Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 12, No.3, December : 2024

PERFORMANCE ASSESSMENT OF HYBRID ARTIFICIAL BEE COLONY AND NSGA-II ALGORITHMS ON MULTI-DIMENSIONAL META-HEURISTIC OPTIMIZATION: CEC 2024 BENCHMARK EVALUATION

Mr. Vishal T. Metkari, Assistant Professor, Dept.Of Electrical Engineering, Sanjeevan Engineering and Technology Institute, Tal-Panhala, Maharashtra
(Prof.) Dr. H. T. Jadhav, Dean, Faculty of Science & Technology, S.N.D.T. Women's university Mumbai.

ABSTRACT
The multi-objective optimization algorithms (MOO) are used to obtain the best compromising solutions when two or more objective functions need to be optimized simultaneously. To develop new meta-heuristic algorithms and evaluate on the benchmark functions is the most challenging task. In this paper, performance of the Hybrid Artificial Bee Colony and NSGAII developed meta-heuristic algorithms are evaluated on the recently developed CEC 2024 benchmark functions. However, the solutions reported in the literature, are either inferior or the constraints are violated. The algorithm proposed in this paper is an integration of an Artificial Bee Colony optimization algorithm and a Non-Dominated Sorting Genetic Algorithm-II. The effectiveness of the proposed algorithm is evaluated with well-defined IEEE CEC 2024 benchmark suite. Therefore, different meta-heuristic algorithms are considered which solve the benchmark functions with different dimensions. The performance of some basic, advanced meta-heuristics algorithms and the algorithms that participated in the CEC2024 competition have been experimentally investigated and many observations, recommendations, conclusions have been reached. This study focuses on researching two issues. The first is the redesigning and implementation of the Hybrid ABC-NSGAII, which is a novel meta-heuristic hybrid algorithm, in order to find better solutions to multi-objective benchmark problems. The second is the application of this hybrid multi-objective ABC-NSGA-II algorithm to solve the many objective, non-linear problem. As a result of this research, We used CEC 2024 as one of the most up-to-date and prestigious benchmark suites in the literature in order to test and verify the performance of the ABC-NSGA-II algorithm we had developed. In addition, we used the most up-to-date and widely used performance metrics and statistical analysis methods to analyze and compare the performance of the ABC-NSGAII and the five competing algorithms. Performance metrics and statistical analysis results clearly revealed that the proposed ABC-NSGAII is a competitive algorithm for finding PSt (the space of decision variables belonging to the optimization problem) and PFt (the space of objective functions), which represent optimum solution sets for multi-objective problems of different types and characteristics. Using the crowding distance-based Pareto archiving process, a significant improvement was achieved in the exploitation and exploration performance in the multi-objective search space of the ABC-NSGAII. We investigated the optimum PSt and PFt for the MOO problem using the ABC-NSGAII. We compared the results obtained from the experimental trials with the studies in the literature and the analysis results showed that we had found the best quality solution for the MOO problem.
Keywords: Constrained multi-objective evolutionary algorithm, constrained multi-objective optimization problem, metaheuristic optimization algorithm.

1. Introduction:

Optimisation is used in engineering, science, operations research, planning, finance, mechanics, etc. to achieve goals [1]. Single-objective optimisation problems consider one objective function. In such cases, the objective function's optimum value is sought. The literature contains traditional and nature-inspired optimisation algorithms for these problems. However, most engineering and other practical problems require simultaneous optimisation of many objective functions. Multi-objective optimisation problems are difficult to solve because they involve conflicting objectives. In this case, no single solution meets all competing goals. Instead, the decision-maker receives a set of solutions that compromise the objectives and are not superior to each other. All such solutions are non-dominated, or Pareto-optimal. In real-world problems, the Pareto-optimal set may have many non-dominated solutions and increase with problem size [2]. Traditional algorithms for solving multi-objective optimisation problems fail to solve problems with discontinuous and/or discrete search spaces or provide suboptimal solutions. Because they simultaneously consider multiple solutions, evolutionary and swarm intelligence-based optimisation seem suitable for multi-objective optimisation problems. This lets the solver find all Pareto optimal set solutions in one algorithm run instead of several [3]. A constrained multi-objective optimisation problem in computational science is evaluated using an artificial bee colony algorithm and a non-dominated genetic algorithm (ABC-NSGA-II). Constraint-based multi-objective optimisation problems dominate today's problem-solving. Multi-objective optimisations offer more solutions than single-objective optimisations, which usually find one. People call these solutions Pareto Fronts. A reliable multi-objective optimisation strategy should consistently produce solutions that span these PFs, ensuring they are feasible and optimal for the problems. These strategies are complicated by balancing many goals. Metaheuristic (MH) algorithms are typically tested on simple optimisation problems. Engineering design has unique and varied requirements, unlike other search problems. Effective optimisation requires tailoring and refining these algorithms to engineering tasks' complex requirements.

This study focuses on researching two issues. The first is the redesigning and implementation of the Hybrid ABC-NSGAII, which is a novel meta-heuristic hybrid algorithm, in order to find better solutions to multi-objective benchmark problems. The second is the application of this hybrid multi-objective ABC-NSGA-II algorithm to solve the many objective, non-linear problem. As a result of this research, We used CEC 2024 as one of the most up-to-date and prestigious benchmark suites in the literature in order to test and verify the performance of the ABC-NSGA-II algorithm we had developed. In this suite, 24 different multimodal-type optimization problems having different features are defined. Moreover, to evaluate the superiority of the proposed ABC-NSGA-II algorithm, it was compared with the OMNI [8], NSGA [9,10], NSGAIII [11,12], SPEA2 [13], and MOABC [14] algorithms found in the literature, which are among the latest and most powerful methods developed to solve this type of multiobjective optimization problem. In addition, we used the most up-to-date and widely used performance metrics and statistical analysis methods to analyze and compare the performance of the ABC-NSGAII and the five competing algorithms. Performance metrics and statistical analysis results clearly revealed that the proposed ABC-NSGAII is a competitive algorithm for finding PSt (the space of decision variables belonging to the optimization problem) and PFt (the space of objective functions), which represent optimum solution sets for multi-objective problems of different types and characteristics. Using the crowding distance-based Pareto archiving process, a significant improvement was achieved in the exploitation and exploration performance in the multi-objective search space of the ABC-NSGAII. We investigated the optimum PSt and PFt for the MOO problem using the ABC-NSGAII. We compared the results obtained from the experimental trials with the studies in the literature and the analysis results showed that we had found the best quality solution for the MOO problem.

