Computer Modeling in
Engineering & Sciences

Tech Science Press

Doi:10.32604/cmes.2025.062332

ARTICLE

MOCBOA: Multi-Objective Chef-Based Optimization Algorithm Using Hybrid Dominance Relations for Solving Engineering Design Problems

Nour Elhouda Chalabi[1], **Abdelouahab Attia**[2], **Abdulaziz S. Almazyad**[3], **Ali Wagdy Mohamed**[4,5], **Frank Werner**[6], **Pradeep Jangir**[7] **and Mohammad Shokouhifar**[8,9,*]

[1]Computer Science Department, University of M'sila, Ichebilia, M'sila, 28000, Algeria
[2]Computer Science Department, Mohamed El Bachir El Ibrahimi University of Bordj BouArreridj, Bordj BouArreridj, 34000, Algeria
[3]Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, 11543, Saudi Arabia
[4]Operations Research Department, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza, 12613, Egypt
[5]Applied Science Research Center, Applied Science Private University, Amman, 11931, Jordan
[6]Faculty of Mathematics, Otto-von-Guericke University, Magdeburg, 39016, Germany
[7]University Centre for Research and Development, Chandigarh University, Mohali, 140413, India
[8]Department of Electrical and Computer Engineering, Shahid Beheshti University, Tehran, 14399, Iran
[9]DTU AI and Data Science Hub (DAIDASH), Duy Tan University, Da Nang, 550000, Vietnam

*Corresponding Author: Mohammad Shokouhifar. Email: mohammadshokouhifar@duytan.edu.vn

Received: 16 December 2024; Accepted: 07 March 2025; Published: 11 April 2025

ABSTRACT: Multi-objective optimization is critical for problem-solving in engineering, economics, and AI. This study introduces the Multi-Objective Chef-Based Optimization Algorithm (MOCBOA), an upgraded version of the Chef-Based Optimization Algorithm (CBOA) that addresses distinct objectives. Our approach is unique in systematically examining four dominance relations—Pareto, Epsilon, Cone-epsilon, and Strengthened dominance—to evaluate their influence on sustaining solution variety and driving convergence toward the Pareto front. Our comparison investigation, which was conducted on fifty test problems from the CEC 2021 benchmark and applied to areas such as chemical engineering, mechanical design, and power systems, reveals that the dominance approach used has a considerable impact on the key optimization measures such as the hypervolume metric. This paper provides a solid foundation for determining the most effective dominance approach and significant insights for both theoretical research and practical applications in multi-objective optimization.

KEYWORDS: Multi-objective optimization; chef-based optimization algorithm (CBOA); pareto dominance; epsilon dominance; cone-epsilon dominance; strengthened dominance

1 Introduction

An essential concept in several disciplines, such as engineering, economics, and artificial intelligence, is multi-objective optimization. These domains frequently feature situations where it's necessary to simultaneously optimize several competing goals. For instance, it is typical in engineering design to have to balance performance with cost and weight reduction. It is difficult to develop solutions to such challenges that meet all of the objectives because of the inherent trade-offs. Even though several works have been presented to solve them [1,2], sophisticated multi-objective optimization approaches are required [3,4] due to their complexity in traversing these trade-offs and finding a set of balanced solutions.

Dominance relations, which offer a framework for assessing and contrasting alternatives, are essential to this process (Multi-objective optimization). These relations establish a solution's relative superiority or inferiority based on how well it performs in comparison to other solutions across the given objectives. Therefore, the choice of a dominance relation strongly impacts the diversity, convergence, and efficiency of the optimization process. The most prevalent dominance relations include Pareto dominance, Epsilon dominance, Cone-epsilon dominance, and Strengthened dominance relations, each providing distinct benefits for various optimization problems. Given the widespread use of those relations as a foundational concept, several algorithms have been created to take advantage of them. Due to the extensive number of algorithms only the state-of-the-art ones are briefly reviewed here.

Probably the most well-known and frequently applied dominance notion is Pareto dominance. If a solution is better in at least one objective and not worse in any, it is said to Pareto-dominate another. Numerous multi-objective algorithms, such as the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [5], are based on this idea. NSGA-II is well known for its ability to quickly sort non-dominated solutions using a crowding distance mechanism and a variety of Pareto-optimal solutions. In Strength Pareto Evolutionary Algorithm II (SPEA-II) [6], each solution is given a strength value by SPEA2, which indicates how many solutions it dominates. To preserve diversity, it also takes into consideration the density of solutions in the objective space. Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [7] is also a known algorithm. In this algorithm, a multi-objective optimization issue is broken down by MOEA/D into many scalar optimization subproblems, which are then all optimized at the same time. The optimum answers to these subproblems are found using Pareto dominance. Multi-Objective Particle Swarm Optimization (MOPSO) [8] expands upon the standard single-objective PSO algorithm to enable it to handle multiple objectives. It updates the particle positions using Pareto dominance and a global repository of non-dominated solutions. Other techniques include Multi-Objective Grey Wolf Optimization (MOGWO) [9], Multi-Objective Cat Swarm Optimization (MOCSO) [10], and Multi-objective Cohort Intelligence (MOCI) [11].

Another dominance relation is Epsilon Dominance, a variation of Pareto Dominance. It introduces a tolerance level, ε (epsilon), to account for small variations within solutions in each objective. Some of the recent algorithms that use Epsilon dominance include Guided Multi-objective equilibrium optimizer (GMOEO) [12] which enhances exploration and diversity by updating solutions through Epsilon dominance, resulting in efficient movement towards the Pareto front. Another method named Guided Multi-objective Marine Predator Algorithm (GMOMPA) [13], is a modified MPA-based framework for addressing multi-objective optimization issues. It includes an external archive Epsilon dominance, a fast non-dominated solution, and crowding distance. to obtain non-dominated solutions, ensuring fast convergence towards the Pareto optimal. Multi-Objective Gaining–Sharing Knowledge optimization (MOGSK) [14] introduces the extended version of the gaining-sharing knowledge optimization (GSK), to tackle real-world optimization challenges. The algorithm utilizes an external archive population, fast nondominated sorting, and the Epsilon dominance relation to accomplish diversity, coverage, and convergence. The study in [15] investigates a co-evolutionary approach to determine the precise value of ε in multi-objective evolutionary algorithms (MOEAs). The authors in [16] present a new decomposition-based multi-objective evolutionary algorithm for complex Pareto fronts that uses a hybrid weighting technique and adaptive Epsilon dominance to achieve optimal diversity. By employing a population archive, crowding distance, and Epsilon dominance to direct search regions, the work in [17] expands the single objective Manta-Ray foraging optimization to multi-objective cases named (MOMRFO).

Cone-epsilon dominance relation is a variant of Epsilon dominance designed for optimization problems with three or more objectives, or many-objective optimization. It combines Epsilon dominance with a conical region of dominance. Different works discuss that Cone-epsilon dominance is a good method for evolutionary algorithms such as [18] and [19], where it increases the diversity of solutions and enhances the performance in evolutionary multi-objective optimization. Another work that has employed this method is [12], which presents a comparison with Epsilon dominance relation.

Strengthened dominance relation also known as (SDR) is an advanced approach generally used in MOEAs, especially when tackling optimization problems with many different objectives (more than three). It is intended to enhance the selection process by striking a balance between two crucial factors: variety and convergence. Many works have employed SDR or its other variant in multi/many objectives optimization such as Multi-Objective Harris Hawks Optimization (MOHHO) [20], which solves multi-objective optimization problems using a Strengthened dominance relation. In [21], the performance of SDR is examined, and a new dominance relation (CSDR) based on SDR is proposed. In NSGA-II, the CSDR takes the place of Pareto dominance. In [22], a new Reference Points-based Strengthened dominance relation (RPS-dominance) has been introduced into NSGA-II. To differentiate and further stratify Pareto-equipment solutions, it presents a reference point set and the convergence metric. The Guided Differential Evolution method (MGDE) [23] is another approach to solving many objective optimization problems. It applies a Strengthened dominance relation and bi-goal evolution, as well as modified differential evolutionary operators for crossover and mutation. This guided exploration technique promotes convergence towards the Pareto front while maintaining good solution diversity.

A comprehensive review of existing multi-objective optimization algorithms is presented in Table 1. The table summarizes the dominance relations, key contributions, and relevance of each algorithm to the current study. While Pareto dominance is still the most commonly employed due to its simplicity and efficacy, Epsilon dominance and its variants provide more control over solution variety and convergence, especially in many-objective optimization. In contrast, stronger dominance relations address the issues of balancing convergence and diversity in complicated optimization environments. Although current algorithms such as NSGA-II, SPEA2, and MOEA/D have proved the efficacy of these dominance relationships, algorithm-specific evaluations are still required to find the most appropriate dominance strategy for emerging optimization strategies. These evaluations are critical for customizing dominance relations to the specific needs of new algorithms and problem contexts.

Table 1: Summary of the multi-objective optimization algorithms reviewed

Algorithm	Dominance relation	Key features	Relevance
NSGAII [5]	Pareto dominance	Introduce Fast non-dominated sorting, crowding distance mechanism, variety of Pareto-optimal solutions.	Baseline for comparing MOCBOA's performance and diversity preservation.
SPEA2 [6]	Pareto dominance	Demonstrate the strength value for dominance, density preservation for diversity.	Demonstrates the use of external archives, similar to MOCBOA.
MOEA/D [7]	Pareto dominance	Introduce the decomposition of problems into scalar subproblems, optimized simultaneously.	Highlights the importance of decomposition methods in multi-objective optimization.

(Continued)

Table 1 (continued)

Algorithm	Dominance relation	Key features	Relevance
MOPSO [8]	Pareto dominance	Extends PSO to handle multiple objectives using a global repository.	Provides insights into swarm intelligence-based approaches for MOCBOA.
MOGWO [9]	Pareto dominance	Adapts grey wolf optimization for multi-objective problems.	Demonstrates the use of metaheuristics in multi-objective optimization.
MOCSO [10]	Pareto dominance	Extends cat swarm optimization for multi-objective problems.	Highlights the use of bio-inspired algorithms in optimization.
MOCI [11]	Pareto dominance	Cohort Intelligence with coevolutionary design principles	Demonstrates the use of coevolutionary principles.
ε-MOEA [24]	Epsilon dominance	Introduces ε-dominance for balancing convergence and diversity.	Provides a benchmark for MOCBOA's use of Epsilon dominance.
GMOEO [12]	Epsilon dominance	Extends Equilibrium optimization. Uses ε-dominance and guided exploration for improved convergence.	Highlights the effectiveness of guided exploration, relevant to MOCBOA's design.
GMOMPA [13]	Epsilon dominance	Extends Marine predator optimization. Uses ε-dominance and crowding distance.	Demonstrates the use of Epsilon dominance in marine predator-inspired algorithms.
MOGSK [14]	Epsilon dominance	Extends GSK for multi-objective problems using ε-dominance.	Highlights the use of knowledge-sharing mechanisms, relevant to MOCBOA's design (human based).
MOMRFO [17]	Epsilon dominance	Expands manta ray foraging optimization using ε-dominance.	Demonstrates the use of Epsilon dominance in bio-inspired algorithms.
Cone-ε-MOEA [19]	Cone-epsilon-dominance	Introduces cone-ε-dominance for many-objective problems.	Provides insights into Cone-epsilon dominance, relevant to MOCBOA's comparison.
MOHHO [20]	Strengthened dominance	Uses Strengthened dominance for balancing convergence and diversity.	Demonstrates the effectiveness of Strengthened dominance, relevant to MOCBOA's design.

(Continued)

Table 1 (continued)

Algorithm	Dominance relation	Key features	Relevance
NSGAII (CSDR) [21]	Strengthened dominance	Replaces Pareto dominance with CSDR for improved performance.	Highlights the use of Strengthened dominance in NSGA-II.
NSGAII (RPS) [22]	Strengthened dominance	Introduces RPS-dominance for better stratification of solutions.	Demonstrates the use of reference points in dominance relations.
MGDE [5,23]	Strengthened dominance	Uses Strengthened dominance and bi-goal evolution for many-objective problems.	Highlights the use of Strengthened dominance in differential evolution.

Building on these insights, in this work we present an extension of the Chef-Based Optimization Algorithm (CBOA) [25] called the Multi-objective Chef-Based Optimization Algorithm (MOCBOA) to tackle multi-objective optimization problems. MOCBOA incorporates mechanisms such as an external archive, fast non-dominated sorting (FNS), and crowding distance (CD) to ensure a robust performance. A key contribution of this work is the systematic comparison of four dominance relations, Pareto dominance, Epsilon dominance, Cone-epsilon dominance, and Strengthened dominance to determine the most effective strategy for updating archive solutions in MOCBOA. This comprehensive evaluation addresses a gap in the literature, by conducting a systematic comparison of all four dominance relations.

This work's specific contributions include the following:

- **Extension of CBOA to Multi-Objective Optimization:** We propose MOCBOA, the first multi-objective extension of the Chef-Based Optimization Algorithm. MOCBOA uses an external archive to store the non-dominated solutions and lead the total solutions to the best Pareto front.
- **Diversity and Convergence:** To maintain diversity, assure solutions convergence, and guarantee an effective solution distribution, MOCBOA uses the FNS and crowding distance.
- **Systematic Comparison of Dominance Relations:** The performance of the four dominance relations Pareto dominance, Epsilon dominance, Cone-epsilon dominance, and Strengthened dominance in the context of MOCBOA is compared through an experimental study while used to update the archive solutions. In order to assist in selecting the best approach for a particular optimization problem, this study offers insights into each dominance relation.
- **Validation on Real-World Problems Benchmark:** We validate the effectiveness of MOCBOA on a wide range of Real-World constrained MOPs (RWMOPs) including mechanical design, chemical engineering, process, design, and synthesis, power electronics, and power systems.
- **Impact:** By providing a robust and versatile tool for solving complex MOPs, MOCBOA addresses real-world challenges in engineering and other domains. The algorithm's ability to adapt to different dominance relations and problem contexts makes it a valuable contribution to the field of multi-objective optimization. Our results underscore that the choice of a dominance relation significantly shapes the optimization outcomes, influencing both solution diversity and convergence toward the Pareto front.

As shown in Fig. 1, the rest of the paper is structured as follows: Section 2 covers the mathematical background of MOPs, Pareto dominance, Epsilon dominance, Cone-epsilon dominance, and Strengthened dominance. Section 3 presents the single-objective version of the CBOA algorithm. Section 4 introduces the proposed MOCBOA algorithm. Section 5 presents the findings and discussion. Finally, some conclusions are conveyed in Section 6.

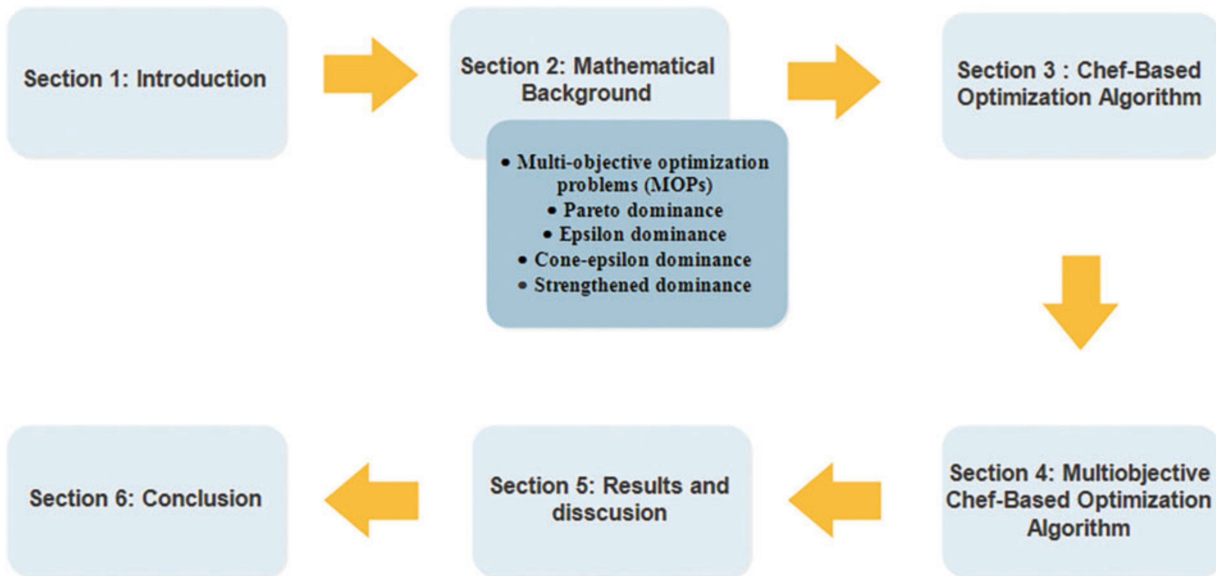

Figure 1: Paper structure

2 Mathematical Background

This section contains definitions for concepts such as information on MOPs and Pareto dominance, Epsilon dominance, Cone-epsilon dominance, and Strengthened dominance relations.

2.1 Multi-Objective Optimization Problems

Simultaneously gathering and optimizing opposing objective functions is a systematic procedure known as multi-objective optimization. Depending on the problem's nature, optimization may include maximization or minimization of an objective. Such a minimization problem can be formulated as follows:

$$Minimize\colon f_m\left(x\right); m = \left(1, 2, \ldots, M\right), \tag{1}$$

subject to:

$$\left\{g_j\left(x\right)\right\} = 0;\ j = \left(1, 2, 3, \ldots, J\right), \tag{2}$$

$$h_k\left(x\right) \leq 0;\ k = \left(1, 2, 3, \ldots, K\right), \tag{3}$$

$$lo_i \leq x_i \leq up_i,\ i = \left(1, 2, 3, \ldots, L\right), \tag{4}$$

where the solution for the L decision variables is denoted by $x = (x_1, x_2, \ldots, x_L)$, while meeting the equality constraints K for $h_k\left(x\right)$ and the inequality constraints J for $g_j\left(x\right)$. The value M denotes the number of objective functions. Furthermore, lo_i and up_i stand for lower and upper bounds, respectively.

2.2 Pareto Dominance

Using the relational arithmetic operators to compare the produced solutions in a MOP proved to be challenging. Consequently, the idea of Pareto optimal dominance offered a straightforward method for contrasting options in the multi-objective search space. Moreover, the search space had a collection of Pareto optimal solutions instead of a single optimum solution with a Pareto front image. The key concepts of the Pareto dominance relation are as follows:

Definition 1. Pareto dominance: Consider two solutions u and u'. u is considered to dominate u' (noted as: $u \succ u'$) if:

$$\forall i \in \{1, 2, 3, \ldots, M\} : f_i(u) \leq f_i(u') \; and \; \exists j \in \{1, 2, 3, \ldots, M\} : f_j(u) < f_i(u'). \tag{5}$$

Definition 2. Non-dominated set: Non-dominated solutions are those that are not dominated by any other solution. Let B be a set of solutions including non-dominated solutions as members. $B' \subseteq B$, denotes the set of solutions that are not dominated by any other solution from the set B.

