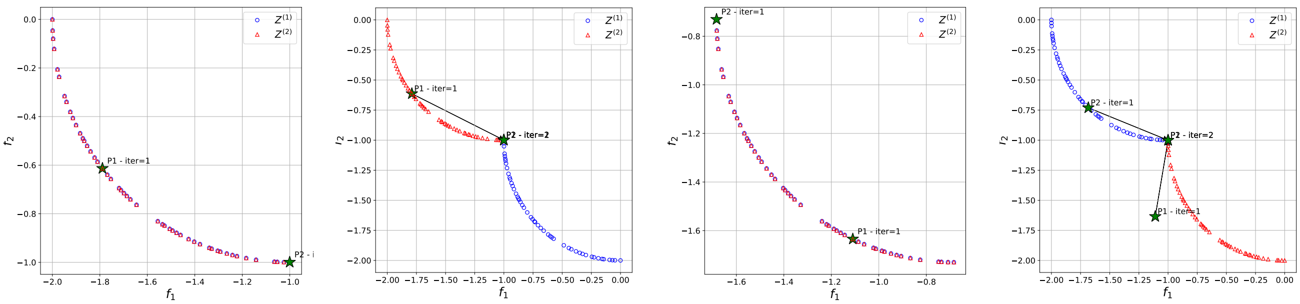

**(a) Iteration 1. – Case 1** **(b) Iteration 2. – Case 1** **(c) Iteration 1. – Case 2** **(d) Iteration 2. – Case 2**

**Figure 10: Aspiration point based used in sequential iterative decision-making method for cooperative-cooperative scenario**



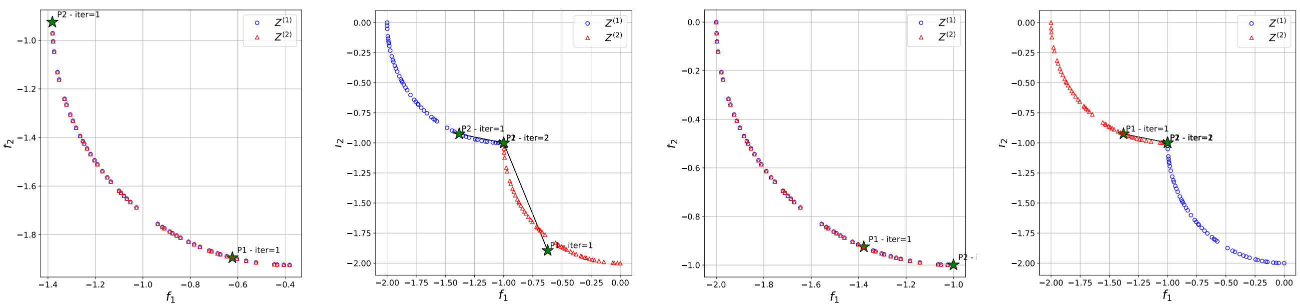

**(a) Iteration 1. – Case 1** **(b) Iteration 2. – Case 1** **(c) Iteration 1. – Case 2** **(d) Iteration 2. – Case 2**

**Figure 11: Weighted-sum based used in sequential iterative decision-making method for cooperative-cooperative scenario**