2. Literature Review and Research gap/Novelty in presented paper:

MOO has gained popularity in many fields because it boosts industrial productivity. Traditional approaches for MOO problem solving include ε–Constraint, Scalarization, and Pareto. The ε–Constraint method prioritises one objective while limiting others. Weights for each objective turn multi-objective problems into single-objective problems during scalarization. As the best non-scalar compromise, the Pareto approach addresses all objectives. Evolutionary algorithms (EAs) and swarm intelligence methods followed the Vector-Evaluated Genetic Algorithm (VEGA) in MOO optimisation. TLBO, Harmony Search, and Differential Evolution (DE) have been shown to solve complex MOO problems. Hybrid methods like Evolutionary Algorithms (EAs) and swarm intelligence have improved solution exploration, exploitation, and convergence. The artificial bee colony (ABC) algorithm is popular due to its simplicity, few control parameters, and adaptability. An established MOO problem-solving method, Non-dominated Sorting Genetic Algorithm II (NSGA-II), is combined with the ABC algorithm. The ABC algorithm's bee phases modify solutions, while NSGA-II's crossover and mutation operations create offspring. NSGA-II's non-dominated sorting finds the Pareto-optimal front, guiding the algorithm to the best solutions. Data clustering and other complex optimisation tasks benefit from this hybrid method's exploration-exploitation balance. Recent benchmarking studies using IEEE Congress on Evolutionary Computation (CEC) competitions have helped evaluate cutting-edge optimisation algorithms. The CEC benchmark suite tests single-objective, multi-objective, and large-scale optimisation problems with a variety of functions. Including noisy, noiseless, constrained, and unconstrained test environments allows for comprehensive algorithm performance evaluation. For validation and algorithm comparison, the suite uses statistical methods like the Friedman test and Wilcoxon signed-rank test. CEC 2024 benchmark test functions are crucial for assessing modern multi-objective optimisation algorithms like the hybrid ABC-NSGA-II. Optimisation algorithms face challenges from non-convex and complex numerical test functions. Several multi-criteria decision-making methods are used to evaluate the hybrid ABC-NSGA-II algorithm and select the best solutions from the non-dominated set. Hypervolume indicators and statistical analysis compare the hybrid algorithm to basic NSGA-II to assess its efficacy. The CEC competitions and benchmarking frameworks, especially the COCO benchmark suite, have shaped meta-heuristic algorithm evaluation and development. Benchmarking exercises improve nature-inspired algorithms like ABC and NSGA-II and provide insights into algorithm performance. Researchers can better understand hybrid optimisation methods' strengths and weaknesses in real-world applications by assessing algorithmic efficiency through performance metrics, computation time, and complexity. In conclusion, the hybrid ABC-NSGA-II algorithm is promising for multi-objective optimisation. Combining ABC's solution modification and NSGA-II's offspring generation powers offers a balanced approach to complex MOO problems. Benchmarking the algorithm on CEC 2024 test functions can assess its real-world applicability and effectiveness, improving optimisation techniques in various domains. Research Contributions are as follows;

- Hybrid ABC-NSGA-II Algorithm Development: Multi-objective optimisation problems can be solved using a novel hybrid artificial bee colony (ABC) algorithm and the non-dominated sorting genetic algorithm-II (NSGA-II). A hybrid approach efficiently addresses multiple conflicting objectives, improving optimisation results.
- The proposed methodology can be used to solve more complex optimisation problems like single-objective computationally expensive numerical optimisation, multi-niche optimisation, constrained real-parameter single-objective problems, and constrained/multi-objective optimisation problems.
- Simulation of Real-World Optimisation Challenges: The algorithm simulates real-world optimisation problems' difficulty, reflecting practical challenges and ensuring its applicability in diverse problem domains.
- New optimisation algorithms' strengths and weaknesses are examined in the study. The hybrid ABC-NSGA-II improves algorithmic performance over previous methods.
- Statistics for Performance Comparison: Hypervolume, Inverted Generational Distance (IGD), and Spread indicators are used to compare the proposed hybrid algorithm to the basic NSGA-II, providing a detailed performance assessment.
- CEC 2024 Benchmark Suite testing: The developed Hybrid ABC-NSGA-II is tested on 15 CEC 2024 benchmark suite multi-modal, multi-objective optimisation problems. Statistical analysis shows the algorithm's performance improvement in difficult optimisation environments.
- Latest Competitors: The improved Hybrid ABC-NSGA-II is compared to cutting-edge algorithms. The hybrid algorithm solves multi-objective optimisation problems competitively according to statistical analysis of four performance metrics.

- Top-notch MOO solutions: The Hybrid ABC-NSGA-II algorithm solves the multi-objective optimisation (MOO) problem with Pareto front (PF) and Pareto set (PS) solutions of higher quality than literature solutions, proving its superiority.

For the sake of convenience, some of the common acronyms used in this paper are summarized in Table 1. The remaining part of the paper is organized as follows: Section 2 presents the formulation of the Multi-Objective Optimization Problem (MOOP). Section 3 provides the theory of the hybrid ABC-NSGA-II algorithm. Section 4 explains the detailed procedure of applying the ABC-NSGA-II algorithm to the MOOP problem. The strategy to identify the best compromising solution is given in Section 5. The results of three test systems, using the proposed approach, along with comparative analysis, are discussed in Section 6. Section 7 includes the statistical analysis and comparison of the proposed method with ABC-NSGA-II. Finally, conclusions are drawn in Section 8.

3. Concept of Multi-objective Optimization:

The multi-objective optimization problem is defined as given in Eq. 1 [31].

$$\underset{x \in R^i}{\text{minimize/maximize}}\ F(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), f_3(\vec{x}), \ldots, f_i(\vec{x})\}$$

$$\begin{array}{ll} \text{Objectives} & \\ \text{Equality constraints} & \emptyset_a(\vec{x}) = 0, (a = 1, 2, \ldots, A), \\ \text{Inequality constraints} & \varphi_b(\vec{x}) \le 0, (b = 1, 2, \ldots, B) \\ \vec{x} = (x_1, x_2, x_3, \ldots, x_D), & d = 1, 2, \ldots, D l_d \le x_d \le u_d \end{array}$$

where F ($\vec{x}$) is a vector consisting of the i-objective functions and $\vec{x}$ is the vector represented by the D-dimensional decision variables used in defining the F ($\vec{x}$). The boundaries on which decision variables are defined in the search space are represented by l_d and u_d, respectively. The equality and inequality constraints that are defined using decision variables and that must be met by the solution found for the optimization problem are represented by $\emptyset_a(\vec{x})$and $\phi_b(\vec{x})$, respectively [31]. Thus, conflicting objective functions in Eq. make it difficult to find the best solution. Most meta-heuristic search (MHS) algorithms are designed to solve single-objective optimisation problems, so they cannot find feasible solutions for conflicting objective functions. Problems with conflicting objective functions require Pareto-based solutions. This requires MHS algorithm redesign. This requires integrating Pareto-based archiving into MHS algorithms that solve single-objective optimisation problems. Most multi-objective optimisation algorithms in the literature are designed this way. However, the Pareto approach has drawbacks. This method archives non-dominated solutions. You must choose between these solutions. Handling operations are needed when the archive exceeds its maximum solution count. In the Pareto-based approach, archive handling is one of the biggest challenges because it's hard to decide which solutions to keep and which to delete. This process can use random, sequential, or probabilistic selection. None of these methods are reliable or effective. Archive handling is stable and effective using crowding distance. Thus, Pareto-based multi-objective optimisation algorithms perform better. Based on these explanations, this study's multi-objective ABC-NSGAII algorithm used crowding distance-based archiving, resulting in a stable and effective MOO problem optimisation.

Definitions for the terminology of the Pareto archiving method and crowding distance are presented below [14,31–33]: Decision and objective spaces: In optimization problems, all of the values that decision variables can take are called decision variables space. This space is defined by the D-dimensional $\vec{x} = (x_1, x_2, x_3, \ldots, x_D)$, $l_d \le x_d \le u_d$ in Eq. (1) [31]. The objective space is defined as the equivalent of the values of each decision variable vector defined in the space of decision variables in the i-dimensional objective functions $F(\vec{x}) = \{f1(\vec{x}), f2(\vec{x}), f3(\vec{x}), \ldots, fi(\vec{x})\}$ in Eq. (1). Pareto dominance relationship: The comparison of the dominance of the vectors defined in the i-dimensional objective space with respect to each other is i > 1 and ∀i ∈ N. According to the definitions in Eq. (1), let $\vec{x}$ and $\vec{y}$ vectors be defined in the D-dimensional space of the decision

variables. Let these two vectors be represented by the vectors F($\vec{x}$) and F($\vec{y}$) in the I dimensional objective space for the objective functions defined in Eq. (1). According to these definitions, if the F($\vec{x}$) vector has a better objective value than the F ($\vec{y}$) vector for at least one of the objective functions and does not have a worse objective value than the F ($\vec{y}$) vector for any of the objective functions, then the $\vec{x}$-vector is dominant over the $\vec{y}$-vector (denoted as $\vec{x} \prec \vec{y}$) [31].
Pareto-optimal solution: Given a set of solutions, if a solution ($\vec{x}$) is not dominated by any other solution, then that solution is said to be Pareto-optimal [31].