Definition 3. Pareto optimal set: The Pareto optimal set represents a collection of all non-dominated solutions in the search space.

2.3 Epsilon Dominance

Laumanns et al. [24] presented the Epsilon dominance (ϵ-dominance) technique, which uses a two-level selection process to keep a representative and diverse group of solutions for multi-objective optimization. Using this method, the objective space is divided into hyper-boxes, each of which has a single solution vector. The box-level dominance relation guarantees the preservation of a set of non-dominated boxes. The objective space is first divided into hyper-boxes, and a vector B identifies each box. Every solution in the reference set $R(t)$ is represented by this vector. Simultaneously, another set of solutions, $N(t)$, is taken into consideration, wherein $B = (B_1, B_2, \ldots, B_M)$ denotes the identification vector linked to the M objectives and is expressed as follows:

$$B_i(f) = \left\lfloor \frac{\log(f_i)}{\log(\epsilon + 1)} \right\rfloor, \tag{6}$$

where ε stands for the permitted error, f_i indicates the objective cost of the $i\text{-}th$ solution, and $\lfloor . \rfloor$ indicates the absolute value.

Let $R(t)$ be updated according to Epsilon dominance using a new set of solutions denoted as $N(t)$, following the computation of the identification vectors for the new set of solutions, the ϵ-dominance relation is used to compare each new solution with the solutions from the reference set that already exist. Whether a new solution belongs to the reference set or not is decided by this comparison. To be more precise, the new solution n is admitted into the reference set, and the associated reference solution R_i is eliminated if the new solution's identification vector B_n dominates any identification vector BR_i of a solution in the reference set. In contrast, the new solution gets rejected if any of the BR_i dominates B_n. A supplementary process is used if neither of these scenarios is valid (that is, if B_n and BR_i are incomparable): if the new solution n dominates the reference solution R_i in regular dominance, then n is accepted. If no answer dominates another one, the option that is closest to B is selected. Algorithm 1 presents the update process using the Epsilon dominance relation.

Algorithm 1: Reference set update using Epsilon dominance

Input: *Reference set* $R(t)$, *t is the iteration, n a candidate from the new set of solutions* $N(t)$
Calculate vector B_n, *calculate vector BR for all Reference set solutions* $R(t)$,
  If $\exists\, x \in R(t)\,|B_x \succcurlyeq B_n$ *then*
    n is not accepted
  End if
  If $\exists\, x \in R(t)\,|B_n \succcurlyeq B_x$ *then*
    n replaces x in $\boldsymbol{R(t)}$
  End if
  If *neither of the preceding cases occurs* ***then***
    If $\exists\, x \in R(t)\,|B_n \sim B_x$ *then*
      If $n \sim x$ *then*
        Keep the solution with the shortest distance to the identifier vector B
      Else
        Choose the solution that dominates another one
      End if
    Else
      Add the solution n to the Reference set $R(t)$
    End If
  End If

2.4 Cone-Epsilon Dominance

Cone-epsilon dominance, a more flexible form of Pareto dominance that falls under the larger category of relaxed dominance, was first proposed by Batista et al. [19]. Ikeda et al. [26] first proposed relaxed dominance to deal with situations in which a solution has a relatively low value in one objective, making it challenging to dominate. This was further refined by Ikeda et al. by adaptation of the α-dominance technique, which defines upper and lower bounds between two primary objectives using linear trade-off functions. This method enables a more flexible selection of non-dominated solutions, allowing a solution that would normally be rejected by standard Pareto dominance to dominate another one if it is significantly better in one objective but slightly inferior in another one.

Cone-epsilon dominance, which is based on relaxed dominance, is a hybrid form of both α- and ϵ-dominance. It makes use of cones to manage the dominance zone while attempting to maintain the convergence property of ϵ-dominance.

Definition 4. Cone: A set S is said to be a cone *if* $\lambda x \in S$ for any $x \in S\ \forall\ \lambda \geq 0$.

Definition 5. Generated cone for the vectors: u_1 and u_2: the generated cone is the set:

$$S = \{z{:}\, z = \lambda_1 u_1 + \lambda_2 u_2, \forall \lambda_1, \lambda_2 \geq 0\}\,, \tag{7}$$

With respect to dimension m, and considering u_i $(i \in \{1, 2, \ldots, m\})$, it can be rewritten as follows:

$$S = \{z{:}\, z = \lambda_1 u_1 + \ldots + \lambda_m u_m + \ldots + \lambda_i u_i, \forall \lambda_i \geq 0\}\,. \tag{8}$$

Furthermore, in relation to the origins of the box $u_1 = [\varepsilon_1 k \varepsilon_2]^T$ and $u_2 = [k \varepsilon_1 \varepsilon_2]^T$, the cone can be written as follows:

$$S = \left\{ z : \overbrace{\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}}^{z} = \overbrace{\begin{bmatrix} \varepsilon_1 k \varepsilon_2 \\ k \varepsilon_1 \varepsilon_2 \end{bmatrix}}^{\psi} \overbrace{\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}}^{\lambda}, \forall \lambda_1, \lambda_2 \geq 0 \right\}, \tag{9}$$

where ψ represents the cone dominance matrix, and $k \in [0, 1)$ is a parameter that controls the cone opening.

Let the set $R(t)$ be updated according to Cone-epsilon dominance using a new set of solutions denoted as $N(t)$, Cone-epsilon dominance creates hyper-boxes in the objective space, much like ϵ-dominance does. Additionally, it adjusts to two types of dominance: regular Pareto dominance and box-level dominance. With upper and lower bounds similar to the property adapted in α-dominance, the box contains a single unique identification vector, b. Each solution derived from *R(t)* and *N(t)* is now allocated a box, and b is determined as follows:

$$b_i(x) = \begin{cases} \varepsilon_i \left\lfloor \dfrac{x_i}{\varepsilon_i} \right\rfloor, i \in \{1, 2, \ldots, m\} \ for\,minimizing\, f_i \\ \varepsilon_i \left\lceil \dfrac{x_i}{\varepsilon_i} \right\rceil, i \in \{1, 2, \ldots, m\} \ for\,maximizing\, f_i \end{cases}, \tag{10}$$

where $\lceil . \rceil$ returns the closest upper integer to their argument and $\lfloor . \rfloor$ returns the closest lower integer.

Dominance at the box level is now necessary to add a solution to the reference set. Consider two identification vectors of the $N(t)$ solutions and the $R(t)$ solutions, respectively, be b_n and b_r. The b_n is said to cone-ε-dominate b_r only and only if b_n Pareto dominates b_r or the solution to the linear system is $\psi\lambda = z$, with $z = b_r - [b_n - \varepsilon]$, $\varepsilon_i > 0$, gives $\lambda_i \geq 0, \forall i \in \{1, \ldots, m\}$. Similarly, we say $b_n \prec^{cone\varepsilon} b_r$ if and only if:

$$(b_n \prec b_r) \vee (\psi\lambda = z | \lambda_i \geq 0, i \in \{1, \ldots, m\}) \tag{11}$$

A regular Pareto dominance is used when the identification vectors have the same box; the reference set solution is substituted only if the new set solution also exhibits Pareto dominance. If not, the closest point to the box's origin is selected. The process for updating the reference set utilizing Cone-epsilon dominance is displayed in Algorithm 2.

Algorithm 2: Reference set update using Cone-epsilon dominance

Input: *Reference set R, Candidate solution n from a new set*
Calculate vector b_n, calculate vector b for all reference set solutions R,
  If *n is cone-ε-dominated by any $r \in R$ then*
    Reject n
  Else If *n share the same box as reference set solution r then*
    If *n dominates r or is closer to the origin of the box than r, then*
      Delete all archive members cone-ε-dominated and
      Replace r by n
    Else
      Reject n
    End If

(Continued)

Algorithm 2 (continued)

Else If *n cone-ε-dominates any* $r \in R$, *then*
Delete all of the cone-ε-dominated reference set members
Insert n in the reference set.
Else
Insert n in the reference set.
End If

2.5 Strengthened Dominance Relation (SDR)

As proposed by Tian et al. [27], the Strengthened dominance relation employs a customized niching strategy to achieve an optimal balance between the non-dominated solution set's diversity and convergence. A potential solution a is considered to be more dominant than another potential solution b by SDR (represented as $a \prec_{SDR} b$) if and only if:

$$\begin{cases} Con(a) < Con(b), \theta_{ab} \le \overline{\theta} \\ Con(a).\left(\dfrac{\theta_{ab}}{\overline{\theta}}\right) < Con(b), \theta_{ab} > \overline{\theta} \end{cases}, \quad (12)$$

where $Con(a)$ determines the degree of convergence of a, and it is expressed as Eq. (13). Furthermore, the acute angle between the two candidate solutions' objective values is denoted by θ_{ab} according to Eq. (14).

$$Con(a) = \sum_{i=1}^{M} f_i(a), \quad (13)$$

$$\theta_{ab} = \arccos(f(a), f(b)), \quad (14)$$

where $\overline{\theta}$ indicates the niche size that each potential solution falls within. As per the first formula of Eq. (12), a is considered to dominate b when a has a lower degree of convergence than b and the angle between a and any candidate solution b is less than $\overline{\theta}$. According to the second formula of Eq. (12), the likelihood that a will dominate b is negatively correlated with the angle θ_{ab}, however, if two candidate solutions, a and b, do not belong to the same niche, a can dominate b if a has a lower degree of convergence than b. As a result, the SDR relation does not penalize solution a for choosing the solution with good convergence if the two candidate solutions, a and b, are in the same niche. If they are not, the SDR relation penalizes solution a by a factor of $\theta_{ab}/\overline{\theta}$ to preserve diversity within the set of selected solutions. This dominance relation thereby facilitates improvements in both the convergence and diversity of the solutions.

3 Chef-Based Optimization Algorithm (CBOA)

Before introducing the extended proposed version (MOCBOA), we first introduce the original single-objective version. The CBOA [25] is a recently presented metaheuristic algorithm, inspired by the culinary school learning process. CBOA's innovative approach is inspired by culinary training. Training programs offered by culinary schools are intended to improve students' abilities and help them become skilled cooks. This iterative refinement of viable solutions to arrive at an ideal result is akin to the process of continuous improvement in metaheuristic algorithms. The conversion of student cooks into professional chefs is embodied by the CBOA framework, which includes various crucial components:

- **Chef Teachers:** Each chef teacher at a cooking school is in charge of leading a class.

- **Class Selection:** Students get to pick whatever classes they want to take, and their chef instructors will teach them a variety of cooking techniques.
- **Skill Refinement:** With one-on-one instruction and supervision from a master chef instructor, chef instructors continuously enhance their abilities.
- **Students Learning:** Students who study cooking strive to imitate and refine the methods that their chef instructors teach them, honing their skills via experience.
- **Graduation:** After completing the program, culinary students are skilled cooks who have honed their craft through instruction.

Mathematical Model

The CBOA is a population-based algorithm that is divided into two groups: chefs and cooking students. Every component of the CBOA encapsulates details about the problem variables and suggests a potential solution. Each member of the CBOA is represented mathematically as a vector and can be modeled in a matrix. First, the population is initialized randomly, followed by the computation of the objective function for each member of the population. Then, the selection process of the different groups of chefs/students is done according to the objective function values, where the best results obtained will be selected as chefs while the remaining are selected as students. The population members of COBA must be updated with each iteration because the process is iterative. The update procedure varies depending on the group (chef/student) according to the following two phases:

- **Phase One:** The competitive aspect of culinary expertise is shown by chefs striving to create quality ratings.
- **Phase Two:** Students compete based on how well they can cook to get their quality ratings.

In phase one, the chefs adapt two strategies for an update: strategy one entails modeling themselves after the best chef teacher to learn their techniques, improves the algorithm by not depending only on the highest-ranking member of the population to train students, but also allowing the best chefs to hone their skills before instructing pupils. Additionally, through the prevention of the algorithm becoming stuck in local optima, this method allows the search space to be scanned more precisely and effectively. Eq. (15) determines each culinary teacher's new position for $i = 1, 2, \ldots, N_C$ with N_C be the number of chefs and $j = 1, 2, \ldots, m$.

$$xs_{i,j}^{C/S1} = xs_{i,j} + r.\left(BC_j - I.xs_{i,j}\right), \tag{15}$$

where $xs_{i,j}^{C/S1}$ is the $j\text{-}th$ coordinate of the newly calculated position based on the first strategy (C/S1) of updating the chef instructor for the $i\text{-}th$ sorted member of the algorithm, BC_j is the $j\text{-}th$ coordinate of the best chef instructor, r is a random number from the interval [0, 1], and I is a number that is selected randomly during the execution from the set $\{1, 2\}$. This new position is acceptable if it improves the value of the objective function according to:

$$XS_i = \begin{cases} XS_{i,j}^{C/S1}, FS_{i,j}^{C/S1} < F_i \\ XS_i, else \end{cases} \tag{16}$$

where $FS_{i,j}^{C/S1}$ is the objective function value associated with $xs_{i,j}^{C/S1}$.

In the second phase, through individualized exercises and activities, the chef instructor aims to enhance his cooking abilities. attempts to find better solutions close to its location, regardless of the positions of other members of the population. It is possible that local search and exploitation, with slight adjustments to population members' positions in the search space, can yield superior results. This idea states generating a random position around each instructor in the search space. This random position is appropriate for updating if it improves the objective function value.

$$xs_{i,j}^{C/S2} = xs_{i,j} + lb_j^{local} r + \left(ub_j^{local} - lb_j^{local}\right), i = 1, 2, \ldots, N_C, j = 1, 2, \ldots, m, \tag{17}$$

$$XS_i = \begin{cases} XS_{i,j}^{C/S2}, FS_{i,j}^{C/S2} < F_i \\ XS_i, else \end{cases}, \tag{18}$$

where $ub_j^{local} = \frac{ub_j}{t}$ is an upper bound, $lb_j^{local} = \frac{lb_j}{t}$ is a lower bound, t is the iteration count, $XS_{i,j}^{C/S2}$ is the new position, $C/S2$ is the chef updating according to the second strategies and $FS_{i,j}^{C/S2}$ is the corresponding objective function value.

In the second phase, students adapt three tactics for updating: The first one is that each cooking student selects a class taught by a chef at random, and this chef instructor then teaches him cooking techniques. This results in students learning from the best and moving in the search space according to Eq. (20) and moving the students' positions if the new objective function value is better:

$$XS_{i,j}^{S/S1} = xs_{i,j} + r.(CI_{k_i,j} - I.xs_{i,j}), \tag{19}$$

$$XS_i = \begin{cases} XS_{i,j}^{S/S1}, FS_{i,j}^{S/S1} < F_i \\ XS_i, else \end{cases}, \tag{20}$$

where CI_j is the selected chef, S/S1 is the students' first updating strategy, and k_i is randomly selected from the set $\{1, 2, \ldots, NC\}$.

The second tactic adapts a "skill" based update, where a random chef is selected and one of his variables is taken to replace one of the student's variables. This variable is regarded as a skill.

$$XS_{i,j}^{S/S2} = \begin{cases} CI_{k_i,j}, j = l; \\ xs_{i,j}, else \end{cases}, \tag{21}$$

$$XS_i = \begin{cases} XS_{i,j}^{S/S2}, FS_{i,j}^{S/S2} < F_i \\ XS_i, else \end{cases}. \tag{22}$$

The third updating tactic focuses on local search, where students try to improve all of their variables. Using a randomly generated position close to the student, similar to chef's second strategy (Eqs. (17) and (18)) a new position is generated as follows:

$$XS_i = \begin{cases} XS_{i,j}^{S/S3}, FS_{i,j}^{S/S3} < F_i, \\ XS_i, else \end{cases} \tag{23}$$

where $FS_{i,j}^{S/S3}$ is the objective function value for the new position $XS_{i,j}^{S/S3}$, and S/S3 is the students' third updating strategy.

The above-mentioned whole process is repeated until the stopping criterion is met.

4 Multi-Objective Chef-Based Optimization Algorithm (MOCBOA)

This section provides a detailed outline of the proposed MOCBOA algorithm. The core of MOCBOA is a mechanism for identifying non-dominated solutions by storing them in an external archive. The Pareto dominance relation plays a fundamental role in distinguishing non-dominated solutions, while the crowding distance metric enhances diversity among the solutions, promoting a well-distributed set along the Pareto front. Additionally, the FNS is incorporated to facilitate the generation of multiple Pareto fronts. To ensure that the archive remains up-to-date with the most relevant solutions, MOCBOA uses a combination of dominance relations to compare and refine the entries. In addition to the Pareto dominance, we employ Epsilon dominance to allow a degree of tolerance in solution comparisons, Cone-epsilon dominance to focus on solutions within a specific performance range, and Strengthened dominance to prioritize certain solution characteristics. Together, these strategies enable a robust approach to maintaining an archive that reflects a balanced and high-quality solution set throughout the optimization process. In summary, the employed approach covers multiple key points:

- an external repository that can direct the candidates towards an optimal set and hold the best non-dominated solutions.
- The update process of the archive's solutions uses dominance relations, Epsilon dominance relations, Cone-epsilon dominance, and Strengthened dominance relations, and a comparison between them is established.
- Crowding distance and FNS are considered to ensure a well-diversified set of solutions and an effective convergence towards the Pareto optimal.

The archive plays a crucial role in guiding solutions and preserving diversity in multi-objective optimization algorithms, helping to produce an adequate set of solutions and to maintain an appropriate balance amid exploration and exploitation. The quality and variance of the archive are vital to the algorithm's performance as it contains and updates the best solutions discovered during the optimization process. It is crucial to experiment with various dominance relations since the selection of a relation might have a big influence on the optimization process. Though Pareto dominance is the simplest and the most used one, Epsilon dominance introduces a tolerance margin, which increases flexibility and diversity. Cone-epsilon dominance is helpful when certain objectives are prioritized since it expands on this by concentrating on particular areas of the objective space. By boosting solutions in less crowded regions and limiting premature convergence, the Strengthened dominance promotes diversity and convergence. Thus, a comparison of different dominance relations is required to identify the most efficient one, since the optimal dominance relation for the archive population can greatly increase the algorithm's efficiency by guaranteeing well-distributed and high-quality solutions. On the other side, the crowding distance and FNS are essential elements of multi-objective optimization algorithms. FNS effectively groups solutions into distinct Pareto fronts, facilitating the algorithm's rapid identification and prioritization of non-dominated solutions. Conversely, the crowding distance promotes diversity by giving priority to solutions in less crowded places and quantifying the density of solutions near a particular point in the objective space. By properly balancing exploration and exploitation, these strategies guarantee that the algorithm not only converges towards the Pareto-optimal front but also preserves a diversified variety of solutions.