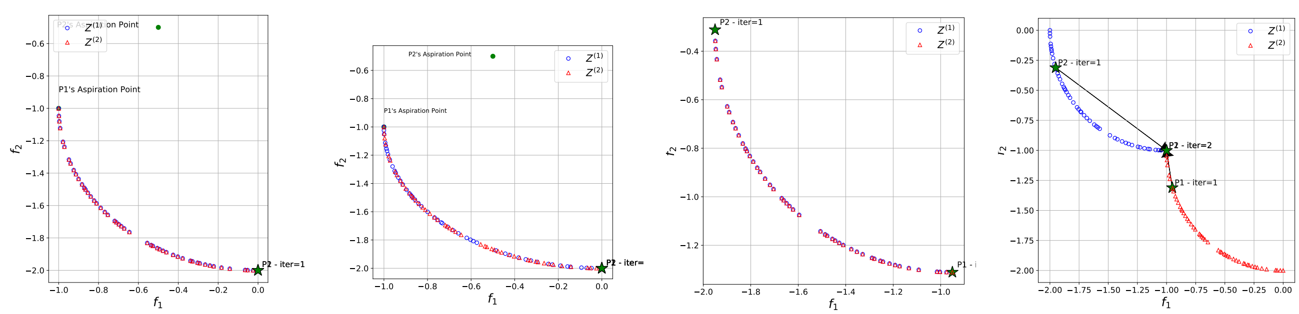

**(a) Iteration 1. – Case 1** **(b) Iteration 2. – Case 1** **(c) Iteration 1. – Case 2** **(d) Iteration 2. – Case 2**

**Figure 12: ASF based used in sequential iterative decision-making method for cooperative-cooperative scenario**



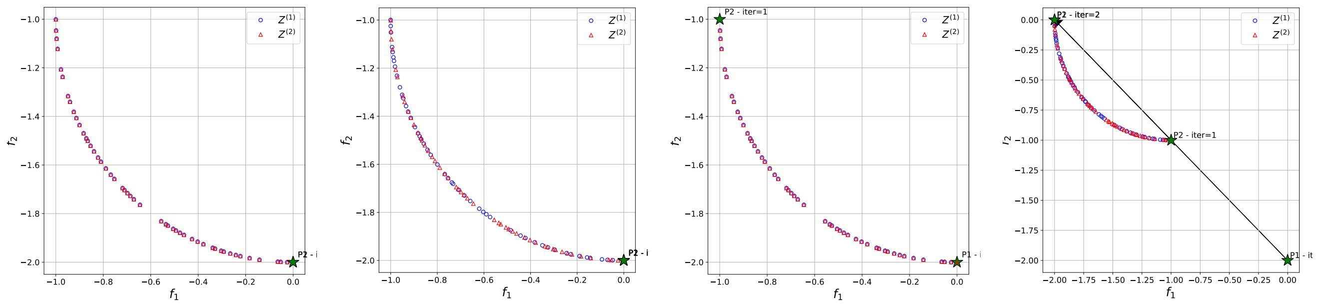

**(a) Iteration 1. – Case 1** **(b) Iteration 2. – Case 1** **(c) Iteration 1. – Case 2** **(d) Iteration 2. – Case 2**

**Figure 13: Tchebysheff based used in sequential iterative decision-making method for cooperative-cooperative scenario**



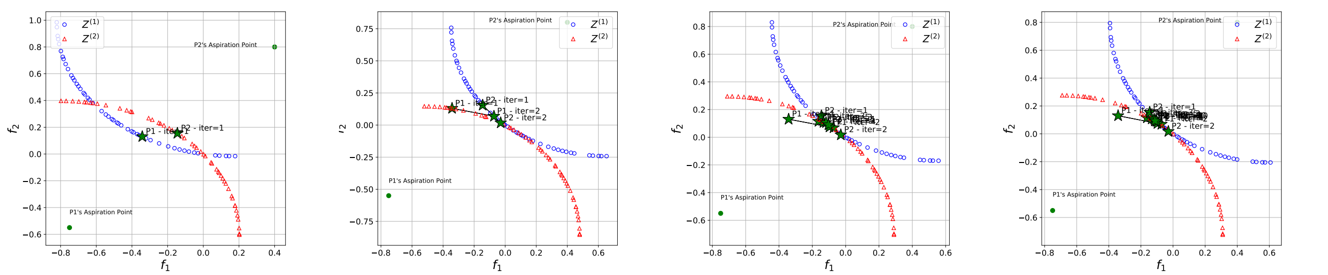

**(a) Iteration 1. – Case 2** **(b) Iteration 2. – Case 2** **(c) Iteration 1. – Case 1** **(d) Iteration 2. – Case 1**

**Figure 14: Aspiration and Weighted-sum based used in sequential iterative decision-making worst-case selection method for cooperative-cooperative scenario**




(a) Iteration 1. – Case 1 (b) Iteration 2. – Case 1 (c) Iteration 1. – Case 2 (d) Iteration 2. – Case 2

**Figure 15: ASF and Tchebysheff based used in sequential iterative decision-making worst-case selection method for cooperative-cooperative scenario**




(a) Iteration 1. – Case 2 (b) Iteration 2. – Case 2 (c) Iteration 5. – Case 2 (d) Iteration 10. – Case 2

**Figure 16: Failed to converge - aspiration point based used in simultaneous iterative DM method for competitive-competitive scenario – Case 2.**