$$\{\nexists \vec{y} \in X | \vec{y} \prec \vec{x}\}$$

Pareto-optimal set (PS): This is a set of solutions that cannot be dominated in the objective space. The set of these solutions in the decision space is called the PS [31].

$$PS = \{\vec{x} \text{ and } \vec{y} \in \mathrm{X} | \nexists \vec{y} \prec \vec{x}\}$$

Pareto-optimal front (PF): The set of values obtained by solution candidates for the objective functions in the Pareto-optimal set is called the PF [31].

$$|PF = \{O\left(\vec{x}\right) \vec{x} \in PS\}$$

4. Hybrid Meta-heuristic Algorithm

4.1: Artificial Bee Colony:

The ABC algorithm's artificial bee colony has employed, onlooker, and scout bees. An onlooker bee waits on the dance area to choose a food source, while an employed bee goes to the food source it previously visited. Scouts are bees that randomly search.. The main algorithm steps are below: The ABC algorithm divides each search cycle into three steps: sending the employed bees onto food sources and measuring their nectar amounts; selecting food sources by onlookers after sharing the information of employed bees and determining nectar amounts; and determining the scout bees and sending them onto possible food sources. Bees randomly select food source positions and nectar amounts during initialisation. Then, these bees enter the hive and share source nectar information with the dance area bees. After sharing the information, every employed bee goes to the food source area she visited in the previous cycle since it's in her memory, then chooses a new food source using visual information in the neighbourhood. An onlooker chooses a food source based on nectar information distributed by employed bees on the dance area at the third stage.
Each solution (food source) x_i (i = 1, 2, . . . , SN) is a D-dimensional vector. Here, D is the number of optimization parameters. After initialization, the population of the positions (solutions) is subjected to repeated cycles, C = 1, 2, . . . ,Cmax, of the search processes of the employed bees, the onlooker bees and scout bees. An artificial employed or onlooker bee probabilistically changes her memory position (solution) for finding a new food source and tests its nectar value (fitness value). REAL bees produce new food sources by visually comparing food sources in a region. A comparison of food source positions is used to create a new position in our model. In the model, artificial bees use no comparison data. The bee forgets the old position and remembers the new one if the nectar amount is higher. She retains her position otherwise. After finding food sources (solutions), employed bees share their position and nectar information with onlooker bees on the dance area.. If its nectar is better, the bee remembers the new position and forgets the old one. An onlooker bee chooses a food source depending on the probability value associated with that food source, p_i, calculated by the following expression (2.1):

$$p_i = \frac{\text{fit}_i}{\sum_{n=1}^{SN} \text{fit}_n}, \qquad (2.1)$$

where fit$i$ is the fitness value of the solution i evaluated by its employed bee, which is proportional to the nectar amount of the food source in the position i and SN is the number of food sources which is equal to the number of employed bees (BN). In this way, the employed bees exchange their information with the onlookers.
In order to produce a candidate food position from the old one, the ABC uses the following expression (2.2):

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \qquad (2.2)$$

where $k \in \{1, 2, . . . , \text{BN}\}$ and $j \in \{1, 2, . . . , D\}$ are randomly chosen indexes. Although k is determined randomly, it has to be different from i. $\varphi i,j$ is a random number between [−1, 1]. It controls the production of a neighbour food source position around xi,j and the modification represents the comparison of the neighbour food positions visually by the bee. Equation 2.2 shows that as the difference between the parameters of the xi,j and xk,j decreases, the perturbation on the position xi,j decreases, too. Thus, as the search approaches to the optimum solution in the search space, the step length is adaptively reduced. If a parameter from this operation exceeds its limit, it can be set to an acceptable value. This work sets the parameter exceeding its limit to its limit. Scouts replace the bees' abandoned nectar with a new food source. This is simulated by randomly creating a position and replacing the abandoned one in the ABC algorithm. If a position cannot be improved after a limit number of cycles, the ABC algorithm abandons that food source. Each candidate source position vi,j is produced and evaluated by the artificial bee, then compared to xi,j. The old food is remembered if the new one has equal or better nectar. Otherwise, the old one stays. Thus, greedy selection is used to choose between old and new food sources. ABC algorithm uses four selection processes: (1) a global selection process used by artificial onlooker bees to discover promising regions (2.1), (2) a local selection process carried out in a region by artificial bees and onlookers based on local information (in the case of real bees, this includes the colour, shape, and fragrance of the flowers) (bees cannot identify the type of nectar). The bee remembers the present one otherwise. (4) scout-random selection. To search robustly, exploration and exploitation must occur simultaneously. Scouts control exploration in the ABC algorithm, while onlookers and employed bees exploit the search space.

4.2 Non-dominated Sorting Genetic Algorithm-II

Deb et al. [32] developed the powerful multi-objective optimisation algorithm Non-dominated Sorting Genetic Algorithm II (NSGA-II) in 2002. Kalyanmoy Deb and colleagues developed NSGA-II, a sophisticated and efficient multiobjective optimisation algorithm. NSGA-II was created to improve multiobjective evolutionary algorithms (MOEAs) like the original NSGA. High computational complexity, lack of elitism, diversity maintenance dependent on user-defined parameters, and inefficient constraint handling were these issues.
The efficient nondominated sorting procedure of NSGA-II is its core. The algorithm calculates how many and which solutions dominate each population solution. This lets solutions be grouped by dominance count into fronts. In the first front, no other solutions dominate, in the second front, only those in the first front dominate, and so on. This sorting is much faster than the original NSGA, reducing computational complexity from $O(MN^3)$ to $O(MN^2)$, where M is the number of objectives and N is the population size. NSGA-II calculates crowding distance to maintain solution diversity. The density of solutions around each front solution is estimated by calculating a crowding distance. Sorting solutions by objective and calculating normalised distances between neighbouring solutions accomplishes this. An infinite crowding distance protects boundary solutions in the population. This method eliminates the need for a predefined sharing parameter, a major drawback of the original NSGA. For real-coded solutions, NSGA-II uses simulated binary crossover (SBX) and polynomial mutation to maintain genetic diversity. After creating the offspring population, NSGA-II combines the parent and offspring populations into a 2^N population. This population is sorted again using nondominated sorting to divide solutions into fronts. New generations are formed from the best N

solutions. The best solutions are preserved in this selection process, ensuring elitism. If the current front does not fit entirely into the new population, the best solutions are chosen by crowding distance until the population is filled.
A NSGA-II algorithm stepI. Initialisation Population Creation: Generate an N-sized random population. Every person is a potential solution. Objective Functions: Assess each population member's objective function. These values determine how well each solution meets each goal. Sorting without dominance: Calculating Dominance: Determine the dominance count and which solutions each solution dominates. Formation of Fronts: Divide solutions by dominance counts: First Front: Undominated solutions. Second Front: Mostly first-front solutions. Like the previous fronts, subsequent fronts are defined. Calculating Crowding Distance: Use a crowding distance to estimate the density of solutions around each front solution. This is done by sorting solutions by objective and calculating normalised distances between neighbours. Set an infinite crowding distance for boundary solutions to preserve them in the population. Selection: Binary Tournament Select solutions with better ranks (lower front number) and larger crowding distances within the same front. Gene Operators: Create offspring by crossing selected solutions. SBX is used for real-coded NSGA-II solutions. Mutation: Randomly alter offspring to maintain genetic diversity. Real-coded solutions often use polynomial mutation. Create a new population of offspring by applying crossover and mutation operators to the selected solutions. Mix Populations: Parent and Offspring Populations: Combine the parent and offspring populations into a 2N population. Nondominated Combined Population Sorting: Sort Combined Population: Sort solutions into fronts using nondominated sorting. Selecting Next Generation: Elitism: Choose the top N solutions from the population to create the next generation: Fill Population: Include all first-front solutions. If the current front does not fit entirely into the new population, sort solutions by crowding distance and include the best until the population is filled. Handling constraints: Modify dominance to handle constraints: We always prioritise feasible solutions over infeasible ones. The infeasible solution with the smallest constraint violation is best. Repeat: Iteration: Repeat fitness evaluation and generation selection until a stopping criterion is met.This elitist strategy is more robust. SPEA's clustering techniques require careful parameter tuning, but NSGA-II's crowding distance mechanism maintains solution diversity without a predefined sharing parameter. The constraint handling mechanism in NSGA-II is simple and effective. NSGA-II simplifies constraint-handling without penalty parameters or complex computations by prioritising feasible solutions and preferring those with fewer constraint violations. It is more robust and easier to implement than the original NSGA and SPEA, which can struggle with parameter settings and diversity in constrained environments. For multiobjective optimisation problems, NSGA-II outperforms SPEA and the original NSGA due to its efficient nondominated sorting, elitism, parameter-less diversity preservation, and simple constraint handling. Its improvements improve convergence to the Pareto-optimal front, maintain a diverse set of solutions, and handle constraints, creating a balanced and powerful optimisation tool. The Hybrid Artificial Bee Colony (ABC) and Non-dominated Sorting Genetic Algorithm II (NSGA-II) solves multi-objective optimisation problems by combining the ABC algorithm's exploration powers with the NSGA-II algorithm's exploitation powers. The hybrid ABC-NSGA-II algorithm combines ABC exploration phases with NSGA-II genetic operations and non-dominated sorting.