The proposed MOCBOA operates by going through a set of defined steps that are intended to evaluate candidates, systematically explore the solution space, and iteratively refine the population toward optimal solutions. Following establishing the initial population and its evaluation against the M objectives. we began by applying the FNS to the initial population, followed by the crowding distance calculation to identify the most optimal solutions. These top-performing solutions then formed the core of our Selected/Candidate population. After that, we made use of an external archive known as the archive population, which initially contained the first selected population's non-dominated solution. During every step of the optimization process, this archive kept the best result or the non-dominated set. To achieve Pareto set solutions in every iteration, the algorithm has to carry out the subsequent actions: update the selected population, update the archive population, and then update the population position.

4.1 Selected Population Updating

This population is important since it has a direct impact on the upcoming archive update phase. Its efficacy influences the diversity and quality of the solutions that are kept, which influences the optimization process as a whole. The algorithm iterates through a combined pool of previous, current, and archived populations, selecting the top solutions and systematically evaluating and classifying them to retain the best candidates for further analysis.

- **Create a pool of potential candidates:** This pool is created based on combining the previous, current and archive population in order to preserve diversity, maintain a well distributed set of solutions and prevent premature convergence.
- **Sorting the pool:** once the pool is created, the FNS and crowding distance are implemented to sort the solutions accurately according to ranks, after comparing the solutions individually and identifying the non-dominated solutions by checking the two parameters n_i the number of solutions that dominate the solution i and S_i the set of solution it dominates. Therefore, if $n_i = 0$, then those solutions are excluded in a separate front named $F1$, after that the whole process is repeated to find the next front solutions.

Algorithm 3 describes the steps adapted by FNS. Once the FNS finishes and to maintain the diversity, the crowding distance is used by calculating for each solution the mean distance to its two surrounding solutions of its cuboid, and ranking them according to this distance. The formulation of the crowding distance is as follows:

$$CD^i_{fj} = \frac{f_j^{i+1} + f_j^{i-1}}{f_j^{MAX} + f_j^{MIN}} \tag{24}$$

where CD^i_{fj} is the crowding distance of the i-th solution, f_j is the j-th objective function value for the neighboring solutions $i+1$ and $i-1$ of solution i. The max and min values for the j-th objective function are represented by f_j^{MAX} and f_j^{MIN}, respectively, as shown in Fig. 2.

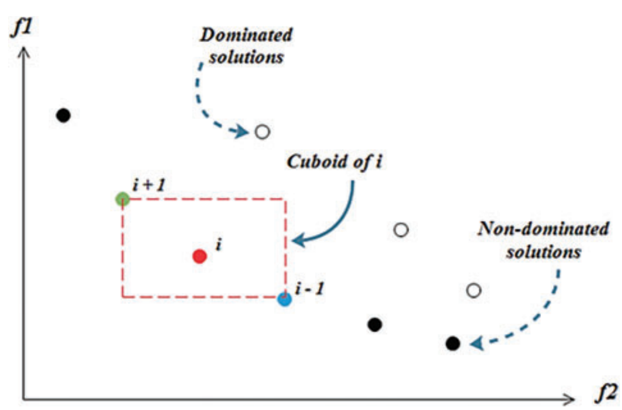

Figure 2: Crowding distance

After calculating the crowding distance of each solution and for each objective function, the solutions are sorted by their crowding distance values that were found in ascending order.

Algorithm 3: Fast non-dominated sort (FNS)

Input: X
For *each* $x \in X$***do***
  For *each* $y \in X$***do***
    If x *dominates* y***then***
      Add y *to* S_x
    Else
      If x *is dominated by* y, *then*
        Increase the count n_x
      If *no solution dominated* x, *then*
        x *belongs to the first front* F
        While $F_i \neq 0$ *do*
          For *each individual* $pinF_i$ *do*
            For *modify each member from the set* S_x *do: Decrease the count* n_y *by one*
              If $n_y = 0$, *for the member* y *in* H, *then*
                Current front is formed by the members of the H
End

4.2 Archive Updating

The archive population is essential to maintaining the best solutions discovered during the process of optimization. It helps direct the search by influencing subsequent decisions and stores non-dominated solutions to make sure they are not lost. Additionally, the archive preserves diversity by preserving a range of solutions and progresses over time by demonstrating how the quality of solutions increases. It also makes it possible to compare several dominance strategies, including Strengthened dominance, Cone-epsilon dominance, Epsilon dominance, and Pareto dominance. The most optimal and varied set of solutions can be obtained by utilizing one dominant method at a time and examining the archive. Since a clearly defined dominance relation improves convergence, making sure that the optimization process proceeds gradually in the direction of better solutions while maintaining population variety and the search space can be thoroughly explored.

In the update process of the archive, the previously mentioned dominance strategies were used, one at a time. Once the selected population is obtained, it participates in this process by comparing one solution of the archive against the whole selected population. If the archive solution dominates, then it stays in the archive, if not it is replaced by the solution from the selected population. To control the size of the archive and ensure a smooth optimization process, the archive size is limited to the first N solutions.

4.3 Population Updating

Updating the population position to move on with the optimization process is the same as the one employed in the version with only one objective. However, to direct the population towards the Pareto optimal set, the archive population is used in the process. The archive population has been incorporated according to each strategy.

4.3.1 Chef Instructors Update Process

Strategy 1: Modelling themselves after the best chef teacher: This strategy adapts the same equation as the single-objective version using a randomly selected solution of the archive population in the process as follows:

$$xs_{i,j}^{C/S1} = xs_{i,j} + r.(A_{sol} - I.xs_{i,j}), \tag{25}$$

where this new position is acceptable to the MOCBOA if it dominates as follows:

$$XS_i = \begin{cases} XS_{i,j}^{C/S1}, FS_{i,j}^{C/S1} \succ F_i \\ XS_i, else \end{cases}. \tag{26}$$

Strategy 2: Through individualized exercises and activities: The single-objective version uses a local search and exploitation, with slight adjustments to population members' positions in the search space, which can yield superior results. This idea states generating a random position around each instructor in the search space. However, since the archive holds the best solutions discovered so far. it is used to help guide the solution towards the optimal results. In the MOCBOA algorithm, instead of generating a local random solution, a random solution A_{sol} is chosen from the archive and used as follows:

$$xs_{i,j}^{C/S2} = xs_{i,j} + A_{sol}, i = 1, 2, \ldots, N_C, j = 1, 2, \ldots, m, \tag{27}$$

$$XS_i = \begin{cases} XS_{i,j}^{C/S2}, FS_{i,j}^{C/S2} \succ F_i \\ XS_i, else \end{cases}. \tag{28}$$

4.3.2 Students Update Process

Strategy 1: Selecting a random chef and learning a skill: In this strategy, each cooking student selects a class taught by a chef at random, and this chef instructor then teaches cooking techniques. It results in students learning from the best and moving in the search space. In the MOCBOA algorithm, instead of selecting a chef, a solution is selected from the archive as A_{sol} as follows:

$$XS_{i,j}^{S/S1} = xs_{i,j} + r.(A_{sol} - I.xs_{i,j}). \tag{29}$$

Accordingly, the student's position is moved, if the new objective function value dominates:

$$XS_i = \begin{cases} XS_{i,j}^{S/S1}, FS_{i,j}^{S/S1} \succ F_i \\ XS_i, else \end{cases}. \tag{30}$$

Strategy 2: Chef selection: One of the variables within the chef is taken to replace by one of the student's variables. Here, a random variable within a solution from the archive A_{sol,k_i} is used as:

$$XS_{i,j}^{S/S2} = \begin{cases} A_{sol,k_i,j}, j = l; \\ xs_{i,j}, else \end{cases}, \tag{31}$$

$$XS_i = \begin{cases} XS_{i,j}^{S/S2}, FS_{i,j}^{S/S2} \succ F_i \\ XS_i, else \end{cases}. \tag{32}$$

Strategy 3: Focus on a local search, where students try to improve all of their variables: Using a randomly generated position close by the student, similar to the chef's second strategy, a new position is generated. As for the MOCBOA algorithm, instead of a random local solution, an archive solution A_{sol} is used as follows:

$$xs_{i,j}^{S/S3} = \begin{cases} xs_{i,j} + A_{sol}, J = q; \\ xs_{i,j}, J \neq q \end{cases}, \tag{33}$$

$$XS_i = \begin{cases} XS_{i,j}^{S/S3}, FS_{i,j}^{S/S3} > F_i \\ XS_i, else \end{cases}, \tag{34}$$

where q is a random number from $\{1, 2, \ldots, m\}$, $i = NC + 1, NC + 2, \ldots, N$, & $j = 1, 2, \ldots, m$.

This process is repeated until the stopping criterion is met. Fig. 3 shows the overall flowchart of the proposed MOCBOA algorithm, and Algorithm 4 describes its pseudo-code.

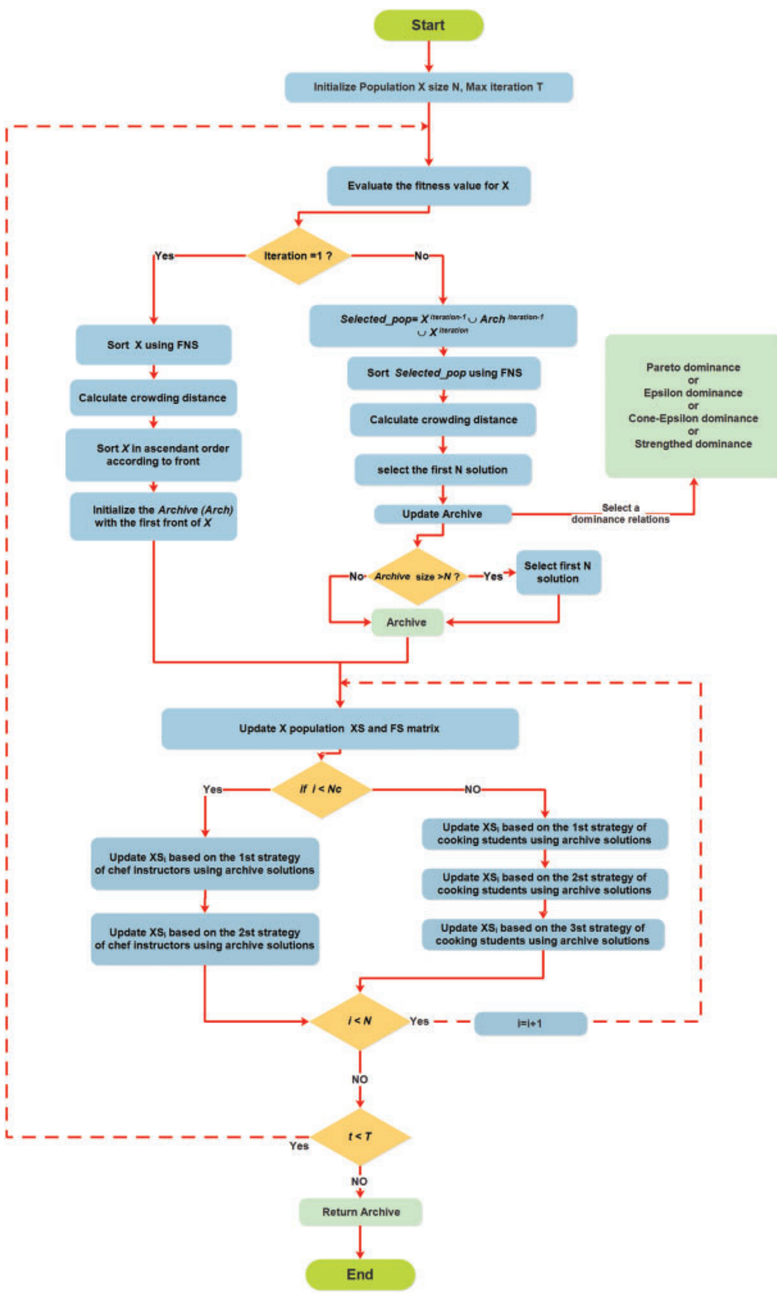

Figure 3: Flowchart of the proposed MOCBOA algorithm

The computational complexity of the MOCBOA algorithm was analyzed by defining N as the population size and M as the number of objectives. The main loop of the algorithm shares the same complexity as the CBOA algorithm [25], which operates in polynomial time of $O(tdN + tcN)$, where t represents the number of iterations, c is the cost function, and d is the problem dimension. The first operation, fast non-dominated sorting (FNS), has a complexity of $O(MN^2)$. The second operation, crowding distance calculation, has a complexity of $O(2Nlog(2N))$. Updating the archive involves several dominance relations depending on the selected one:

- **Pareto dominance**, the most commonly used relation, has a complexity of $O(MN^2)$ due to pairwise comparisons of solutions.
- **Epsilon dominance (ε-dominance)**, also has a complexity of $O(MN^2)$.
- **Cone-ε-dominance**, an extension of ε-dominance, incurs a complexity of $O(M^2N)$.
- **Strengthened dominance relation**, which enhances the selection process by balancing convergence and diversity, typically has a complexity of $O(MN^2)$.

As a result, the overall processing time of GMOEO is determined by the maximum of these complexities: $max\left[O\left(tdN + tcN\right), O\left(MN^2\right), O\left(2Nlog\left(2N\right)\right), O\left(MN^2\right), O\left(M^2N\right), O\left(MN^2\right)\right]$. However, the dominant complexity of MOCBOA is $O(MN^2)$, which is consistent with the complexities of MOPSO [8] and MOGWO [9].

Algorithm 4: Multi-objective chef-based optimization algorithm (MOCBOA)

Initialize the search agent N, max iteration T
Generate a random population X
For *t = 1 to T*
    For *i = 1 to N*
        Evaluate the fitness function for X
    End
    If *t = 1* ***then***
        Sort X by FNS
        Generate the values of crowding distance
        Set the Archive with the non-dominated solutions of X
    Else
        Selected population = $X^{t-1} \cup X^{t} \cup Archive^{t-1}$
        Sort Selected population using FNS
        Generate the values of crowding distance
        Select the first N solutions for selected population
        Update Archive using Pareto ***or*** *Epsilon* ***or*** *Cone-epsilon* ***or*** *Strengthened dominance*
            If *Archive size > N*
                Select the first N solution of the Archive
            End
    End
    Phase 1: Chef instructors update process
    For *i = 1 to* N_C
        Update X based on the 1st strategy of chef instructors using Archive A_{sol}*: Eqs. (25) and (26)*
        Update X based on the 2nd strategy of chef instructors using Archive A_{sol}*: Eqs. (27) and (28)*

(Continued)

Algorithm 4 (continued)

End
End Phase 1
Phase 2: Student update process
For *i* = +1 *to N*
Update X based on the 1st strategy of student using Archive A_{sol}: *Eqs. (29) and (30)*
Update X based on the 2nd strategy of student using Archive A_{sol}: *Eqs. (31) and (32)*
Update X based on the 3rd strategy of student using Archive A_{sol}: *Eqs. (33) and (34)*
End
End Phase 2
End
Return Archive

5 Results and Discussion

In this section, we validate the proposed algorithm through experiments on real-world engineering problems, known as RWMOPs. These benchmark problems span a variety of engineering and scientific domains, compiled from diverse fields [12]. The test suite includes mechanical design problems (R1–R21), chemical engineering problems (R22–R24), process design and synthesis problems (R25–R29), power electronics problems (R30–R35), and power system problems (R36–R50), in total 50 test cases, as detailed in Table 2.

The main metric used to quantify the performance of the proposed algorithm is the hypervolume indicator (HV) [28] which measures both the convergence and diversity of the solutions, a high HV indicates a better performance. HV evaluates the results of an optimization process by concurrently considering the proximity of points to the Pareto front, as well as diversity and spread. HV also referred to as the S measure, represents the volume in the objective space that is dominated by the Pareto front concerning the reference point $r \in \mathcal{R}^m$ as a constraint, for all $z \in S$ such that $z \prec r$. The HV metric is identified as Eq. (35), with λ_m as the m-dimensional Lebesgue measure.

$$HV(S, r) = \lambda_m \left(\bigcup_{z \in S} [z; r] \right), \tag{35}$$

We assess the effectiveness of four distinct dominance strategies on the performance of the MOCBOA algorithm: Pareto, Epsilon, Cone-epsilon, and Strengthened dominance. For each test problem, we conduct a comprehensive analysis, presenting the best, worst, average, and median outcomes and comparing standard deviations to evaluate the consistency of the solutions. Finally, we examine overall patterns observed across the test problems, comparing the performance of each dominance approach against the others, with a focus on identifying which technique yields the best outcomes in terms of the hypervolume (HV) metric. The experimental settings are standardized to 50 search agents, 10 independent runs, and a maximum of 30,000 function evaluations (i.e., 600 iterations).

Table 2: Real-world problems

Problem	Description	Problem	Description
R1	Design of pressure vessels [29]	R2	Design of vibrating platform [30]

(Continued)

Table 2 (continued)

Problem	Description	Problem	Description
R3	Design of two bar truss [31]	R4	Design of welded beam [32]
R5	Disc brake design [33]	R6	Speed reducer design [34]
R7	Gear train design [35]	R8	Car side impact design [36]
R9	Four bar plane truss [37]	R10	Two bar plane truss
R11	Water resources management	R12	Simply supported I-beam design [38]
R13	Gear box design	R14	Multiple disk clutch brake design [39]
R15	Spring design [29]	R16	Cantilever beam design [40]
R17	Bulk carrier design [41]	R18	Front rail design [42]
R19	Multi-product batch plant [43]	R20	Hydrostatic trust baring design [44]
R21	High-speed train crash energy management [45]	R22	Haverly's pooling test [46]
R23	Reactor network design [47]	R24	Heat exchanger network design [48]
R25	Process synthesis problem [49]	R26	Process synthesis and design [50]
R27	Process flow sheeting problem [51]	R28	Two reactor problem [49]
R29	Process synthesis problem [49]	R30	Synchronous optimal pulse-width modulation in 3-level inverters [52]
R31	Synchronous optimal pulse-width modulation in 5-level inverters [53]	R32	Synchronous optimal pulse-width modulation of 7-leval inverters [54]
R33	Synchronous optimal pulse-width modulation of 9-leval inverters [55]	R34	Synchronous optimal pulse-width modulation of 11-level inverters [56]
R35	Synchronous optimal pulse-width modulation of 13-level inverters [56]	R36	Optimal sizing of single-phase DG with Reactive power support to decrease active power loss [57]
R37	Optimal sizing of single-phase DG with reactive power support to decrease reactive power loss [57]	R38	Optimal sizing of single-phase DG with reactive power support to decrease active and reactive power loss [57]
R39	Optimal sizing of single-phase DG with reactive power support to decrease active and reactive power loss [57]	R40	Optimal power flow to decrease active and reactive power loss [58]
R41	Optimal power flow for reducing voltage deviation, active and reactive power loss [59]	R42	Optimal power flow to decrease voltage deviation, and active power loss [60]

(Continued)

Table 2 (continued)

Problem	Description	Problem	Description
R43	Optimal power flow to decrease fuel cost, and active power loss [61]	R44	Optimal power flow to decrease fuel cost, and active and reactive power loss [62]
R45	Optimal power flow to decrease fuel cost, voltage deviation and active power loss [58]	R46	Optimal power flow to decrease fuel cost, voltage deviation, active and reactive power loss [58]
R47	The problem of optimal droop setting to decrease active and reactive power loss [63]	R48	Optimal droop setting to decrease voltage deviation and active power loss [64]
R49	Optimal droop setting to decrease voltage deviation, active, and reactive power loss [65]	R50	Power distribution system planning [66]

5.1 Mechanical Design Problems

Table 3 summarizes the results of MOCBOA across the four dominance techniques to gain a clearer picture of their relative performance on mechanical design problems. Furthermore, Figs. 4 and 5 visually illustrate the performance trends of each dominance technique across the benchmark problems, providing a comparative view of both consistency and variability in results.

Table 3: Results for the mechanical design problems R1–R21

		R1	R2	R3	R4	R5	R6	R7
MOCBOA Pareto dominance	Worst	9.52E − 01	1.00E + 00	0.00E + 00	7.71E − 01	5.89E − 01	3.04E − 01	3.84E − 01
	Best	8.11E − 01	1.35E − 01	0.00E + 00	6.58E − 01	4.43E − 01	2.53E − 01	0.00E + 00
	Average	8.63E − 01	7.13E − 01	0.00E + 00	7.23E − 01	5.60E − 01	2.86E − 01	2.06E − 01
	Median	8.46E − 01	8.16E − 01	0.00E + 00	7.26E − 01	5.77E − 01	2.86E − 01	2.41E − 01
	Std	4.70E − 02	3.07E − 01	0.00E + 00	3.36E − 02	4.39E − 02	1.59E − 02	1.37E − 01
MOCBOA Epsilon dominance	Worst	1.00E + 00	1.00E + 00	0.00E + 00	7.86E − 01	5.94E − 01	3.05E − 01	3.79E − 01
	Best	9.92E − 01	7.40E − 01	0.00E + 00	6.24E − 01	4.89E − 01	2.78E − 01	0.00E + 00
	Average	9.98E − 01	9.04E − 01	0.00E + 00	7.31E − 01	5.71E − 01	2.97E − 01	1.37E − 01
	Median	9.98E − 01	9.41E − 01	0.00E + 00	7.41E − 01	5.87E − 01	3.00E − 01	9.24E − 02
	Std	2.16E − 03	9.19E − 02	0.00E + 00	5.51E − 02	3.51E − 02	8.32E − 03	1.30E − 01
MOCBOA Cone-epsilon dominance	Worst	9.81E − 01	8.65E − 01	0.00E + 00	7.61E − 01	5.51E − 01	2.89E − 01	2.10E − 01
	Best	8.00E − 01	1.08E − 02	0.00E + 00	5.92E − 01	4.19E − 01	2.51E − 01	0.00E + 00
	Average	8.80E − 01	4.23E − 01	0.00E + 00	6.64E − 01	4.76E − 01	2.67E − 01	6.63E − 02
	Median	8.93E − 01	3.80E − 01	0.00E + 00	6.64E − 01	4.83E − 01	2.65E − 01	5.64E − 02
	Std	6.67E − 02	2.33E − 01	0.00E + 00	5.39E − 02	4.02E − 02	1.15E − 02	6.31E − 02
MOCBOA Strengthened dominance	Worst	9.19E − 01	1.00E + 00	0.00E + 00	7.40E − 01	5.89E − 01	3.04E − 01	3.90E − 01
	Best	0.00E + 00	1.16E − 01	0.00E + 00	5.46E − 01	5.05E − 01	2.74E − 01	6.77E − 02
	Average	3.38E − 01	7.90E − 01	0.00E + 00	6.96E − 01	5.63E − 01	2.93E − 01	2.76E − 01
	Median	2.04E − 01	1.00E + 00	0.00E + 00	7.04E − 01	5.81E − 01	2.95E − 01	3.04E − 01
	Std	3.31E − 01	3.47E − 01	0.00E + 00	5.65E − 02	3.37E − 02	1.01E − 02	1.07E − 01

(Continued)

Table 3 (continued)

		R8	R9	R10	R11	R12	R13	R14
MOCBOA Pareto dominance	Worst	1.92E − 02	3.40E − 01	8.35E − 01	7.17E − 02	4.81E − 01	8.61E − 02	6.05E − 01
	Best	1.59E − 02	2.90E − 01	7.70E − 01	4.65E − 02	3.74E − 01	6.54E − 02	3.60E − 01
	Average	1.69E − 02	3.17E − 01	8.08E − 01	6.11E − 02	4.26E − 01	8.02E − 02	5.25E − 01
	Median	1.65E − 02	3.15E − 01	8.09E − 01	6.13E − 02	4.21E − 01	8.28E − 02	5.46E − 01
	Std	9.64E − 04	1.56E − 02	1.95E − 02	6.76E − 03	2.86E − 02	6.74E − 03	8.11E − 02
MOCBOA Epsilon dominance	Worst	1.80E − 02	3.40E − 01	8.31E − 01	7.29E − 02	4.62E − 01	8.90E − 02	5.89E − 01
	Best	1.57E − 02	2.96E − 01	7.59E − 01	6.61E − 02	3.92E − 01	7.78E − 02	4.03E − 01
	Average	1.68E − 02	3.19E − 01	8.12E − 01	6.92E − 02	4.22E − 01	8.42E − 02	5.10E − 01
	Median	1.65E − 02	3.21E − 01	8.20E − 01	6.94E − 02	4.20E − 01	8.45E − 02	5.34E − 01
	Std	9.79E − 04	1.66E − 02	2.13E − 02	1.98E − 03	2.62E − 02	3.83E − 03	7.38E − 02
MOCBOA Cone-epsilon dominance	Worst	1.87E − 02	3.26E − 01	8.37E − 01	6.91E − 02	4.39E − 01	8.65E − 02	4.57E − 01
	Best	1.57E − 02	2.86E − 01	7.67E − 01	5.26E − 02	3.34E − 01	7.05E − 02	2.72E − 01
	Average	1.71E − 02	3.09E − 01	8.09E − 01	6.24E − 02	3.98E − 01	8.07E − 02	3.69E − 01
	Median	1.70E − 02	3.10E − 01	8.12E − 01	6.39E − 02	3.99E − 01	8.25E − 02	3.60E − 01
	Std	1.01E − 03	1.35E − 02	2.47E − 02	5.79E − 03	3.10E − 02	5.66E − 03	5.72E − 02
MOCBOA Strengthened dominance	Worst	1.95E − 02	3.35E − 01	8.26E − 01	6.75E − 02	4.59E − 01	8.64E − 02	5.78E − 01
	Best	1.55E − 02	2.98E − 01	7.53E − 01	5.09E − 02	3.48E − 01	7.07E − 02	3.57E − 01
	Average	1.73E − 02	3.22E − 01	7.86E − 01	6.00E − 02	4.07E − 01	8.04E − 02	5.04E − 01
	Median	1.73E − 02	3.26E − 01	7.83E − 01	6.01E − 02	4.14E − 01	8.10E − 02	5.47E − 01
	Std	1.09E − 03	1.24E − 02	2.39E − 02	4.58E − 03	3.33E − 02	4.39E − 03	8.27E − 02
		R15	**R16**	**R17**	**R18**	**R19**	**R20**	**R21**
MOCBOA Pareto dominance	Worst	7.03E − 01	6.45E − 01	2.89E − 01	3.56E − 02	6.58E − 01	2.13E − 01	2.69E − 02
	Best	1.85E − 01	1.26E − 01	3.39E − 02	3.15E − 02	6.26E − 01	1.73E − 01	2.43E − 02
	Average	5.45E − 01	4.97E − 01	1.48E − 01	3.38E − 02	6.42E − 01	1.86E − 01	2.55E − 02
	Median	6.05E − 01	5.37E − 01	1.47E − 01	3.41E − 02	6.40E − 01	1.74E − 01	2.54E − 02
	Std	1.63E − 01	1.57E − 01	8.20E − 02	1.34E − 03	9.52E − 03	1.74E − 02	7.69E − 04
MOCBOA Epsilon dominance	Worst	7.36E − 01	6.98E − 01	2.56E − 01	3.80E − 02	6.46E − 01	6.29E − 01	2.76E − 02
	Best	1.81E − 01	5.06E − 01	4.26E − 02	3.09E − 02	5.63E − 01	1.71E − 01	2.47E − 02
	Average	5.30E − 01	6.11E − 01	1.40E − 01	3.60E − 02	6.13E − 01	2.32E − 01	2.59E − 02
	Median	5.81E − 01	6.20E − 01	1.69E − 01	3.66E − 02	6.14E − 01	1.74E − 01	2.59E − 02
	Std	1.98E − 01	6.67E − 02	6.88E − 02	2.29E − 03	2.52E − 02	1.42E − 01	8.49E − 04
MOCBOA Cone-epsilon dominance	Worst	7.39E − 01	6.35E − 01	2.60E − 01	3.40E − 02	5.74E − 01	2.32E − 01	2.70E − 02
	Best	3.06E − 01	3.29E − 01	2.68E − 02	2.72E − 02	4.69E − 01	1.71E − 01	2.51E − 02
	Average	5.18E − 01	5.38E − 01	1.04E − 01	3.16E − 02	5.13E − 01	1.89E − 01	2.61E − 02
	Median	4.84E − 01	5.48E − 01	9.74E − 02	3.20E − 02	5.16E − 01	1.75E − 01	2.62E − 02
	Std	1.41E − 01	8.37E − 02	7.62E − 02	2.34E − 03	3.86E − 02	2.20E − 02	6.54E − 04
MOCBOA Strengthened dominance	Worst	7.65E − 01	6.56E − 01	4.41E − 01	3.63E − 02	6.59E − 01	1.00E + 00	2.89E − 02
	Best	2.21E − 01	4.34E − 01	2.85E − 02	2.77E − 02	6.30E − 01	1.74E − 01	2.51E − 02
	Average	4.76E − 01	5.58E − 01	2.12E − 01	3.31E − 02	6.42E − 01	7.58E − 01	2.67E − 02
	Median	4.96E − 01	5.85E − 01	2.00E − 01	3.35E − 02	6.42E − 01	1.00E + 00	2.68E − 02
	Std	1.87E − 01	8.02E − 02	1.50E − 01	2.57E − 03	9.31E − 03	3.90E − 01	1.19E − 03

Starting with Pareto dominance, we observe that it consistently achieves respectable median values across the benchmarks, suggesting steady performance. However, it struggles with variability, as indicated by relatively high standard deviations in certain cases, such as in R2 and R7. This variability hints at potential inconsistencies when tackling more intricate problem landscapes. Overall, Pareto dominance demonstrates a solid performance, yet it might be less dependable under more complex conditions. Epsilon dominance, on the other hand, shows stability, particularly with narrow standard deviations in many of the results.

For example, its performance on R8 and R16 remains within a tight range, illustrating its ability to handle complex benchmarks consistently. However, its "worst" performance values are generally slightly higher than those observed with Pareto dominance, suggesting that while it maintains reliability, it may sometimes miss achieving the most optimal solutions, possibly due to its tolerance parameter, which affects precision. Moving to Cone-epsilon dominance, this technique seems adept at balancing solution precision and consistency, particularly on benchmarks R9 and R13. It often yields lower best values across several benchmarks, indicating its ability to reach closer-to-optimal solutions in these cases. However, a slight trade-off is apparent, as its median values do not consistently outperform the other techniques. This pattern suggests that while Cone-epsilon is effective at handling specific instances with precision, it may be outperformed in terms of general robustness across varying problem types. Lastly, Strengthened dominance showcases impressive performance on some challenging benchmarks, such as R17 and R18. This technique is especially notable for occasionally achieving exceptionally low "best" values. However, it exhibits considerable variability with broader ranges between best and worst results on some benchmarks. This variability implies that while Strengthened dominance can excel under the right conditions, it may not be as reliable in consistently producing optimal solutions across the board.

The results show that each dominance technique demonstrates unique strengths and weaknesses in addressing the benchmarks. Pareto dominance is solid but somewhat variable, Epsilon dominance is stable yet less precise in finding optimal solutions, Cone-epsilon dominance achieves good precision but has limited general consistency, and Strengthened dominance excels with low best values yet faces variability challenges. These observations underline the importance of selecting the appropriate dominance technique based on specific optimization goals and problem characteristics.

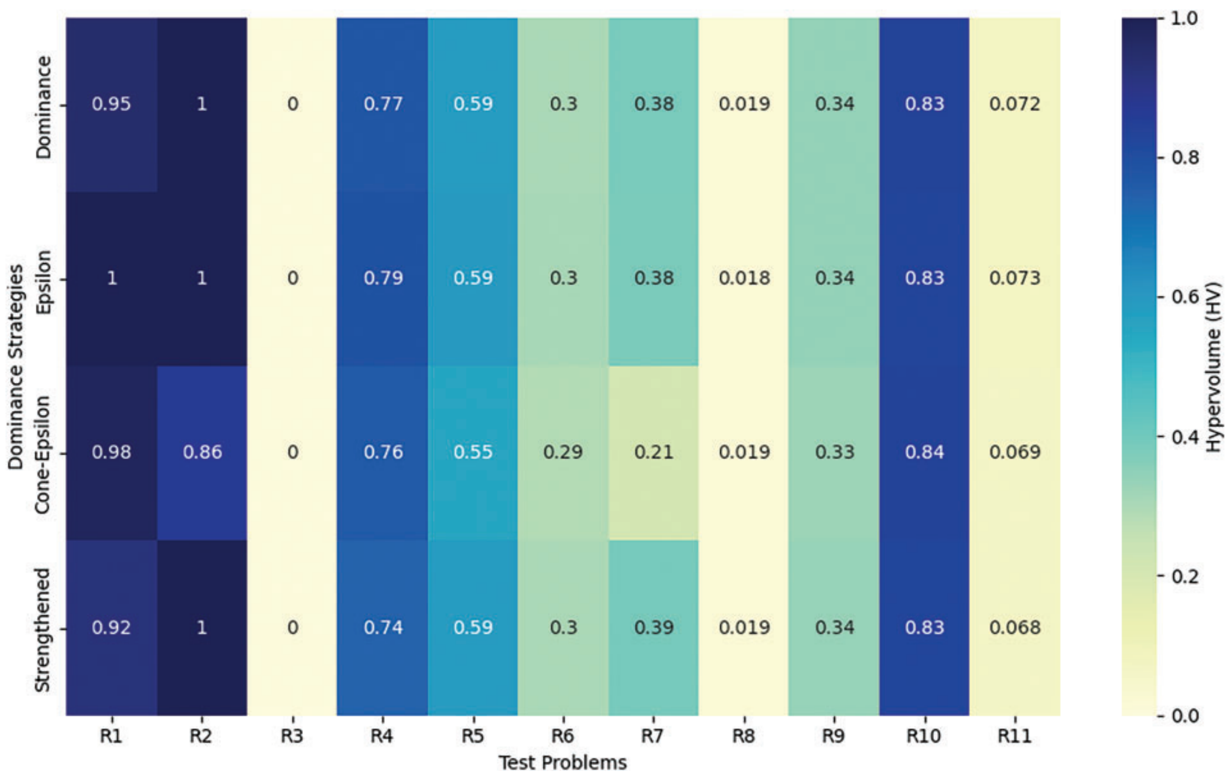

Figure 4: HV analysis across mechanical design problems (R1–R11)

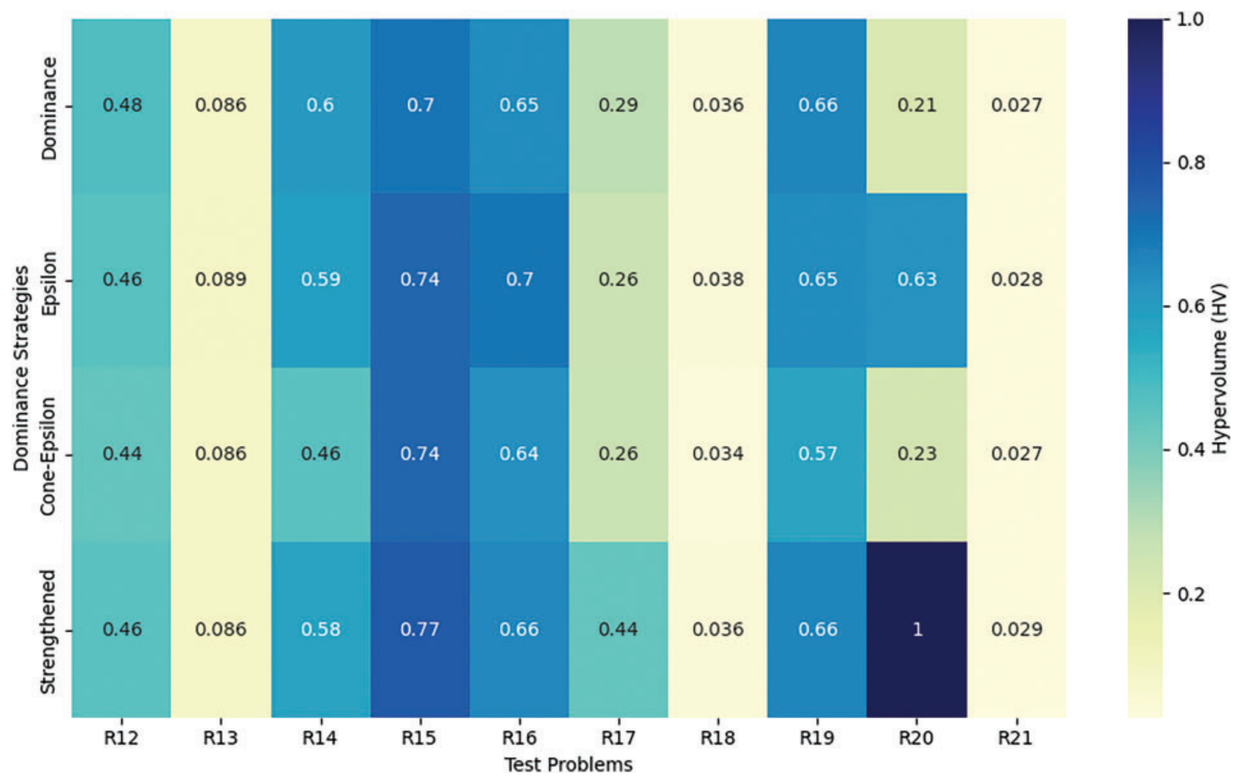

Figure 5: HV analysis across mechanical design problems (R12–R21)

5.2 Chemical Engineering Problems

The results from the MOCBOA algorithm in Table 4 demonstrate clear distinctions in the performance of different dominance approaches across the chemical engineering problems R22–R24. The Epsilon dominance approach consistently delivers stable and high-quality outcomes, showcasing its effectiveness in multi-objective optimization tasks. Its performance is characterized by minimal variability, making it a reliable method for solving complex problems. As shown in Fig. 6, this dominance method performs exceptionally well across all test cases, maintaining consistency even in more challenging scenarios. This consistency underscores the Epsilon dominance approach as a dependable choice when stability is a key requirement in multi-objective optimization.