4.3 Pseudocode of Hybrid ABC-NSGA-II

Initialize population P with size N
Evaluate objective function values for each solution in P
Repeat until termination criterion is met:

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 12, No.3, December : 2024

Employed Bee Phase:
For each solution i in P:
Generate new solution i' using employed bee formula
If i' is better than i, replace i with i' in P and reset trial counter
Else, increment trial counter for i
Calculate fitness and probability for each solution in P
Onlooker Bee Phase:
For each solution i in P:
Select a solution j based on probability
Generate new solution i' using onlooker bee formula
If i' is better than i, replace i with i' in P and reset trial counter
Else, increment trial counter for i
Scout Bee Phase:
For each solution i in P:
If trial counter exceeds limit:
Replace i with a new random solution and reset trial counter
NSGA-II Phase:
Crossover and Mutation:
Perform crossover and mutation to generate offspring population Q
Combine P and Q
Perform non-dominated sorting on combined population
Calculate crowding distance for each solution
Select top N solutions based on rank and crowding distance to form new P
Output the non-dominated solutions in P

The Hybrid Artificial Bee Colony (ABC) and Non-dominated Sorting Genetic Algorithm II (NSGA-II) solve multi-objective optimisation problems by combining the ABC algorithm's exploration and exploitation powers. ABC exploration phases are combined with NSGA-II genetic operations and non-dominated sorting in the hybrid algorithm. Important hybrid approach choices: Traditional ABC algorithms use a separate archive for non-dominated solutions, but the hybrid algorithm uses a single integrated population for all solutions, promoting diversity from the start. The modified onlooker bee phase balances solution modification by determining neighbour selection probability based on dominance and fitness. Constrained Handling: Before non-dominated sorting, feasible and infeasible solutions are ordered by constraint violation severity. Every solution's fitness and selection probability are calculated using dominance count and crowding distance.

4.4 Hybrid ABC-NSGA-II Algorithm

The Hybrid Artificial Bee Colony (ABC) and Non-dominated Sorting Genetic Algorithm II (NSGA-II) algorithm combines the ABC algorithm's exploratory search mechanism with NSGA-II's robust sorting and crowding distance mechanisms. This hybrid approach improves diversity and convergence efficiency, making it ideal for complex multi-objective optimisation problems. Hybrid ABC-NSGA-II Algorithm Unified Population Management Highlights: Traditional ABC algorithms use a separate archive for non-dominated solutions, but the hybrid algorithm uses one population. This approach promotes diversity from the start by allowing all population solutions to mate and contribute to the next generation. In the enhanced onlooker bee phase, solutions are selected based on their dominance probability, calculated using non-dominated sorting and crowding distance. This maintains population diversity and balanced solution modification. Constrained Problems: Before non-dominated sorting, the algorithm separates feasible and infeasible solutions. The rank and crowding distance of feasible solutions determine their ranking, while constraint violation severity ranks infeasible solutions. The dominance count and crowding distance values are used to calculate each solution's fitness and selection probability. This method ensures fair and balanced selection, improving algorithm performance. Even for ten-objective problems, computational experiments show that NSGA-III does not always outperform NSGA-II. Comparison results depend more on test problem type than objective number. Since its proposal, NSGA-II has been one of the most popular evolutionary multiobjective optimisation (EMO) algorithms, despite its limitations with many-objective problems. Performance evaluation of new EMO algorithms usually involves NSGA-II. NSGA-III performed better on frequently-used DTLZ1-4 test problems than maximisation ones. NSGA-II solved many-objective knapsack problems better. These findings suggest that test problem type affects performance comparison results more than objective number. In conclusion, NSGA-III is a useful benchmark for new many-objective algorithms, but NSGA-II remains a crucial reference point in evolutionary multiobjective optimisation due to its track record and availability. Optimisation algorithm performance evaluation must take test problem type into account.

5. Performance metrics

Performance indicators compare algorithm solutions. Optimisation studies with conflicting objective functions seek the true Pareto sets (PFt and PSt) of the best solutions that cannot dominate each other. These PFt and PSt vectors represent the solution candidates' objective function values and decision variables. These explain the four main metrics used to evaluate multi-objective optimisation studies: HV [34–36], PSP [37], IGDX [32], and IGDF [32]. HV and IGDF metrics measure algorithms' Pareto-front coverage. These metrics measure the overlap ratio of the estimated PFe vector and the true PFt vector. HV volume and IGDF distance calculations are used. PSP and IGDX evaluate algorithm Pareto-set performance. Both metrics assess estimated PSe and true PSt vector coverage. This is done over the overlap ratio and IGDX for the PSP metric, and distance calculation for IGDX. Metrics are defined below: 1/HV hypervolume: Zitzler and Thiele proposed this PFe set formation performance metric [31,36]. An s-vector representing the objective space's worst point is used in HV metric calculations. The volume calculation between the s-vector and the z-vector representing PFe in w-space is given in Eq. (21).

$$HV\,(s, w) = \lambda_D \left(\bigcup_{z \in s} [z; w] \right) \qquad (21)$$

where w ∈ R is for all z ∈ s z≺w in this space and λD is the D-dimensional Lebesgue measure.

Inverted generational distances (IGDF and IGDX): These metrics evaluate algorithm convergence in the decision variable and objective spaces [31]. Both methods are remote. Both metrics are calculated using Eq. (22) according to these explanations.

$$IGD\,(F, X)\,(P^*, P) = \frac{\sum_{v \in P^*} d\,(v, P)}{|P^*|} \qquad (22)$$

In Eq. (22), when calculating the IGDF metric, PFt is used as the P* vector and PFe is used as the P-vector. Similarly, when calculating the IGDX metric, PSt is used as the P* vector and PSe as the P-vector.

Pareto sets proximity (1/PSP): The PSP metric is calculated in three steps by referencing the PSt and PSe vectors defined in the space of decision variables. In first step, using Eq. (20), the overlap ratio of these two vectors is calculated [37].

$$CR = \left(\prod_{i=1}^{m} \delta_i \right)^{1/2D} \qquad (23)$$

where m is the dimension of the objective space and the δi is the normalized overlap ratio between PFt and PFe for the ith objective function and δi is calculated as given in Eq. (24).

$$\delta_i = \left(\frac{\min(PFe_i^{\max}, PFt_i^{\max}) - \max(PFe_i^{\min}, PFt_i^{\min})}{PFt_i^{\max} - PFt_i^{\min}} \right)^2 \qquad (24)$$

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 12, No.3, December : 2024

Finally, the performances of the algorithms according to the PSP metric in an m-dimensional objective space are calculated as in Eq. (25) [37].

$$PSP = \frac{CR}{IGDX} \quad (25)$$

More information on performance metrics can be found in [34–37] reference studies.