Table 4: Results for the chemical engineering problems R22–R24

		R22	R23	R24
MOCBOA Pareto dominance	Worst	1.00E + 00	8.68E − 01	1.00E + 00
	Best	9.37E − 01	5.67E − 01	0.00E + 00
	Average	9.90E − 01	7.56E − 01	1.36E − 01
	Median	1.00E + 00	7.76E − 01	0.00E + 00
	Std	2.00E − 02	9.20E − 02	3.24E − 01

(Continued)

Table 4 (continued)

		R22	R23	R24
MOCBOA Epsilon dominance	Worst	1.00E + 00	9.72E − 01	1.00E + 00
	Best	9.82E − 01	5.22E − 01	0.00E + 00
	Average	9.95E − 01	6.95E − 01	8.00E − 01
	Median	9.98E − 01	6.75E − 01	1.00E + 00
	Std	6.39E − 03	1.41E − 01	4.22E − 01
MOCBOA Cone-epsilon dominance	Worst	8.66E − 01	6.23E − 01	0.00E + 00
	Best	7.19E − 01	3.12E − 01	0.00E + 00
	Average	8.12E − 01	4.28E − 01	0.00E + 00
	Median	8.13E − 01	4.20E − 01	0.00E + 00
	Std	4.38E − 02	9.34E − 02	0.00E + 00
MOCBOA Strengthened dominance	Worst	1.00E + 00	9.99E − 01	2.02E + 04
	Best	6.92E − 01	5.53E − 01	0.00E + 00
	Average	9.49E − 01	7.80E − 01	2.02E + 03
	Median	9.99E − 01	7.77E − 01	0.00E + 00
	Std	9.72E − 02	1.27E − 01	6.39E + 03

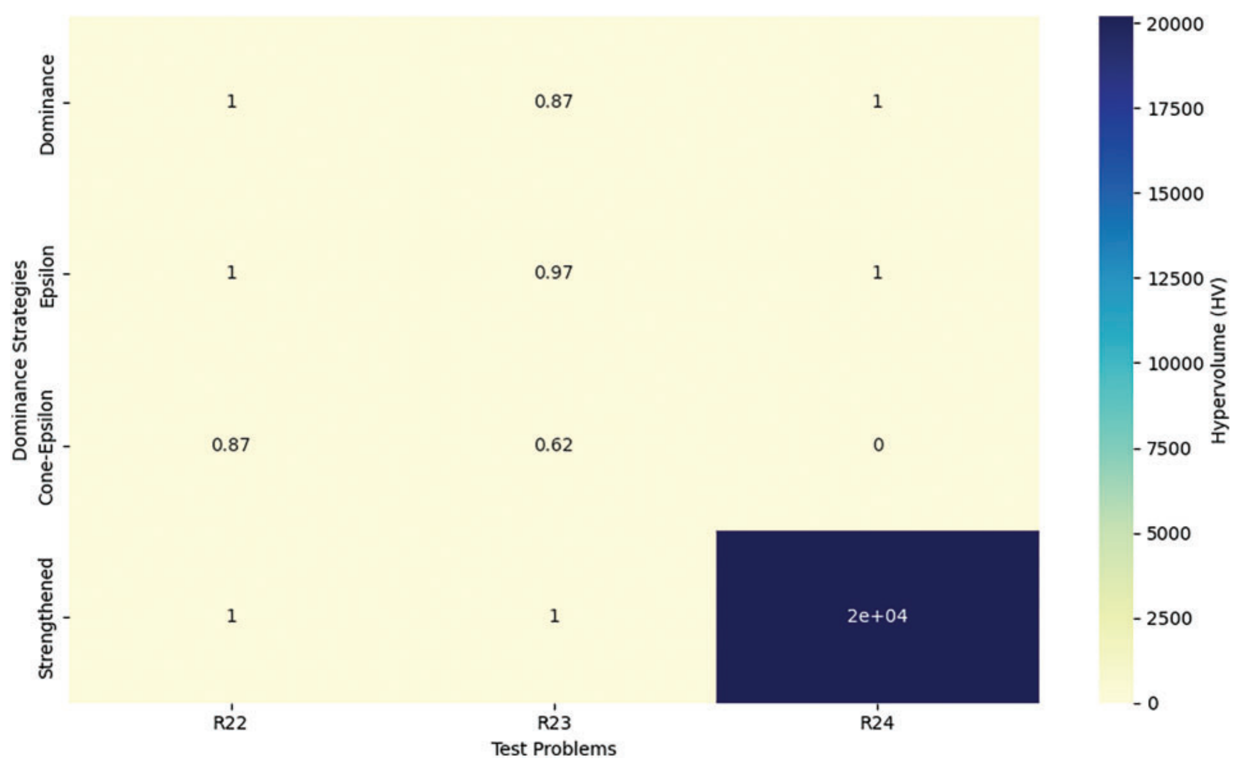

Figure 6: HV analysis across chemical engineering problems (R22–R24)

In contrast, the Dominance approach performs reasonably well in simpler problems but struggles significantly when applied to more complex situations. The results reveal noticeable inconsistencies and variability, particularly in more intricate problems, which make this approach less stable and reliable. These issues suggest that while the Dominance approach may work effectively for straightforward cases, its performance degrades when faced with the added complexity of real-world multi-objective challenges. The variability seen in the results further emphasizes the need for a more robust approach when dealing with difficult problems, making the Dominance approach less suitable for such tasks.

The Cone-epsilon dominance method presents a more nuanced performance. While it shows promising results in certain cases, it also experiences significant variability in others, indicating a high sensitivity to the nature of the problem. This inconsistent behavior suggests that while Cone-epsilon dominance can be effective for some problem types, its performance may not be dependable across the full spectrum of multi-objective optimization problems. Similarly, the Strengthened dominance method demonstrates both strengths and weaknesses. It performs well in some instances but exhibits considerable variability in others, particularly when tackling more challenging problems. According to the obtained results, this approach struggles with stability, particularly in cases where the problem complexity is higher, further emphasizing its unreliability in more difficult scenarios. These findings suggest that the Epsilon dominance approach remains the most stable and reliable choice, while the others may require further refinement or may only be suitable for less complex tasks.

5.3 Process, Design and Synthesis Problems

The results of the MOCBOA algorithm for the process design and synthesis problems (R25–R29) are summarized in Table 5. For the Pareto dominance approach, the results exhibit good consistency in most cases, especially for problems R25, R26, and R29, where the worst, best, and average performance measures show relatively tight ranges. This suggests that Pareto dominance can provide solid solutions in these problems, though there is some variability, particularly with problem R27, which demonstrates a high degree of difficulty as evidenced by the very large worst-case value. This indicates that while Pareto dominance can perform reliably in simpler scenarios, it struggles significantly with more complex problem sets, highlighting its limitations in handling the entire spectrum of multi-objective optimization problems.

Table 5: Results for the process, design and synthesis problems R25–R29

		R25	**R26**	**R27**	**R28**	**R29**
MOCBOA Pareto dominance	Worst	1.76E − 01	6.62E − 02	1.40E + 04	1.00E + 00	1.00E + 00
	Best	1.38E − 01	5.15E − 02	1.00E + 00	1.00E + 00	9.55E − 01
	Average	1.54E − 01	5.98E − 02	2.73E + 03	1.00E + 00	9.92E − 01
	Median	1.53E − 01	6.09E − 02	2.18E + 01	1.00E + 00	9.99E − 01
	Std	1.10E − 02	5.16E − 03	5.55E + 03	0.00E + 00	1.53E − 02
MOCBOA Epsilon dominance	Worst	1.00E + 00	8.75E − 01	2.06E + 00	1.00E + 00	1.00E + 00
	Best	8.64E − 01	7.33E − 01	1.65E + 00	1.00E + 00	0.00E + 00
	Average	9.68E − 01	8.08E − 01	1.86E + 00	1.00E + 00	8.65E − 01
	Median	9.74E − 01	8.18E − 01	1.82E + 00	1.00E + 00	9.93E − 01
	Std	3.86E − 02	4.64E − 02	1.17E − 01	0.00E + 00	3.14E − 01

(Continued)

Table 5 (continued)

		R25	R26	R27	R28	R29
MOCBOA Cone-epsilon dominance	Worst	1.87E − 01	6.99E − 02	1.02E + 01	4.88E − 01	0.00E + 00
	Best	1.38E − 01	4.38E − 02	1.00E + 00	6.44E − 02	0.00E + 00
	Average	1.54E − 01	5.75E − 02	3.09E + 00	2.31E − 01	0.00E + 00
	Median	1.49E − 01	5.78E − 02	1.86E + 00	2.58E − 01	0.00E + 00
	Std	1.59E − 02	7.61E − 03	3.08E + 00	1.27E − 01	0.00E + 00
MOCBOA Strengthened dominance	Worst	4.76E + 00	7.10E − 02	1.34E + 00	1.00E + 00	1.00E + 00
	Best	0.00E + 00	0.00E + 00	1.00E + 00	1.00E + 00	9.65E − 01
	Average	9.70E − 01	5.60E − 02	1.04E + 00	1.00E + 00	9.94E − 01
	Median	4.76E − 02	6.24E − 02	1.00E + 00	1.00E + 00	9.98E − 01
	Std	1.95E + 00	2.06E − 02	1.07E − 01	0.00E + 00	1.06E − 02

The Epsilon dominance approach, on the other hand, shows strong performance in most problems, with notably high average and median values for R25, R26, and R28. The approach excels particularly in its ability to handle the worst-case scenarios, providing a good balance of consistency and quality. However, problems like R27 and R29, especially with the latter having lower performance on average, demonstrate that Epsilon dominance can sometimes struggle with certain problem complexities, leading to a higher standard deviation. This points to a method that is highly effective in many cases but still sensitive to the specific nature of the problem, requiring further refinement or adjustments to handle outliers effectively.

The Cone-epsilon dominance approach exhibits a mixed performance across the problem set, with some issues like R27 and R29 showing significant variability and poor results. The worst-case values for R27 in particular are quite high, signaling that this approach may not be the best choice for all types of optimization problems. The results reveal that while Cone-epsilon dominance can be effective in simpler cases like R25 and R26, it struggles significantly in more complicated scenarios, as seen in R27. Similarly, the Strengthened dominance approach presents a contrasting performance, with some very large worst-case values for problems like R25 and R27, but it also provides some strong results, particularly for R28 and R29. This shows that the Strengthened dominance method can excel in some areas but has considerable instability, especially for more complex problems. As illustrated in Fig. 7, these variations in the performance across different dominance methods reinforce the need for a careful selection of the right approach depending on the problem's characteristics and complexity.

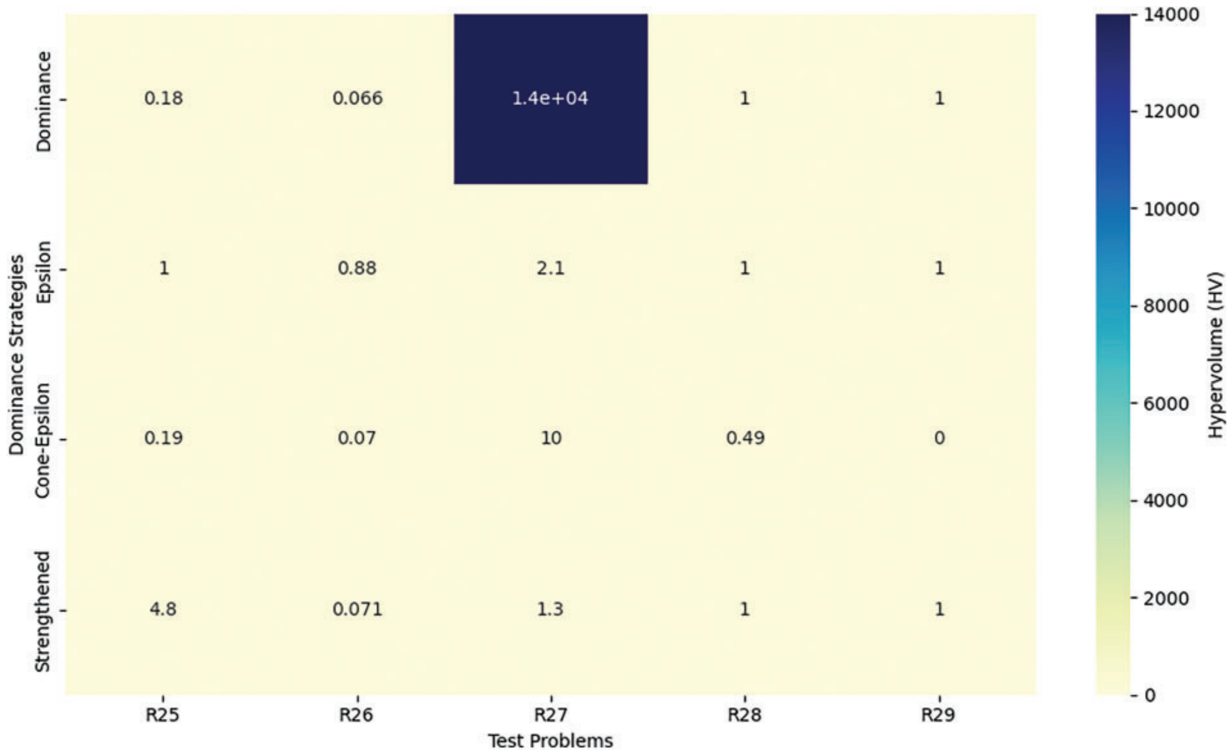

Figure 7: HV analysis across process, design and synthesis problems (R25–R29)

5.4 Power Electronics Problems

The results of the MOCBOA algorithm applied to power electronics problems (R30–R35) in Table 6 reveal interesting patterns across the various dominance approaches. The obtained results in this section can be discussed as follows:

Table 6: Results for the power electronics problems R30–R35

		R30	R31	R32	R33	R34	R35
MOCBOA Pareto dominance	Worst	9.09E − 02	6.60E − 01	4.85E − 01	1.68E − 01	5.36E − 01	9.94E − 01
	Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	4.68E − 01
	Average	9.09E − 03	3.74E − 01	1.67E − 01	3.96E − 02	1.87E − 01	7.36E − 01
	Median	0.00E + 00	4.05E − 01	1.68E − 01	0.00E + 00	1.51E − 01	7.49E − 01
	Std	2.87E − 02	2.21E − 01	1.64E − 01	6.64E − 02	2.04E − 01	1.99E − 01
MOCBOA Epsilon dominance	Worst	2.68E − 01	7.09E − 01	4.89E − 01	5.00E − 01	4.49E − 01	8.05E − 01
	Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	3.89E − 01
	Average	4.42E − 02	3.70E − 01	1.09E − 01	8.80E − 02	1.04E − 01	6.17E − 01
	Median	0.00E + 00	3.93E − 01	0.00E + 00	0.00E + 00	0.00E + 00	6.01E − 01
	Std	9.58E − 02	2.56E − 01	1.72E − 01	1.66E − 01	1.81E − 01	1.61E − 01
MOCBOA Cone-epsilon dominance	Worst	0.00E + 00	8.59E − 03	0.00E + 00	0.00E + 00	0.00E + 00	5.87E − 01
	Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	3.30E − 01
	Average	0.00E + 00	8.59E − 04	0.00E + 00	0.00E + 00	0.00E + 00	4.56E − 01
	Median	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	4.71E − 01
	Std	0.00E + 00	2.72E − 03	0.00E + 00	0.00E + 00	0.00E + 00	7.70E − 02

(Continued)

Table 6 (continued)

		R30	R31	R32	R33	R34	R35
MOCBOA Strengthened dominance	Worst	9.09E − 02	5.82E − 01	4.89E − 01	4.91E − 01	6.46E − 01	9.92E − 01
	Best	0.00E + 00	9.09E − 02	0.00E + 00	0.00E + 00	0.00E + 00	4.23E − 01
	Average	2.64E − 02	4.09E − 01	1.67E − 01	1.26E − 01	1.34E − 01	7.58E − 01
	Median	0.00E + 00	4.55E − 01	1.25E − 01	4.56E − 02	0.00E + 00	8.04E − 01
	Std	4.26E − 02	1.57E − 01	1.73E − 01	1.85E − 01	2.29E − 01	1.83E − 01

When examining the Pareto dominance results, we see a considerable variation in the performance across the different problems. The worst-case performance for R31 and R35 stands out, with values much higher than for other problems, especially in the case of R35, where the worst-case performance reaches nearly 1. This suggests that Pareto's dominance faces significant challenges when dealing with these specific power electronics problems, likely due to their complexity. On the other hand, the best-case results for problems like R30, R32, and R34 show promising outcomes, indicating that the algorithm can sometimes find near-optimal solutions under favorable conditions. The median and average values further illustrate a consistent ability to find good solutions, with some issues (R33, R35) showing high average performance despite the large worst-case values.

The Epsilon dominance approach generally follows a similar trend, where problems like R30 and R35 exhibit high worst-case values, indicating challenges in finding optimal solutions. However, this approach performs quite well in terms of average performance, with the average values across most problems (e.g., R30, R31, R33) showing promising results. Notably, the standard deviations are generally higher compared to the Pareto dominance approach, especially for problems R30 and R31, suggesting that Epsilon dominance is more sensitive to problem complexities and may struggle with certain problem instances. Nevertheless, it still demonstrates the potential for high-quality solutions in many cases, particularly for problems where the best values are zero, indicating that the algorithm can achieve optimal or near-optimal results.

The Cone-epsilon dominance approach, which generally shows lower worst-case values, particularly for problems R30 to R34, tends to offer more stability. However, the average and median values for many of the problems (such as R30, R32, and R34) are consistently zero, which indicates that the approach is unable to generate diverse solutions and may fail to address the full range of objectives in these problems. For more complex problems like R35, Cone-epsilon dominance performs relatively better, although the performance still falls short compared to other approaches.

Lastly, the Strengthened dominance approach presents mixed results. While the worst-case values are high for most problems (particularly for R30, R31, and R35), the best-case results indicate that Strengthened dominance can find optimal solutions under certain conditions. This method tends to generate solid performance in terms of average values, though the variability is high, as evidenced by the large standard deviations for most problems.

Overall, while each dominance approach shows its strengths, the results indicate that selecting the most appropriate method depends heavily on the specific characteristics of the power electronics problems at hand. The HV analysis, as shown in Fig. 8, further highlights the differences in the performance across these approaches, suggesting that a hybrid or adaptive strategy might be beneficial to optimize the results for complex problem sets.

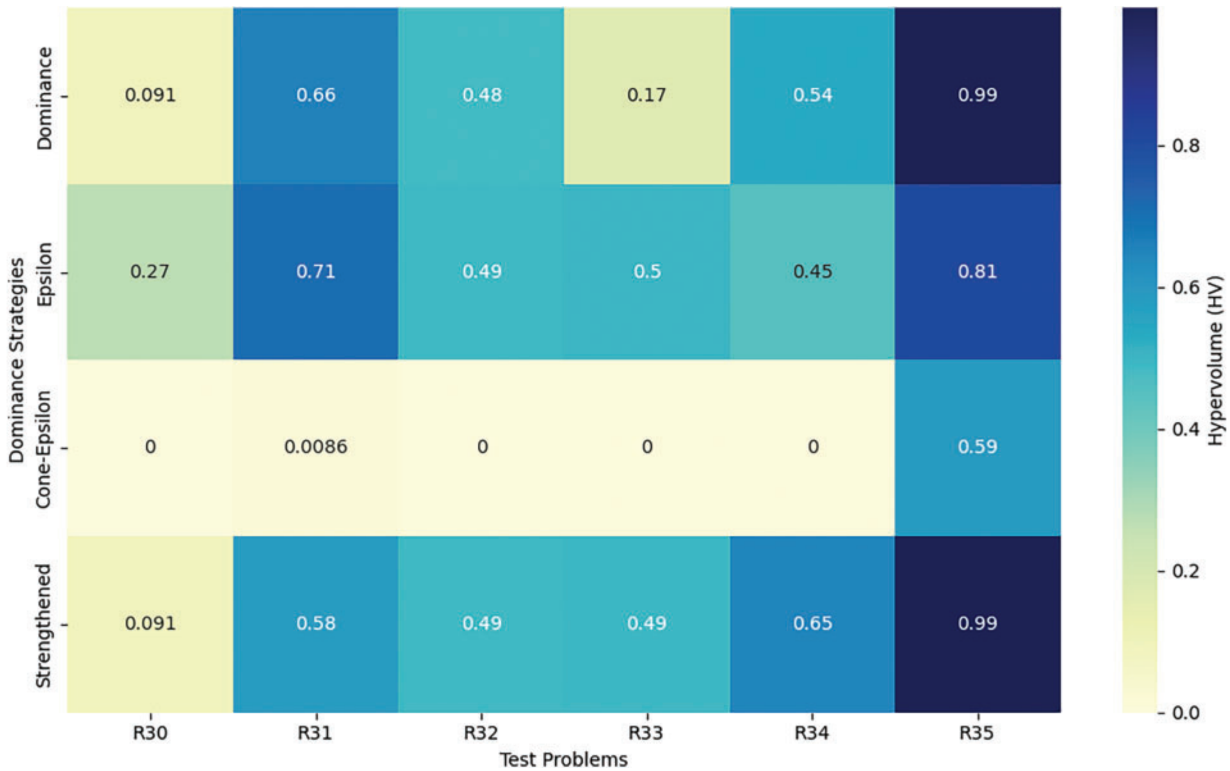

Figure 8: HV analysis across power electronics problems (R30–R35)

5.5 Power System Problems

In this section, we present an analysis and discussion of the results from Table 7, which shows the performance of the MOCBOA algorithm across different power system problems (R36–R50). The table includes metrics based on four dominance techniques: Pareto, Epsilon, Cone-epsilon, and Strengthened dominance. To visualize and further interpret these results, Figs. 9 and 10 depict the HV analysis across the problems R36–R43 and R44–R50, respectively.

Table 7: Results for the power system problems R36–R50

		R36	R37	R38	R39	R40
MOCBOA Pareto dominance	Worst	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	6.13E − 01
	Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	1.29E − 01
	Average	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	4.75E − 01
	Median	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	5.32E − 01
	Std	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	1.60E − 01
MOCBOA Epsilon dominance	Worst	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	5.66E − 01
	Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	2.34E − 01
	Average	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	3.86E − 01
	Median	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	3.78E − 01
	Std	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	8.60E − 02

(Continued)

Table 7 (continued)

		R36	R37	R38	R39	R40
MOCBOA Cone-epsilon dominance	Worst	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Average	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Median	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
MOCBOA Strengthened dominance	Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	4.67E − 01
	Worst	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	4.95E − 03
	Average	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	2.49E − 01
	Median	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	2.31E − 01
	Std	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	1.38E − 01
		R41	**R42**	**R43**	**R44**	**R45**
MOCBOA Pareto dominance	Worst	0.00E + 00	0.00E + 00	9.01E − 01	1.43E − 01	0.00E + 00
	Best	0.00E + 00	0.00E + 00	4.69E − 01	0.00E + 00	0.00E + 00
	Average	0.00E + 00	0.00E + 00	7.33E − 01	2.21E − 02	0.00E + 00
	Median	0.00E + 00	0.00E + 00	7.77E − 01	0.00E + 00	0.00E + 00
	Std	0.00E + 00	0.00E + 00	1.69E − 01	4.63E − 02	0.00E + 00
MOCBOA Epsilon dominance	Worst	0.00E + 00	0.00E + 00	9.33E − 01	0.00E + 00	0.00E + 00
	Best	0.00E + 00	0.00E + 00	5.74E − 01	0.00E + 00	0.00E + 00
	Average	0.00E + 00	0.00E + 00	8.01E − 01	0.00E + 00	0.00E + 00
	Median	0.00E + 00	0.00E + 00	8.19E − 01	0.00E + 00	0.00E + 00
	Std	0.00E + 00	0.00E + 00	1.22E − 01	0.00E + 00	0.00E + 00
MOCBOA Cone-epsilon dominance	Worst	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Average	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Median	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
MOCBOA Strengthened dominance	Worst	3.07E + 01	0.00E + 00	9.88E − 01	2.07E − 01	0.00E + 00
	Best	0.00E + 00	0.00E + 00	7.81E − 01	0.00E + 00	0.00E + 00
	Average	3.07E + 00	0.00E + 00	8.90E − 01	3.28E − 02	0.00E + 00
	Median	0.00E + 00	0.00E + 00	8.92E − 01	0.00E + 00	0.00E + 00
	Std	9.72E + 00	0.00E + 00	7.62E − 02	6.61E − 02	0.00E + 00
		R46	**R47**	**R48**	**R49**	**R50**
MOCBOA Pareto dominance	Worst	0.00E + 00	1.00E + 00	7.60E − 01	0.00E + 00	6.11E − 01
	Best	0.00E + 00	1.00E + 00	0.00E + 00	0.00E + 00	6.09E − 01
	Average	0.00E + 00	1.00E + 00	2.50E − 01	0.00E + 00	6.11E − 01
	Median	0.00E + 00	1.00E + 00	8.41E − 02	0.00E + 00	6.11E − 01
	Std	0.00E + 00	0.00E + 00	3.20E − 01	0.00E + 00	5.85E − 04

(Continued)

Table 7 (continued)

		R36	R37	R38	R39	R40
MOCBOA Epsilon dominance	Worst	0.00E + 00	1.00E + 00	5.01E − 01	0.00E + 00	6.11E − 01
	Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	6.11E − 01
	Average	0.00E + 00	4.65E − 01	1.31E − 01	0.00E + 00	6.11E − 01
	Median	0.00E + 00	4.23E − 01	0.00E + 00	0.00E + 00	6.11E − 01
	Std	0.00E + 00	4.02E − 01	2.10E − 01	0.00E + 00	1.17E − 16
MOCBOA Cone-epsilon dominance	Worst	0.00E + 00	8.08E − 01	0.00E + 00	0.00E + 00	4.71E − 01
	Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	3.37E − 01
	Average	0.00E + 00	2.72E − 01	0.00E + 00	0.00E + 00	4.07E − 01
	Median	0.00E + 00	2.83E − 01	0.00E + 00	0.00E + 00	4.05E − 01
	Std	0.00E + 00	2.61E − 01	0.00E + 00	0.00E + 00	4.26E − 02
MOCBOA Strengthened dominance	Worst	0.00E + 00	1.00E + 00	0.00E + 00	0.00E + 00	6.11E − 01
	Best	0.00E + 00	1.00E + 00	0.00E + 00	0.00E + 00	6.09E − 01
	Average	0.00E + 00	1.00E + 00	0.00E + 00	0.00E + 00	6.11E − 01
	Median	0.00E + 00	1.00E + 00	0.00E + 00	0.00E + 00	6.11E − 01
	Std	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	8.57E − 04

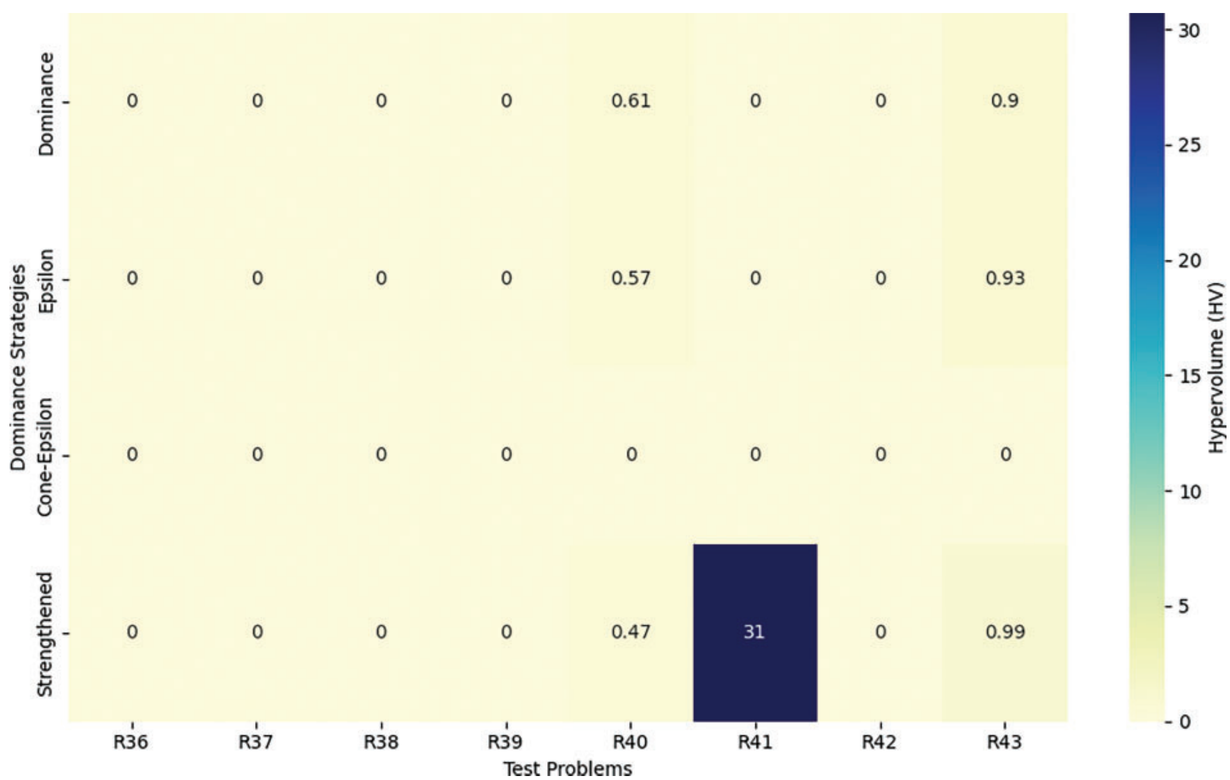

Figure 9: HV analysis across power systems problems (R36–R43)

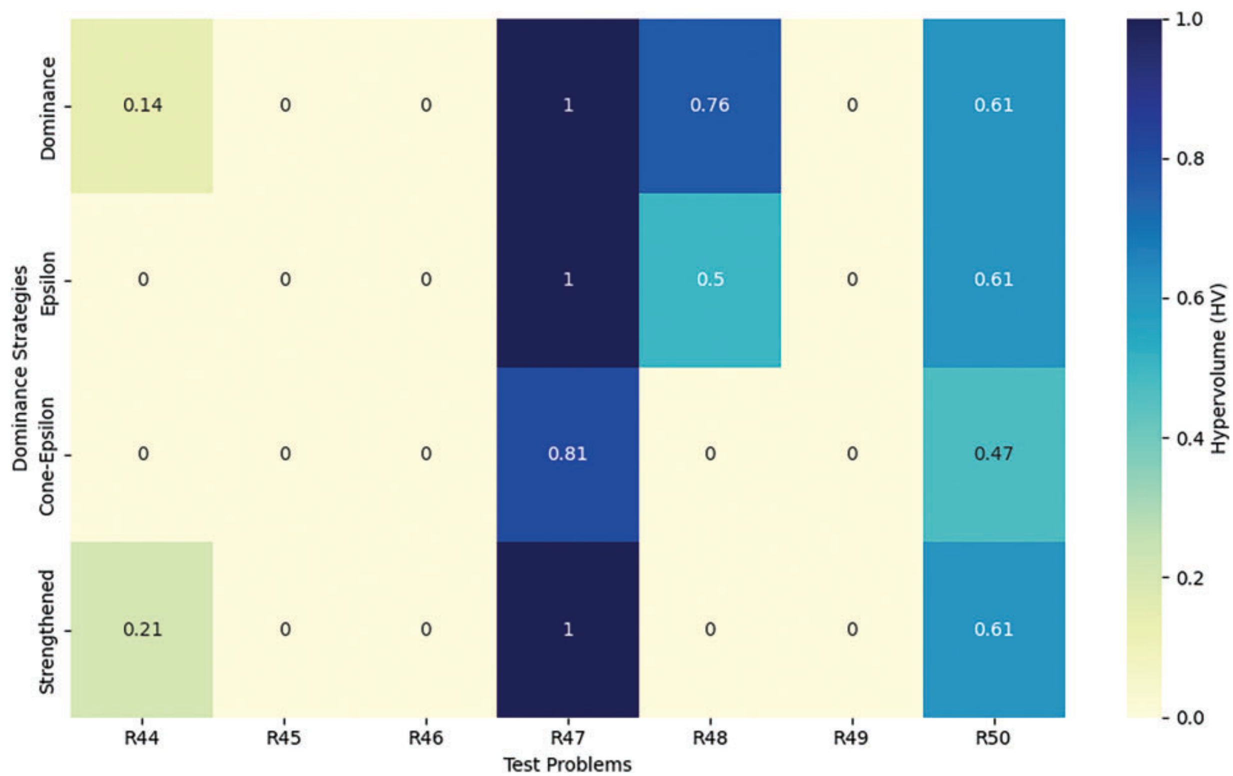

Figure 10: HV analysis across power systems problems (R44–R50)

From Table 7, it is clear that the MOCBOA algorithm consistently yields a "Worst" value of 0.00E + 00 across the majority of the problems under the Pareto and Epsilon dominance techniques, which indicates that the algorithm performs well in avoiding poor solutions. However, the "Best" values vary more noticeably, especially for problems R39–R42, where the results are non-zero, showing that the algorithm reaches better solutions in these cases. The "Average" values for all problems are generally low, confirming the robustness of the algorithm in maintaining a high-quality solution across different instances. The standard deviation (Std) remains very small, which further supports the consistency of MOCBOA across all problems.

In contrast, under the Cone-epsilon dominance technique, all the metrics for the worst, best, average, median, and standard deviation are zero, indicating that this technique may not have contributed positively to the solution diversity or quality in this particular case. The Strengthened dominance technique, however, shows more variation in the results. For the problems R41–R43, the "Worst" value appears to be considerably high (e.g., 3.07E + 01 for R41), but it still manages to show a "Best" solution around 7.81E − 01 for R43, which suggests that the Strengthened dominance technique might handle specific power system challenges more effectively.

Fig. 9 and 10 complement these findings by presenting the HV analysis for the power system problems R36–R43 and R44–R50, respectively. In Fig. 9, the HV values are generally high for most problems, suggesting that the MOCBOA algorithm performs well in both converging to optimal solutions and maintaining diversity in its solutions. Problems R40 and R42 exhibit slightly lower HV values, indicating a possible area for improvement. On the other hand, in Fig. 10, the HV values remain consistently high, with a noticeable peak in R45, showing that the algorithm achieves an optimal convergence in these problems. This analysis demonstrates that MOCBOA excels in solving complex multi-objective optimization problems in power systems, particularly for problems R44–R50, where the HV values indicate an effective balance between convergence and diversity.

Overall, the results indicate that MOCBOA's performance varies across different power system problems, with significant improvement in convergence and diversity for specific problem instances. The Pareto and Epsilon dominance techniques generally yield good results, while the Strengthened dominance method shows potential for specific problem sets.

5.6 Comparison

Table 8 describes standard parameter settings for each of the algorithms utilized in this experiment: MOGSK, MOEA/D, NSGAII, and SPEA2. Adopting these generally accepted parameters from the literature assures a fair and unbiased comparison by reducing the potential impact of parameter adjustment on the performance outcomes. This method enables us to focus on the inherent strengths and shortcomings of each algorithm under consistent conditions.