6. CEC 2024 Multiobjective Optimization Benchmark Functions

MPMOPs with Independent Pareto Optimal Solutions:BF1 to BF6: The difficulty and dimensionality of these benchmarks allow for a complete algorithm performance evaluation. MPMOPs with Common Pareto Optimal Solutions:1–11: These competitive problems, derived from CEC 2018, have common Pareto optimal solutions across scenarios. Each problem is different in complexity and goals. The benchmark functions used in CEC 2024 to validate multi-objective optimisation algorithms are described here:

Table 1: Multiparty Multiobjective Optimization Problems (MPMOPs) with Independent Pareto Optimal Solutions

Problem	Mathematical Formulation	Range/Constraints	Type of Function	Objectives	Notes
BF1	$f_1(x) = \sum_{i=1}^{n} x_i^2$ $f_2(x) = \sum_{i=1}^{n} (x_i - 2)^2$	$0 \leq x_i \leq 1$	Convex, Continuous	2	Simple quadratic functions
BF2	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - \sqrt{\frac{f_1(x)}{g(x)}}\right)$ $g(x) = 1 + 9\sum_{i=2}^{n} \frac{x_i}{n-1}$	$0 \leq x_i \leq 1$	Concave, Continuous	2	Based on ZDT1
BF3	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right)$ $g(x) = 1 + 9\sum_{i=2}^{n} \frac{x_i}{n-1}$	$0 \leq x_i \leq 1$	Concave, Continuous	2	Based on ZDT2
BF4	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - \sqrt{\frac{f_1(x)}{g(x)}} - \frac{f_1(x)}{g(x)}\sin(10\pi x_1)\right)$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}(x_i - 0.5)^2 -$ $\cos(20\pi(x_i - 0.5))$	$0 \leq x_i \leq 1$	Non-convex, Continuous	2	Based on ZDT3
BF5	$f_1(x) = x_1$ $f_2(x) =$ $g(x)\left(1 - (f_1(x)/g(x))^{0.5} - (f_1(x)/g(x))\sin(10\pi x_1)\right)$ $g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$	$0 \leq x_i \leq 1$	Non-convex, Continuous	2	Based on ZDT4
BF6	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right)$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}(x_i - 0.5)^2 -$ $\cos(20\pi(x_i - 0.5))$	$0 \leq x_i \leq 1$	Non-convex, Continuous	2	Based on ZDT6

1.BF1 to BF6: These benchmark functions encompass various configurations of Pareto optimal front shapes, including convex, concave, and mixed forms. They evaluate the capacity of algorithms to manage diverse complexities and discontinuities in solution space. 2. MPMOP1 to MPMOP9: These functions are intended for multiparty multi-objective optimisation challenges. Each function evaluates distinct facets of optimisation algorithms, including their capacity to navigate dynamic and intricate solution spaces. These tables offer a detailed summary of the benchmark functions and formulations utilised in CEC 2024 for the validation of multi-objective optimisation algorithms.

5.1 Setting experiments

In experimental trials to test and verify the ABC-NSGAII, literature standards were used. One of the most recent standards for multi-objective optimisation algorithm competition is the CEC 2024 [32]. The experimental settings followed these standards to ensure fairness when comparing algorithms

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 12, No.3, December : 2024

and analysis reliability. Settings are listed below:For each benchmark optimisation problem, algorithms were run 21 times independently and the best PFe and PSe sets were saved. The archive and group size was 200 * N_ops, where N_ops is the number of PFt/PSt Pareto-optimal optimisation solutions. MaxFEs was the search process termination criterion. All algorithms had maxFEs set to 10,000 * N_ops. All statistical comparisons used 5% significance. Both the NSGA-II and ABC algorithms had their parameters optimised to maximise performance. NSGA-II had 200 parent and offspring populations and 200 generations, except for the BNH problem, which had 100 generations. Fixed error tolerance (δ) of 0.0001 was used for equality constraints. Random float sampling was used. Binary tournament selection selected the best of two candidates. Simulation Binary Crossover (SIMBX) was used with a distribution index (η) of 15 and a crossover probability (p) of 0.9. Polynomial Mutation was used with a distribution index (η) of 20 and a mutation probability (p) of 0.9. Consider 2 nearest neighbours to control crowding.

Table 2: MPMOPs with Common Pareto Optimal Solutions

Problem	Mathematical Formulation	Range/Constraints	Type of Function	Objectives	Notes
MPMOP1	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - \sqrt{f_1(x)/g(x)}\right)$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i$	$0 \leq x_i \leq 1$	Concave, Continuous	2	Derived from DTLZ1
MPMOP2	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - (f_1(x)/g(x))^2\right)$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i$	$0 \leq x_i \leq 1$	Concave, Continuous	2	Derived from DTLZ2
MPMOP3	$f_1(x) = x_1$ $f_2(x) =$ $g(x)\left(1 - \sqrt{\frac{f_1(x)}{g(x)}} - \frac{f_1(x)}{g(x)}\sin(10\pi x_1)\right)$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i$	$0 \leq x_i \leq 1$	Non-convex, Continuous	2	Derived from DTLZ3
MPMOP4	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right)$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i$	$0 \leq x_i \leq 1$	Concave, Continuous	2	Derived from DTLZ4
MPMOP5	$f_1(x) = x_1$ $f_2(x) =$ $g(x)\left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2 - \frac{f_1(x)}{g(x)}\sin(10\pi x_1)\right)$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i$	$0 \leq x_i \leq 1$	Non-convex, Continuous	2	Derived from DTLZ5
MPMOP6	$f_1(x) = x_1$ $f_2(x) =$ $g(x)\left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2 \sin(10\pi x_1)\right)$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i$	$0 \leq x_i \leq 1$	Non-convex, Continuous	2	Derived from DTLZ6

However, the ABC algorithm, which models bee foraging, used several controls. The colony size was 125, matching the population size in reference [21], and the maximum number of cycles (MCN) was the maximum generation. 50% onlookers, 50% employed bees, and one scout bee comprised the colony. More scouts encourage exploration, while more onlookers on a food source increase exploitation. All experiments were repeated 30 times with different random seeds for reliability. Optimising both algorithms' problem-solving performance required these parameter settings.

Problem	Mathematical Formulation	Range/Constraints	Type of Function	Objectives	Notes
MPMOP7	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right)$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i$	$0 \leq x_i \leq 1$	Concave, Continuous	2	Derived from DTLZ7
MPMOP8	$f_1(x) = x_1$ $f_2(x) =$ $g(x)\left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2 - \frac{f_1(x)}{g(x)}\sin(10\pi x_1)\right)$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i$	$0 \leq x_i \leq 1$	Non-convex, Continuous	2	Derived from DTLZ8
MPMOP9	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right)$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i$	$0 \leq x_i \leq 1$	Concave, Continuous	2	Derived from DTLZ9
MPMOP10	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - \sqrt{\frac{f_1(x)}{g(x)}}\sin(10\pi x_1)\right)$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}(x_i -$ $0.5)^2$	$0 \leq x_i \leq 1$	Non-convex, Continuous	2	Derived from DTLZ10
MPMOP11	$f_1(x) = x_1$ $f_2(x) =$ $g(x)\left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2 - \frac{f_1(x)}{g(x)}\sin(10\pi x_1)\right)$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i$	$0 \leq x_i \leq 1$	Non-convex, Continuous	2	Derived from DTLZ11

6. Results and analyses

The Friedman [38,39] and Wilcoxon [40,41] non-parametric test methods were used to analyse experimental trial data and compare algorithm performance. Experimental results are in two subsections: First subsection: The ABC-NSGAII algorithm's performance on CEC 2024 multi-objective benchmark problems is detailed. The proposed algorithm's exploitation and exploration capabilities were compared to competitors using four performance metrics. The convergence and stability of multi-modal algorithms with different geometric shapes are also shown graphically.