Table 8: Adapted values for parameters of the comparative algorithms

Algorithm	Parameters
MOGSK	Knowledge rate (K_r): 0.5 Knowledge ratio (K_f): 0.5 Dimensional factor (D_f): 0.5 Gaining coefficient (G_c): Controls knowledge gain, [1, 2] Sharing coefficient (S_c): Controls knowledge sharing [1, 2] Learning factor (α): Defines learning rate 1.5–2 Mutation factor (β): Controls mutation strength, 0.1–0.5
MOEA/D	Crossover probability (Pc): 1.0 Mutation probability (Pm): $\frac{1}{n}$ Neighborhood size (T): 20 SBX and polynomial mutation: η_c = 20, η_m = 20 Update limit: 2
NSGAII	Crossover probability (Pc): 0.9 Mutation probability (Pm): $\frac{1}{n}$ (where n is the number of decision variables) Crossover operator: Simulated Binary Crossover (SBX) with η_c = 20 Mutation operator: Polynomial Mutation with η_m = 20
SPEA2	Crossover probability (Pc): 0.9. Mutation probability (Pm): $\frac{1}{n}$ SBX and polynomial mutation: η_c = 20, η_m = 20

Table 9 presents a comprehensive performance comparison between the proposed MOCBOA algorithm and the existing MOGSK algorithm [14], NSGAII [5], MOEA/D [7], and SPEA2 [6], across 50 test problems (R1–R50), using Hyper volume as the evaluation metric. Fig. 11 provides a visual comparison of the average performance across all test problems. We note that the PlatEmov [67] was used to generate the results for the algorithms NSGAII, MOEA/D, and SPEA2.

Table 9: Performance comparison between the MOCBOA, MOGSK, MOEA/D, NSGAII and SPEA2 problems R1–R50

		R1	R2	R3	R4	R5	R6	R7
MOCBOA	Best	9.92E − 01	7.40E − 01	0.00E + 00	6.24E − 01	4.89E − 01	2.78E − 01	0.00E + 00
	Average	9.98E − 01	9.04E − 01	0.00E + 00	7.31E − 01	5.71E − 01	2.97E − 01	1.37E − 01
	Std	2.16E − 03	9.19E − 02	0.00E + 00	5.51E − 02	3.51E − 02	8.32E − 03	1.30E − 01
MOGSK [14]	Best	1.00E + 00	3.80E − 01	0.00E + 00	8.51E − 01	6.28E − 01	3.19E − 01	4.63E − 01
	Average	1.00E + 00	5.08E − 01	0.00E + 00	8.63E − 01	6.29E − 01	3.19E − 01	4.75E − 01
	Std	3.96E − 06	9.14E − 02	0.00E + 00	5.35E − 03	1.55E − 04	1.73E − 04	4.61E − 03
MOEAD [7]	Best	NaN	NaN	NaN	NaN	NaN	NaN	4.79E − 01
	Average	NaN	NaN	NaN	NaN	NaN	NaN	4.77E − 01
	Std	NaN	NaN	NaN	NaN	NaN	NaN	2.16E − 03
NSGAII [5]	Best	5.96E − 01	3.93E − 01	9.02E − 01	8.62E − 01	4.34E − 01	2.77E − 01	4.84E − 01
	Average	5.94E − 01	2.33E − 01	9.02E − 01	8.56E − 01	4.34E − 01	2.77E − 01	4.84E − 01
	Std	1.52E − 03	1.63E − 01	1.92E − 04	4.95E − 03	5.54E − 04	6.59E − 05	1.21E − 04
SPEA2 [6]	Best	NaN	NaN	NaN	NaN	2.77E − 01	NaN	4.83E − 01
	Average	NaN	NaN	NaN	NaN	2.74E − 01	NaN	4.82E − 01
	Std	NaN	NaN	NaN	NaN	2.15E − 03	NaN	3.32E − 04
		R8	**R9**	**R10**	**R11**	**R12**	**R13**	**R14**
MOCBOA	Best	1.57E − 02	2.96E − 01	7.59E − 01	6.61E − 02	3.92E − 01	7.78E − 02	4.03E − 01
	Average	1.68E − 02	3.19E − 01	8.12E − 01	6.92E − 02	4.22E − 01	8.42E − 02	5.10E − 01
	Std	9.79E − 04	1.66E − 02	2.13E − 02	1.98E − 03	2.62E − 02	3.83E − 03	7.38E − 02
MOGSK [14]	Best	2.33E − 02	4.01E − 01	8.35E − 01	8.05E − 02	5.54E − 01	9.66E − 02	7.13E − 01
	Average	2.48E − 02	4.02E − 01	8.40E − 01	8.52E − 02	5.60E − 01	9.71E − 02	7.13E − 01
	Std	7.06E − 04	9.85E − 04	2.93E − 03	3.04E − 03	4.06E − 03	3.68E − 04	2.34E − 04
MOEAD [7]	Best	NaN	5.31E − 02	8.01E − 02	6.53E − 02	NaN	NaN	NaN
	Average	NaN	5.31E − 02	7.93E − 02	5.86E − 02	NaN	NaN	NaN
	Std	NaN	4.14E − 05	6.41E − 04	2.57E − 03	NaN	NaN	NaN
NSGAII [5]	Best	2.59E − 02	4.09E − 01	8.47E − 01	9.65E − 02	5.57E − 01	8.79E − 02	6.15E − 01
	Average	2.58E − 02	4.09E − 01	8.47E − 01	9.44E − 02	5.55E − 01	8.75E − 02	6.14E − 01
	Std	5.98E − 05	1.77E − 04	6.13E − 05	1.21E − 03	1.15E − 03	1.70E − 04	9.48E − 04
SPEA2 [6]	Best	2.38E − 02	4.10E − 01	8.44E − 01	7.08E − 02	5.45E − 01	NaN	3.52E − 01
	Average	2.34E − 02	4.09E − 01	8.41E − 01	6.18E − 02	5.24E − 01	NaN	3.45E − 01
	Std	5.12E − 04	1.56E − 04	4.03E − 03	9.92E − 03	1.98E − 02	NaN	5.21E − 03
		R15	**R16**	**R17**	**R18**	**R19**	**R20**	**R21**
MOCBOA	Best	1.81E − 01	5.06E − 01	4.26E − 02	3.09E − 02	5.63E − 01	1.71E − 01	2.47E − 02
	Average	5.30E − 01	6.11E − 01	1.40E − 01	3.60E − 02	6.13E − 01	2.32E − 01	2.59E − 02
	Std	1.98E − 01	6.67E − 02	6.88E − 02	2.29E − 03	2.52E − 02	1.42E − 01	8.49E − 04
MOGSK [14]	Best	7.82E − 01	7.23E − 01	2.51E − 01	4.10E − 02	6.63E − 01	1.74E − 01	2.79E − 02
	Average	7.84E − 01	7.31E − 01	3.46E − 01	4.10E − 02	6.63E − 01	1.74E − 01	2.88E − 02
	Std	9.11E − 04	6.54E − 03	8.57E − 02	6.19E − 05	8.28E − 17	1.60E − 11	4.66E − 04
MOEAD [7]	Best	NaN	NaN	1.25E − 01	4.02E − 02	NaN	NaN	2.93E − 02
	Average	NaN	NaN	NaN	4.02E − 02	NaN	NaN	2.93E − 02
	Std	NaN	NaN	NaN	2.58E − 05	NaN	NaN	4.09E − 06
NSGAII [5]	Best	5.40E − 01	7.62E − 01	2.71E − 01	4.04E − 02	3.45E − 01	0.00E + 00	3.17E − 02
	Average	5.39E − 01	7.61E − 01	2.64E − 01	4.04E − 02	3.26E − 01	0.00E + 00	3.17E − 02
	Std	7.33E − 04	5.74E − 04	6.57E − 03	1.93E − 05	2.34E − 02	0.00E + 00	3.99E − 06
SPEA2 [6]	Best	NaN	7.59E − 01	NaN	4.03E − 02	NaN	NaN	3.17E − 02
	Average	NaN	7.58E − 01	NaN	4.01E − 02	NaN	NaN	3.17E − 02
	Std	NaN	8.39E − 04	NaN	1.44E − 04	NaN	NaN	2.56E − 06
		R22	**R23**	**R24**	**R25**	**R26**	**R27**	**R28**
MOCBOA	Best	9.82E − 01	5.22E − 01	0.00E + 00	8.64E − 01	7.33E − 01	1.65E + 00	1.00E + 00
	Average	9.95E − 01	6.95E − 01	8.00E − 01	9.68E − 01	8.08E − 01	1.86E + 00	1.00E + 00
	Std	6.39E − 03	1.41E − 01	4.22E − 01	3.86E − 02	4.64E − 02	1.17E − 01	0.00E + 00
MOGSK [14]	Best	1.00E + 00	9.99E − 01	1.00E + 00	9.97E − 01	8.28E − 01	1.48E + 00	1.00E + 00
	Average	1.00E + 00	9.99E − 01	1.00E + 00	9.99E − 01	8.40E − 01	1.56E + 00	1.00E + 00

(Continued)

Table 9 (continued)

		R1	R2	R3	R4	R5	R6	R7	
	Std	0.00E + 00	1.57E − 16	3.00E − 08	1.03E − 03	9.23E − 03	1.12E − 01	0.00E + 00	
MOEAD [7]	Best	NaN	NaN	NaN	4.00E − 01	NaN	1.82E + 02	NaN	
	Average	NaN	NaN	NaN	2.63E − 01	NaN	NaN	NaN	
	Std	NaN	NaN	NaN	7.20E − 02	NaN	NaN	NaN	
NSGAII [5]	Best	NaN	4.57E − 01	NaN	2.39E − 01	1.94E − 01	1.80E + 11	0.00E + 00	
	Average	NaN	NaN	NaN	2.38E − 01	1.50E − 01	5.02E + 10	NaN	
	Std	NaN	NaN	NaN	2.91E − 03	3.34E − 02	6.00E + 10	NaN	
SPEA2 [6]	Best	NaN	NaN	NaN	2.34E − 01	NaN	8.16E + 07	NaN	
	Average	NaN	NaN	NaN	2.29E − 01	NaN	NaN	NaN	
	Std	NaN	NaN	NaN	1.97E − 03	NaN	NaN	NaN	
		R29	**R30**	**R31**	**R32**	**R33**	**R34**	**R35**	
MOCBOA	Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	3.89E − 01	
	Average	8.65E − 01	4.42E − 02	3.70E − 01	1.09E − 01	8.80E − 02	1.04E − 01	6.17E − 01	
	Std	3.14E − 01	9.58E − 02	2.56E − 01	1.72E − 01	1.66E − 01	1.81E − 01	1.61E − 01	
MOGSK [14]	Best	9.93E − 01	2.98E − 01	6.47E − 01	6.41E − 01	4.48E − 01	5.88E − 01	8.80E − 01	
	Average	9.97E − 01	3.59E − 01	7.00E − 01	7.12E − 01	5.72E − 01	6.33E − 01	9.02E − 01	
	Std	1.95E − 03	4.84E − 02	3.09E − 02	4.71E − 02	7.50E − 02	2.66E − 02	1.39E − 02	
MOEAD [7]	Best	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
	Average	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
	Std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
NSGAII [5]	Best	7.82E − 01	5.29E − 01	5.14E − 01	7.80E − 01	0.00E + 00	0.00E + 00	4.92E − 01	
	Average	6.66E − 01	NaN	NaN	NaN	NaN	NaN	NaN	
	Std	1.50E − 01	NaN	NaN	NaN	NaN	NaN	NaN	
SPEA2 [6]	Best	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
	Average	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
	Std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
		R36	**R37**	**R38**	**R39**	**R40**	**R41**	**R42**	
MOCBOA	Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	2.34E − 01	0.00E + 00	0.00E + 00	
	Average	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	3.86E − 01	0.00E + 00	0.00E + 00	
	Std	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	8.60E − 02	0.00E + 00	0.00E + 00	
MOGSK [14]	Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	7.73E − 01	2.98E − 01	0.00E + 00	
	Average	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	3.05E + 00	4.73E + 01	0.00E + 00	
	Std	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	6.95E + 00	5.29E + 01	0.00E + 00	
MOEAD [7]	Best	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
	Average	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
	Std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
NSGAII [5]	Best	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
	Average	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
	Std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
SPEA2 [6]	Best	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
	Average	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
	Std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
		R43	**R44**	**R45**	**R46**	**R47**	**R48**	**R49**	**R50**
MOCBOA	Best	5.74E − 01	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	6.11E − 01
	Average	8.01E − 01	0.00E + 00	0.00E + 00	0.00E + 00	4.65E − 01	1.31E − 01	0.00E + 00	6.11E − 01
	Std	1.22E − 01	0.00E + 00	0.00E + 00	0.00E + 00	4.02E − 01	2.10E − 01	0.00E + 00	1.17E − 16
MOGSK [14]	Best	8.21E − 01	0.00E + 00	2.15E − 01	0.00E + 00	4.70E − 02	8.16E − 01	2.05E − 02	6.11E − 01
	Average	9.60E − 01	8.50E − 01	7.17E − 01	3.45E − 01	1.69E − 01	9.05E − 01	9.69E − 02	6.11E − 01
	Std	6.07E − 02	3.16E − 01	2.93E − 01	2.90E − 01	4.70E − 02	6.15E − 02	4.36E − 02	1.98E − 05
MOEAD [7]	Best	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Average	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NSGAII [5]	Best	NaN	NaN	NaN	NaN	NaN	NaN	NaN	7.51E − 03
	Average	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
SPEA2 [6]	Best	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Average	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

(Continued)

Table 9 (continued)

	R1	R2	R3	R4	R5	R6	R7	
Std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

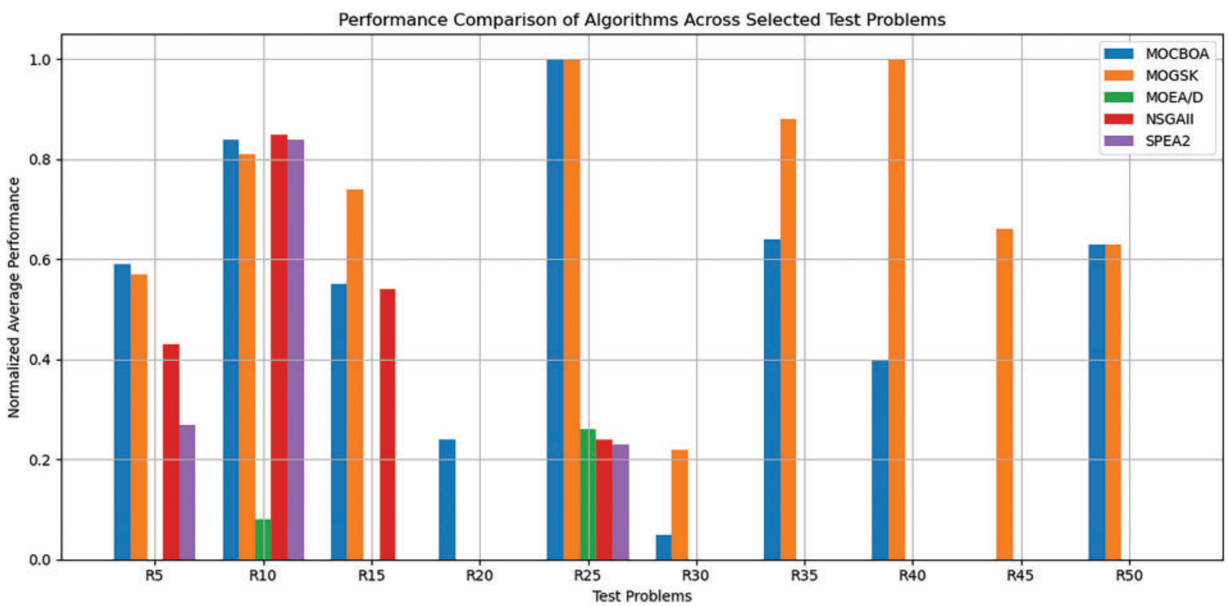

Figure 11: Comparing the proposed MOCBOA algorithm with the MOGSK, NSGAII, MOEA/D and SPEA2 algorithms, on average across selected test problems from R1–R50

Across multiple test problems, MOGSK generally achieves higher and more stable Hypervolume values than MOCBOA. However, MOCBOA occasionally outperforms MOGSK in certain instances, particularly in the best values. For example, in R2, MOCBOA achieves 7.40E − 01, outperforming MOGSK's 3.80E − 01. However, MOGSK performs better in many other test cases, such as R1 (9.92E − 01) and R22 (9.82E − 01), where it surpasses MOCBOA.

Stability-wise, MOGSK tends to exhibit lower standard deviations in many cases, indicating a more consistent performance. Although MOCBOA has competitive results in some problems, it displays more variability across the test problems, leading to higher standard deviations in multiple cases.

Regarding MOEA/D, the results indicate a large number of NaN values, meaning it fails to converge in multiple instances. When MOEA/D does produce results, its Hypervolume values are generally lower than those of MOGSK and MOCBOA. For example, in R25, MOEA/D records 4.00E − 01, whereas MOCBOA achieves 8.64E − 01. However, in R7, MOEA/D (4.79E − 01) performs better than MOCBOA (0.00E + 00).

NSGA-II performs competitively in certain cases but is inconsistent overall. For example, in R1, MOCBOA records 9.92E − 01, while NSGA-II only reaches 5.96E − 01. However, in R27, NSGA-II shows an extremely high best value (1.80E + 11), which might be an anomaly. Additionally, NSGA-II often exhibits higher standard deviations, suggesting a less stable performance compared to MOGSK and MOCBOA.

SPEA2 also suffers from convergence issues, as indicated by many NaN values across the test cases. However, when valid results exist, MOCBOA often outperforms SPEA2. For instance, in R5, MOCBOA achieves 4.89E − 01, while SPEA2 only records 2.77E − 01. This means that SPEA2 performs better than MOCBOA in R7 (4.83E − 01 vs. 0.00E + 00).

The MOCBOA algorithm's competitive advantage in multi-objective optimization is demonstrated by its performance on a variety of test problems. It continuously performs better than a number of benchmark algorithms, including SPEA2, NSGA-II, and MOEA/D, particularly when it comes to the optimum Hypervolume values. In many cases, MOCBOA performs exceptionally well, demonstrating excellent solution quality and continuously attaining better average Hypervolume outcomes, which suggests its capacity to efficiently explore the solution space. Even though MOCBOA typically outperforms MOGSK, it can occasionally have problems with stability, as evidenced by greater standard deviations, which point to less reliable performance. Despite this, MOCBOA frequently exhibits superior convergence and dependable outcomes in contrast to MOEA/D and SPEA2, which have several NaN values as a result of convergence problems. Although MOCBOA performs exceptionally well in many problem contexts, it occasionally falls behind MOGSK, which exhibits a superior stability and solution quality. However, MOCBOA is an excellent candidate for multi-objective optimization tasks due to its ability to handle a variety of optimization problems and consistently produce competitive results. This is especially true when paired with upcoming enhancements to improve stability and robustness across all test cases.

6 Conclusions

In this paper, we have proposed a multi-objective optimization algorithm, MOCBOA, which extends the recently introduced CBOA algorithm. The MOCBOA has been designed by incorporating several effective strategies to enhance its performance. Specifically, it utilizes fast non-dominated sorting and crowding distance measures to control the distribution and diversity of solutions during both the exploitation and exploration phases of the optimization process. Additionally, an external repository has been employed to store the best solutions, which serves to guide the population towards the optimal Pareto front. The archive of solutions has been updated using various dominance relations. The effectiveness of the proposed MOCBOA algorithm has been tested on a comprehensive set of benchmarks, which were sourced from various domains, including mechanical design, chemical engineering, power electronics, process design and synthesis, and power system optimization. The comparative results of MOCBOA's performance in problems such as R2 and R27 with Hypervolume values of 0.740 and 1.65, along with its capacity to produce dependable and consistent outcomes in situations when some competitors stumble (such as when they run across NaN values), show that it is a robust and competitive strategy, particularly in terms of solution quality, stability, and convergence speed. Since multi-objective optimization is the foundation of our approach, its usefulness goes well beyond the scope of the current study. Because MOCBOA may simultaneously optimize conflicting objectives, it is especially well-suited to solving complicated problems across a variety of fields.

Despite the promising performance of the MOCBOA algorithm, there are a few limitations that need further investigation. First, although Epsilon dominance has proven to be effective, the algorithm's performance might vary for certain types of problems, particularly those with a complex Pareto front. In such cases, the trade-off between exploration and exploitation may need further tuning. Additionally, the computational cost of maintaining and updating the external repository with multiple dominance relations could increase with larger problem instances, affecting the algorithm's scalability. Therefore, future work should focus on improving the algorithm's efficiency for large-scale problems by managing the external repository and reducing the computational burden. Further experiments on a wider range of problem domains, particularly those with non-convex or discontinuous Pareto fronts, would also provide valuable insights into the robustness of MOCBOA. Also, hybridizing MOCBOA with knowledge-based heuristics, fuzzy systems, and machine learning techniques could be explored to further enhance its performance in dynamic and real-time optimization scenarios.

Acknowledgement: The authors present their appreciation to King Saud University for funding the publication of this research through Researchers Supporting Program (RSPD2024R809), King Saud University, Riyadh, Saudi Arabia.

Funding Statement: The research is funded by Researchers Supporting Program number (RSPD2024R809), King Saud University, Riyadh, Saudi Arabia.

Author Contributions: Nour Elhouda Chalabi: Conceptualization, resources, methodology, data curation, experimental design, software implementation, writing—original draft preparation. Abdelouahab Attia: Methodology, data collection, software implementation, validation, visualization, writing—review and editing. Abdulaziz S. Almazyad: Project administration, funding acquisition, validation, formal analysis, writing—review and editing. Ali Wagdy Mohamed: Methodology, validation, formal analysis, writing—review and editing. Frank Werner: Supervision, validation, formal analysis, writing—review and editing. Pradeep Jangir: Methodology, visualization, writing—original draft preparation. Mohammad Shokouhifar: Supervision, formal analysis, writing—review and editing. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Wang WC, Tian WC, Xu DM, Zang HF. Arctic puffin optimization: a bio-inspired metaheuristic algorithm for solving engineering design optimization. Adv Eng Softw. 2024;195:103694. doi:10.1016/j.advengsoft.2024.103694.
2. Wang WC, Xu L, Chau KW, Zhao Y, Xu DM. An orthogonal opposition-based-learning Yin-Yang-pair optimization algorithm for engineering optimization. Eng Comput. 2022;38(2):1149–83. doi:10.1007/s00366-020-01248-9.
3. Al Mindeel T, Spentzou E, Eftekhari M. Energy, thermal comfort, and indoor air quality: multi-objective optimization review. Renew Sustain Energy Rev. 2024;202(11):114682. doi:10.1016/j.rser.2024.114682.
4. Han Z, Han W, Ye Y, Sui J. Multi-objective sustainability optimization of a solar-based integrated energy system. Renew Sustain Energy Rev. 2024;202(4):114679. doi:10.1016/j.rser.2024.114679.
5. Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput. 2002;6(2):182–97. doi:10.1109/4235.996017.
6. Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm. In: TIK report 103. Zurich, Switzerland: ETH Zurich; 2001.
7. Zhang Q, Li H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput. 2007;11(6):712–31. doi:10.1109/TEVC.2007.892759.
8. Coello Coello CA, Reyes-Sierra M. Multi-objective particle swarm optimizers: a survey of the state-of-the-art. Int J Comput Intell Res. 2006;2(3):287–308. doi:10.5019/j.ijcir.2006.68.
9. Mirjalili S, Saremi S, Mirjalili SM, Coelho LDS. Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. Expert Syst Appl. 2016;47(6):106–19. doi:10.1016/j.eswa.2015.10.039.
10. Pradhan PM, Panda G. Solving multiobjective problems using cat swarm optimization. Expert Syst Appl. 2012;39(3):2956–64. doi:10.1016/j.eswa.2011.08.157.
11. Patil MV, Kulkarni AJ. Pareto dominance based multiobjective cohort intelligence algorithm. Inf Sci. 2020;538(1):69–118. doi:10.1016/j.ins.2020.05.019.

12. Chalabi NE, Attia A, Bouziane A, Hassaballah M, Alanazi A, Binbusayyis A. An archive-guided equilibrium optimizer based on *Epsilon* dominance for multi-objective optimization problems. Mathematics. 2023;11(12):2680. doi:10.3390/math11122680.
13. Chalabi NE, Attia A, Bouziane A, Hassaballah M. An improved marine predator algorithm based on *Epsilon* dominance and Pareto archive for multi-objective optimization. Eng Appl Artif Intell. 2023;119(1):105718. doi:10.1016/j.engappai.2022.105718.
14. Chalabi NE, Attia A, Alnowibet KA, Zawbaa HM, Masri H, Mohamed AW. A multi-objective gaining-sharing knowledge-based optimization algorithm for solving engineering problems. Mathematics. 2023;11(14):3092. doi:10.3390/math11143092.
15. Menchaca-Méndez A, Montero E, Antonio LM, Zapotecas-Martínez S, Coello Coello CA, Riff MC. A co-evolutionary scheme for multi-objective evolutionary algorithms based on ε—dominance. IEEE Access. 2019;7:18267–83. doi:10.1109/ACCESS.2019.2896962.
16. Li H, Deng J, Zhang Q, Sun J. Adaptive *Epsilon* dominance in decomposition-based multiobjective evolutionary algorithm. Swarm Evol Comput. 2019;45(1):52–67. doi:10.1016/j.swevo.2018.12.007.
17. Zouache D, Ben Abdelaziz F. Guided *Manta* Ray foraging optimization using *Epsilon* dominance for multi-objective optimization in engineering design. Expert Syst Appl. 2022;189(1):116126. doi:10.1016/j.eswa.2021.116126.
18. Deutz AH, Emmerich M, Wang Y. Many-criteria dominance relations. In: Brockhoff D, Emmerich M, Naujoks B, Purshouse R, editors. Many-criteria optimization and decision analysis. Natural computing series. Cham, Switzerland: Springer Cham; 2023. p. 81–111.
19. Batista LS, Campelo F, Guimaraes FG, Ramírez JA. Pareto cone ε-dominance: improving convergence and diversity in multiobjective evolutionary algorithms. In: Evolutionary Multi-Criterion Optimization: 6th International Conference, EMO 2011; 2011; Ouro Preto, Brazil. p. 76–90.
20. Zouache D, Got A, Drias H. An external archive guided Harris Hawks optimization using strengthened dominance relation for multi-objective optimization problems. Artif Intell Rev. 2023;56(3):2607–38. doi:10.1007/s10462-022-10235-z.
21. Shen J, Wang P, Wang X. A controlled strengthened dominance relation for evolutionary many-objective optimization. IEEE Trans Cybern. 2022;52(5):3645–57. doi:10.1109/TCYB.2020.3015998.
22. Gu Q, Chen H, Chen L, Li X, Xiong NN. A many-objective evolutionary algorithm with reference points-based strengthened dominance relation. Inf Sci. 2021;554:236–55. doi:10.1016/j.ins.2020.12.025.
23. Zouache D, Ben Abdelaziz F. MGDE: a many-objective guided differential evolution with strengthened dominance relation and bi-goal evolution. Ann Oper Res. 2022;19(1):45. doi:10.1007/s10479-022-04641-3.
24. Laumanns M, Thiele L, Deb K, Zitzler E. Combining convergence and diversity in evolutionary multiobjective optimization. Evol Comput. 2002;10(3):263–82. doi:10.1162/106365602760234108.
25. Trojovská E, Dehghani M. A new human-based metahurestic optimization method based on mimicking cooking training. Sci Rep. 2022;12(1):14861. doi:10.1038/s41598-022-19313-2.
26. Ikeda K, Kita H, Kobayashi S. Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal?. In: Proceedings of the 2001 Congress on Evolutionary Computation; 2001 May 27–30; Seoul, Republic of Korea: IEEE; 2002. p. 957–62. doi:10.1109/CEC.2001.934293.
27. Tian Y, Cheng R, Zhang X, Su Y, Jin Y. A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. IEEE Trans Evol Comput. 2019;23(2):331–45. doi:10.1109/TEVC.2018.2866854.
28. Chao T, Wang S, Wang S, Yang M. A many-objective evolutionary algorithm combining simplified hypervolume and a method for reference point sampling based on angular relationship. Appl Soft Comput. 2024;163(3):111881. doi:10.1016/j.asoc.2024.111881.
29. Kannan BK, Kramer SN. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. In: ASME 1993 Design Technical Conferences; 1993 Sep 19–22; Albuquerque, NM, USA; 2021. p. 103–12. doi:10.1115/DETC1993-0382.
30. Narayanan S, Azarm S. On improving multiobjective genetic algorithms for design optimization. Struct Optim. 1999;18(2):146. doi:10.1007/BF01195989.

31. Chiandussi G, Codegone M, Ferrero S, Varesio FE. Comparison of multi-objective optimization methodologies for engineering applications. Comput Mathem Applicat. 2012;63(5):912–42. doi:10.1016/j.camwa.2011.11.057.
32. Deb K. Evolutionary algorithms for in engineering design. Evolution Algor Engd Comput Sci. 1999;2:135–61.
33. Osyczka A, Kundu S. A genetic algorithm-based multicriteria optimization method. In: The First World Congress of Structural and Multidisciplinary Optimization;1995 Goslar, Germany; p.909–14.
34. Azarm S, Tits A, Fan M. Tradeoff-driven optimization-based design of mechanical systems. In: 4th Symposium on Multidisciplinary Analysis and Optimization; 1992; USA. 4758 p. doi:10.2514/6.1992-4758.
35. Ray T, Liew KM. A swarm metaphor for multiobjective design optimization. Eng Optim. 2002;34(2):141–53. doi:10.1080/03052150210915.
36. Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part I: solving problems with box constraints. IEEE Trans Evol Comput. 2013;18(4):577–601. doi:10.1109/TEVC.2013.2281535.
37. Cheng FY, Li XS. Generalized center method for multiobjective engineering optimization. Eng Optim. 1999;31(5):641–61. doi:10.1080/03052159908941390.
38. Huang HZ, Gu YK, Du X. An interactive fuzzy multi-objective optimization method for engineering design. Eng Appl Artif Intell. 2006;19(5):451–60. doi:10.1016/j.engappai.2005.12.001.
39. Steven G. Evolutionary algorithms for single and multicriteria design optimization. A. Osyczka. Springer Verlag, Berlin, 2002, ISBN 3-7908-1418-01. Struct Multidiscip Optim. 2002;24:88–9. doi:10.1007/s00158-002-0218-y.
40. Coello C, Lamont G, Veldhuizen DV. Evolutionary algorithms for solving multi-objective problems. In: Genetic and evolutionary computation series. 2nd ed. New York, NY, USA: Springer; 2007.
41. Parsons MG, Scott RL. Formulation of multicriterion design optimization problems for solution with scalar numerical optimization methods. J Ship Res. 2004;48(1):61–76. doi:10.5957/jsr.2004.48.1.61.
42. Fan L, Yoshino T, Xu T, Lin Y, Liu H. A novel hybrid algorithm for solving multiobjective optimization problems with engineering applications. Math Probl Eng. 2018;2018:5316379. doi:10.1155/2018/5316379.
43. Dhiman G, Kumar V. Multi-objective spotted hyena optimizer: a Multi-objective optimization algorithm for engineering problems. Knowl Based Syst. 2018;150(6):175–97. doi:10.1016/j.knosys.2018.03.011.
44. Mahon K. Optimal engineering design: principles and applications (mechanical engineering series, volume 14). J Oper Res Soc. 1983;34(7):652–4. doi:10.1057/jors.1983.155.
45. Zhang H, Peng Y, Hou L, Tian G, Li Z. A hybrid multi-objective optimization approach for energy-absorbing structures in train collisions. Inf Sci. 2019;481(4):491–506. doi:10.1016/j.ins.2018.12.071.
46. Floudas CA, Pardalos PM. A collection of test problems for constrained global optimization algorithms. Berlin/Heidelberg, Germany: Springer; 1990.
47. Ryoo HS, Sahinidis NV. Global optimization of nonconvex NLPs and MINLPs with applications in process design. Comput Chem Eng. 1995;19(5):551–66. doi:10.1016/0098-1354(94)00097-2.
48. Guillén-Gosálbez G. A novel MILP-based objective reduction method for multi-objective optimization: application to environmental problems. Comput Chem Eng. 2011;35(8):1469–77. doi:10.1016/j.compchemeng.2011.02.001.
49. Kocis GR, Grossmann IE. A modelling and decomposition strategy for the minlp optimization of process flowsheets. Comput Chem Eng. 1989;13(7):797–819. doi:10.1016/0098-1354(89)85053-7.
50. Kocis GR, Grossmann IE. Global optimization of nonconvex mixed-integer nonlinear programming (MINLP) problems in process synthesis. Ind Eng Chem Res. 1988;27(8):1407–21. doi:10.1021/ie00080a013.
51. Floudas CA. Nonlinear and mixed-integer optimization: fundamentals and applications. New York, NY, USA: Oxford University Press; 1995.
52. Rathore AK, Holtz J, Boller T. Synchronous optimal pulsewidth modulation for low-switching-frequency control of medium-voltage multilevel inverters. IEEE Trans Ind Electron. 2010;57(7):2374–81. doi:10.1109/TIE.2010.2047824.
53. Rathore AK, Holtz J, Boller T. Optimal pulsewidth modulation of multilevel inverters for low switching frequency control of medium voltage high power industrial AC drives. In: 2010 IEEE Energy Conversion Congress and Exposition; 2010 Sep 12–16; Atlanta, GA, USA: IEEE; 2010. p. 4569–74. doi:10.1109/ECCE.2010.5618413.

54. Edpuganti A, Rathore AK. Fundamental switching frequency optimal pulsewidth modulation of medium-voltage cascaded seven-level inverter. IEEE Trans Ind Appl. 2015;51(4):3485–92. doi:10.1109/TIA.2015.2394485.
55. Edpuganti A, Dwivedi A, Rathore AK, Srivastava RK. Optimal pulsewidth modulation of cascade nine-level (9L) inverter for medium voltage high power industrial AC drives. In: IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society; 2015 Nov 9–12; Yokohama, Japan: IEEE; 2015. p. 4259–64. doi:10.1109/IECON.2015.7392764.
56. Edpuganti A, Rathore AK. Optimal pulsewidth modulation for common-mode voltage elimination scheme of medium-voltage modular multilevel converter-fed open-end *stator* winding induction motor drives. IEEE Trans Ind Electron. 2017;64(1):848–56. doi:10.1109/TIE.2016.2586678.
57. Mishra S, Kumar A, Singh D, Kumar Misra R. Butterfly optimizer for placement and sizing of distributed generation for feeder phase balancing. Adv Intell Syst Comput. 2019;799:519–30.
58. Biswas PP, Suganthan PN, Mallipeddi R, Amaratunga GAJ. Multi-objective optimal power flow solutions using a constraint handling technique of evolutionary algorithms. Soft Comput. 2020;24(4):2999–3023. doi:10.1007/s00500-019-04077-1.
59. Kumar A, Das S, Mallipeddi R. An inversion-free robust power-flow algorithm for microgrids. IEEE Trans Smart Grid. 2021;12(4):2844–59. doi:10.1109/TSG.2021.3064656.
60. Kumar A, Jha BK, Das S, Mallipeddi R. Power flow analysis of islanded microgrids: a differential evolution approach. IEEE Access. 2021;9:61721–38. doi:10.1109/ACCESS.2021.3073509.
61. Jha BK, Kumar A, Dheer DK, Singh D, Misra RK. A modified current injection load flow method under different load model of EV for distribution system. Int Trans Electr Energ Syst. 2020;30(4):e12284. doi:10.1002/2050-7038.12284.
62. Kumar A, Jha BK, Singh D, Misra RK. A new current injection based power flow formulation. Electr Power Compon Syst. 2020;48(3):268–80. doi:10.1080/15325008.2020.1758846.
63. Kumar A, Jha BK, Dheer DK, Singh D, Misra RK. Nested backward/forward sweep algorithm for power flow analysis of droop regulated islanded microgrids. IET Generation Trans Dist. 2019;13(14):3086–95. doi:10.1049/iet-gtd.2019.0388.
64. Kumar A, Jha BK, Singh D, Misra RK. Current injection-based Newton-Raphson power-flow algorithm for droop-based islanded microgrids. IET Generation Trans Dist. 2019;13(23):5271–83. doi:10.1049/iet-gtd.2019.0575.
65. Kumar A, Jha BK, Dheer DK, Misra RK, Singh D. A nested-iterative Newton-raphson based power flow formulation for droop-based islanded microgrids. Electr Power Syst Res. 2020;180(4):106131. doi:10.1016/j.epsr.2019.106131.
66. Rivas-Dávalos F, Irving MR. An approach based on the strength Pareto evolutionary algorithm 2 for power distribution system planning. In: Evolutionary multi-criterion optimization. Vol. 3410. Berlin/Heidelberg: springer; 2005: 707–20. doi:10.1007/b106458.
67. Tian Y, Cheng R, Zhang X, Jin Y. PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [educational forum]. IEEE Comput Intell Mag. 2017;12(4):73–87. doi:10.1109/MCI.2017.2742868.