Table 2: Friedman Test ranking obtained by algorithm according to 1/HV IGDX, IGDF, and 1/PSP metrics

Algorithms	**1/HV**	**IGDX**	**IGDF**	**1/PSP**	**Average**
ABC-NSGAII	**4.6865**	**1.9107**	**2.6131**	**1.9345**	**2.7862**
NSGAIII	4.5813	2.4365	3.3214	2.4504	3.1974
MOPSO	2.6369	4.0002	3.0516	4.1825	3.4682
DN NSGAII	3.2659	4.2877	4.5794	4.2798	4.1032
NSGAII	**2.6310**	5.3095	3.3631	5.3829	4.1716
SPEA2	4.5119	5.1111	5.2123	5.0536	4.9722
MOABC	5.6865	4.9425	5.8591	4.7163	5.3011

Table 3: Pairwise comparison results between ABC-NSGAII and its competitors by problem types using the 1/PSP metric

Vs. ABC-NSGAII +/=/-	**Convex linear (7, 8, 9, 16, 17, 18)**	**Convex nonlinear (1, 2, 4, 5, 10, 13, 19)**	**Concave linear (11, 12, 20, 22, 23, 24)**	**Concave nonlinear (3, 6, 14, 15, 21)**

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 12, No.3, December : 2024

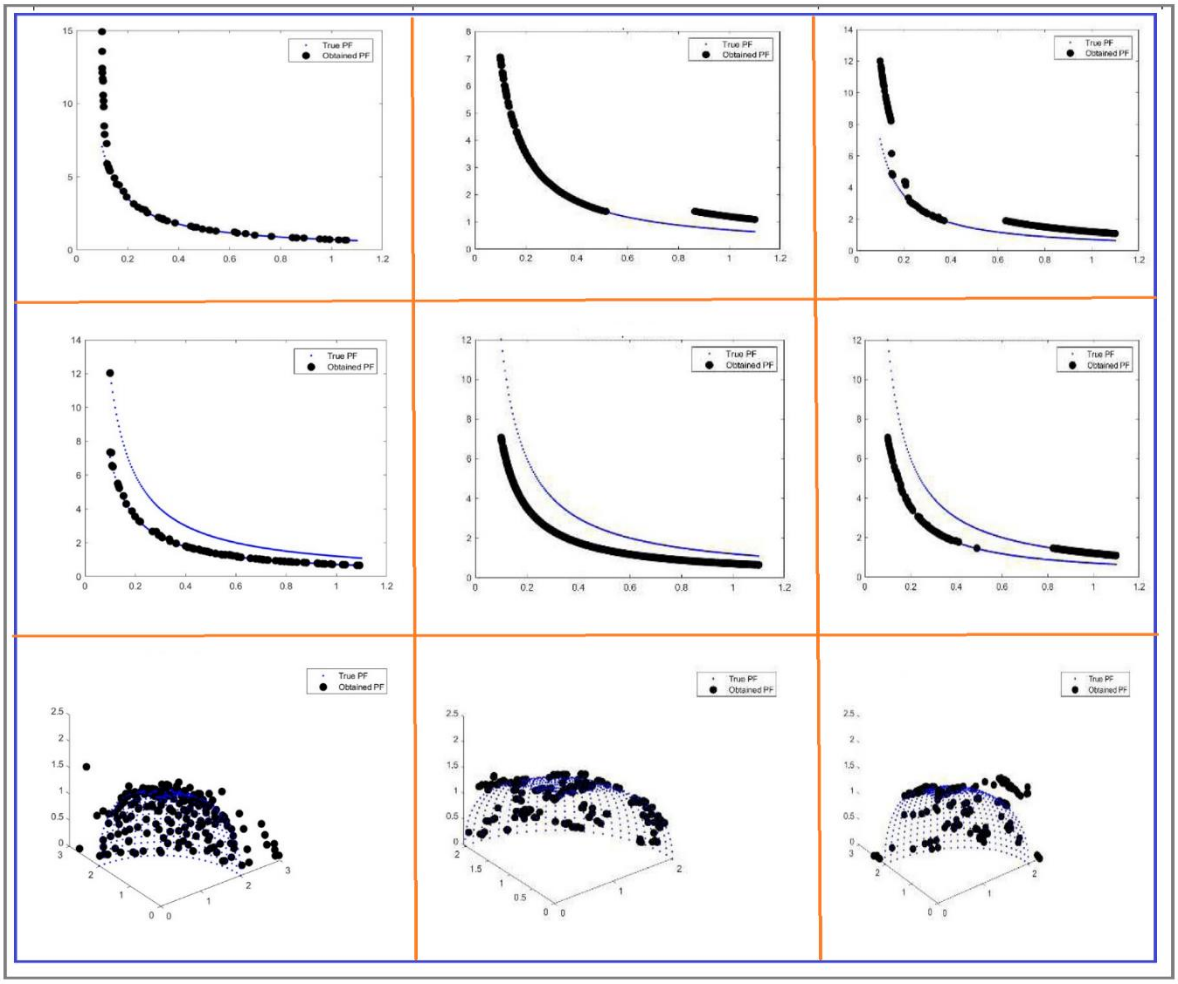
NSGAIII	0/4/2	0/4/3	0/0/6	1/0/4
MOEA/D	2/1/3	1/1/5	0/0/6	0/0/5
MOPSO	2/1/3	1/1/5	0/0/6	0/0/5
MOABC	1/0/5	0/1/6	0/0/6	0/0/5
SPEA2	2/1/3	0/1/6	0/0/6	0/0/5
NSGAII	2/0/4	0/1/6	0/0/6	0/0/5

This table provides the pairwise comparison results between ABC-NSGAII and its competitors across different problem types using the 1/PSP metric. The comparisons are categorized by convex linear, convex nonlinear, concave linear, and concave nonlinear problem types, showcasing the +, =, and - results.

6.1. Performance of the proposed ABC-NSGAII

ABC-NSGAII performance on benchmark problems and comparisons with competing algorithms are examined in this section. Detailed information follows. Benchmark issues: The CEC 2024 benchmark suite [32] used decision variables and objective functions of various types to fairly compare algorithms and show their performance. The CEC 2024 suite includes 15 multimodal multi-objective optimisation problems with two or more global or local Pareto-optimal sets. Competitors: The experimental trials used widely-respected algorithms like MOPSO [8], NSGAIII [9,10], NSGAII [11,12], SPEA2 [13], and MOABC [14]. Performance metrics: Statistical test methods and the latest performance metrics were used to compare algorithm performance. As described in part 3.1.2. (Performance metrics) of the Method section, the four performance metrics were the reciprocal of Hypervolume (1/HV) [34–36], Pareto Sets Proximity (1/PSP) [37], and Inverted Generational Distance in decision and objective space (IGDX, IGDF) [32].

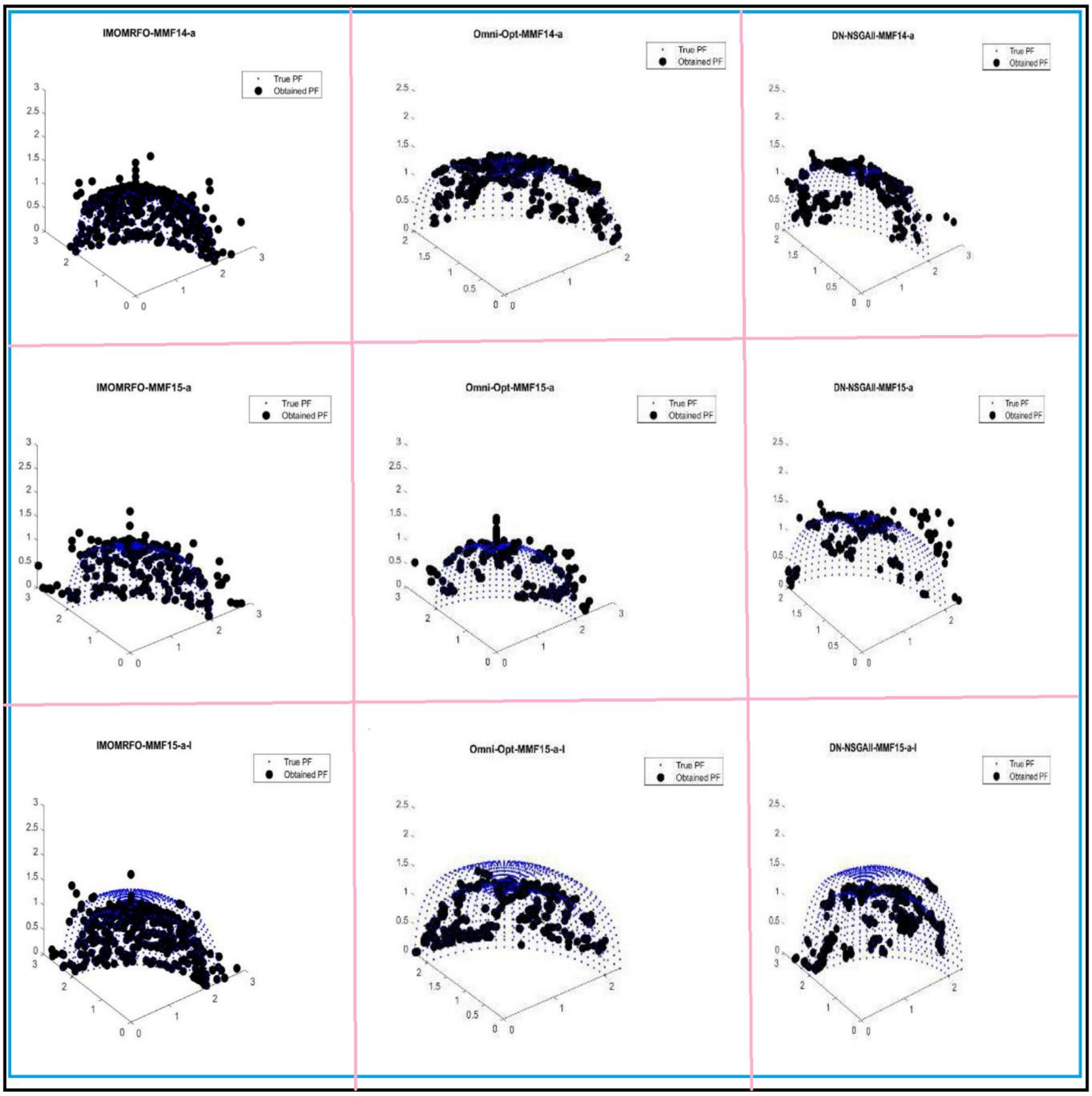
Fig. 2. Performance of algorithms for IGDX, IGDF and 1/PSP metrics.



6.1.1. Statistical analysis results

Based on Section of Experimental Settings, the competing algorithms' PSe and PFe sets for the 24 benchmark problems were used to analyse their performance. Table 2 shows the Friedman test rankings of the six competitor algorithms using 1/HV IGDX, IGDF, and 1/PSP metrics. The NSGAII and OMNI algorithms outperformed their competitors in HV metric ranks. IGDX, IGDF, and PSP ranks showed that the proposed ABC-NSGAII outperformed all competitors. ABC-NSGAII scores for IGDX, PSP, and IGDF metrics indicated good PSe/PSt and PFe/PFt overlap ratios. The ABC-NSGAII's exploitation and exploration capabilities improved with the Pareto-archiving process's crowding distance-based ranking method.

Fig. 3. Performance of algorithms for IGDX, IGDF and 1/PSP metrics.(cont.)



This is evident when comparing MOMRFO and ABC-NSGAII performance metrics scores. Table 3 compares ABC-NSGAII performance to competitors. The (+), (=), and (−) signs indicate the number of optimisation problems where the competitor algorithm surpassed the ABC-NSGAII, where the two algorithms performed similarly, and where the ABC-NSGAII outperformed its competitor. The CEC 2024 benchmark suite has four types of benchmark problems based on objective and decision space forms: convex linear, concave linear, and concave nonlinear. We compared the ABC-NSGAII to its competitors for each problem type. Table 3 shows Wilcoxon statistics. ABC-NSGAII outperformed competitors in all four problem types. In particular, ABC-NSGAII performance improved in multi-objective problems with high Convex Nonlinear, Concave Linear, and Concave Nonlinear complexity.

Table 3: 1/HV, 1/PSP, IGDX, IGDF values of competitors for CEC2020 benchmark problems.

Problem	Metrics	ABC-NSGA-II mean ± std dev	NSGA-III mean ± std dev	MOEA/D mean ± std dev	MOABC mean ± std dev	SPEA2 mean ± std dev	NSGA-II mean ± std dev
BF1	1/HV	1,1445 ± 0,0002	1,1452 ± 0,0013	1,1441 ± 0,0008	1,1567 ± 0,0031	1,1443 ± 0,0004	**1,1439** ± 0,0008
	1/PSP	**0,0278** ± 0,0013	0,0543 ± 0,0101	0,0491 ± 0,0077	0,0904 ± 0,0177	0,0348 ± 0,0045	0,0702 ± 0,0108
	IGDX	**0,0277** ± 0,0013	0,0536 ± 0,0094	0,0487 ± 0,0075	0,0893 ± 0,0167	0,0346 ± 0,0044	0,0693 ± 0,0097
	IGDF	0,0019 ± 0,0001	0,0020 ± 0,0003	0,0017 ± 0,0003	0,0075 ± 0,0013	0,0017 ± 0,0001	**0,0016** ± 0,0002
BF2	1/HV	1,1613 ± 0,0029	1,1565 ± 0,0101	1,1636 ± 0,0344	1,2129 ± 0,0206	1,1671 ± 0,0069	**1,1558** ± 0,0190
	1/PSP	**0,0173** ± 0,0027	0,0593 ± 0,0276	0,0668 ± 0,0504	0,1530 ± 0,0855	0,0418 ± 0,0292	0,0739 ± 0,0555
	IGDX	**0,0168** ± 0,0025	0,0549 ± 0,0230	0,0587 ± 0,0360	0,1430 ± 0,0800	0,0403 ± 0,0254	0,0685 ± 0,0471
	IGDF	**0,0085** ± 0,0009	0,0088 ± 0,0087	0,0165 ± 0,0297	0,0285 ± 0,0080	0,0130 ± 0,0038	0,0088 ± 0,0122
BF3	1/HV	1,8523 ± 0,0014	1,8489 ± 0,0004	1,8475 ± 0,0002	1,8708 ± 0,0074	1,8505 ± 0,0019	**1,8467** ± 0,0003
	1/PSP	**0,0165** ± 0,0013	0,0460 ± 0,0138	0,0432 ± 0,0139	0,0587 ± 0,0154	0,0277 ± 0,0068	0,0746 ± 0,0165
	IGDX	**0,0164** ± 0,0013	0,0460 ± 0,0138	0,0431 ± 0,0139	0,0581 ± 0,0152	0,0276 ± 0,0067	0,0738 ± 0,0165
	IGDF	0,0018 ± 0,0002	0,0016 ± 0,0001	0,0014 ± 0,0001	0,0044 ± 0,0007	0,0016 ± 0,0001	**0,0013** ± 0,0001
BF4	1/HV	1,1445 ± 0,0002	1,1444 ± 0,0007	1,1438 ± 0,0008	1,1519 ± 0,0022	1,1443 ± 0,0003	**1,1433** ± 0,0005
	1/PSP	0,0540 ± 0,0045	**0,0272** ± 0,0062	0,1140 ± 0,0136	0,1573 ± 0,0296	0,0672 ± 0,0059	0,1316 ± 0,0114
	IGDX	**0,0538** ± 0,0045	0,1116 ± 0,0110	0,1131 ± 0,0134	0,1557 ± 0,0277	0,0667 ± 0,0058	0,1306 ± 0,0115
	IGDF	0,0018 ± 0,0001	0,0018 ± 0,0002	0,0015 ± 0,0002	0,0056 ± 0,0009	0,0017 ± 0,0001	**0,0013** ± 0,0001
BF5	1/HV	1,1447 ± 0,0004	1,1452 ± 0,0009	1,1438 ± 0,0002	1,1543 ± 0,0056	1,1442 ± 0,0004	**1,1432** ± 0,0003
	1/PSP	0,0234 ± 0,0048	0,0272 ± 0,0062	0,0248 ± 0,0056	0,0853 ± 0,0264	**0,0226** ± 0,0045	0,0469 ± 0,0077
	IGDX	0,0223 ± 0,0037	0,0270 ± 0,0062	0,0247 ± 0,0055	0,0785 ± 0,0227	**0,0222** ± 0,0044	0,0465 ± 0,0075
	IGDF	0,0022 ± 0,0003	0,0020 ± 0,0002	0,0015 ± 0,0001	0,0072 ± 0,0027	0,0017 ± 0,0001	**0,0013** ± 0,0001
BF6	1/HV	2,3769 ± 0,0086	2,3688 ± 0,0018	2,3646 ± 0,0004	2,4077 ± 0,0199	2,3837 ± 0,0096	**2,3633** ± 0,0003
	1/PSP	**0,0505** ± 0,0140	0,1488 ± 0,0876	0,1646 ± 0,0978	0,4192 ± 0,2608	0,8416 ± 0,4046	0,6662 ± 0,2763
	IGDX	**0,0497** ± 0,0136	0,1451 ± 0,0857	0,1609 ± 0,0963	0,3727 ± 0,1975	0,6758 ± 0,3108	0,5517 ± 0,2067
	IGDF	0,0020 ± 0,0001	0,0019 ± 0,0002	0,0016 ± 0,0001	0,0091 ± 0,0032	0,0017 ± 0,0002	**0,0013** ± 0,0001
MPMOP1	1/HV	**0,0787** ± 0,0003	0,0816 ± 0,0025	0,0797 ± 0,0025	0,0835 ± 0,0022	0,0789 ± 0,0019	0,0820 ± 0,0031
	1/PSP	**0,0140** ± 0,0076	0,1454 ± 0,0872	0,0889 ± 0,0915	0,1356 ± 0,0990	0,0272 ± 0,0305	0,1913 ± 0,1362
	IGDX	**0,0134** ± 0,0066	0,1373 ± 0,0840	0,0823 ± 0,0881	0,1310 ± 0,1006	0,0236 ± 0,0255	0,1875 ± 0,1375
	IGDF	**0,0548** ± 0,0125	0,1758 ± 0,0938	0,1004 ± 0,0981	0,2181 ± 0,0770	0,0649 ± 0,0788	0,1850 ± 0,1078
MPMOP2	1/HV	0,0690 ± 0,0000	**0,0689** ± 0,0000	**0,0689** ± 0,0000	0,0696 ± 0,0003	**0,0689** ± 0,0000	**0,0689** ± 0,0000
	1/PSP	0,0048 ± 0,0002	0,0045 ± 0,0003	0,0043 ± 0,0003	0,0146 ± 0,0037	**0,0034** ± 0,0005	0,0036 ± 0,0004
	IGDX	0,0048 ± 0,0002	0,0045 ± 0,0003	0,0043 ± 0,0003	0,0143 ± 0,0033	**0,0034** ± 0,0005	0,0036 ± 0,0004
	IGDF	0,0154 ± 0,0019	0,0139 ± 0,0014	0,0122 ± 0,0016	0,0625 ± 0,0145	0,0146 ± 0,0017	**0,0110** ± 0,0005
MPMOP 3	1/HV	0,6375 ± 0,0011	0,6365 ± 0,0020	0,6366 ± 0,0030	0,6905 ± 0,0560	0,6362 ± 0,0006	**0,6362** ± 0,0019
	1/PSP	0,0031 ± 0,0003	0,0026 ± 0,0008	0,0028 ± 0,0022	0,0129 ± 0,0106	**0,0015** ± 0,0003	0,0022 ± 0,0012
	IGDX	0,0031 ± 0,0003	0,0026 ± 0,0008	0,0028 ± 0,0022	0,0128 ± 0,0106	**0,0015** ± 0,0003	0,0022 ± 0,0012
	IGDF	0,0047 ± 0,0005	0,0029 ± 0,0003	0,0036 ± 0,0040	0,0279 ± 0,0146	0,0028 ± 0,0002	**0,0027** ± 0,0014
MPMOP4	1/HV	0,0543 ± 0,0000	0,0543 ± 0,0000	0,0543 ± 0,0000	0,0549 ± 0,0002	0,0543 ± 0,0000	**0,0542** ± 0,0000
	1/PSP	**0,0317** ± 0,0020	0,0754 ± 0,0145	0,0708 ± 0,0148	0,1153 ± 0,0670	0,1450 ± 0,0973	0,1046 ± 0,0318
	IGDX	**0,0313** ± 0,0019	0,0735 ± 0,0135	0,0680 ± 0,0115	0,0939 ± 0,0247	0,0989 ± 0,0360	0,0876 ± 0,0147
	IGDF	0,0184 ± 0,0012	0,0226 ± 0,0045	0,0154 ± 0,0011	0,0815 ± 0,0205	0,0172 ± 0,0014	**0,0138** ± 0,0007
MPMOP5	1/HV	0,3654 ± 0,0181	**0,3260** ± 0,0091	0,3348 ± 0,0071	0,3360 ± 0,0347	0,3938 ± 0,0595	0,3543 ± 0,0063
	1/PSP	**0,0520** ± 0,0037	0,0842 ± 0,0074	0,0795 ± 0,0051	0,0758 ± 0,0082	0,2396 ± 0,0648	0,0961 ± 0,0125
	IGDX	**0,0520** ± 0,0037	0,0842 ± 0,0074	0,0795 ± 0,0051	0,0758 ± 0,0082	0,2191 ± 0,0328	0,0961 ± 0,0125
	IGDF	**0,0666** ± 0,0018	0,0963 ± 0,0062	0,0847 ± 0,0031	0,0967 ± 0,0065	0,1832 ± 0,0208	0,0987 ± 0,0056
MPMOP6	1/HV	0,2436 ± 0,0102	**0,2365** ± 0,0158	0,2422 ± 0,0119	0,2454 ± 0,0206	0,2574 ± 0,0320	0,2384 ± 0,0061
	1/PSP	**0,0503** ± 0,0029	0,0805 ± 0,0087	0,0704 ± 0,0056	0,0809 ± 0,0121	0,1236 ± 0,0184	0,0745 ± 0,0083
	IGDX	**0,0503** ± 0,0029	0,0805 ± 0,0087	0,0704 ± 0,0056	0,0808 ± 0,0121	0,1178 ± 0,0156	0,0745 ± 0,0083
	IGDF	**0,0988** ± 0,0043	0,1625 ± 0,0137	0,1390 ± 0,0109	0,1650 ± 0,0136	0,2645 ± 0,0490	0,1433 ± 0,0124
MPMOP7	1/HV	1,1502 ± 0,0063	1,2048 ± 0,1202	1,1534 ± 0,0084	−4,9122 ± 17,2341	1,1593 ± 0,0069	1,1572 ± 0,0118
	1/PSP	2,4679 ± 2,2109	**1,1928** ± 0,5443	1,3824 ± 0,8527	3,7100 ± 4,9374	8,1026 ± 6,5135	2,4469 ± 1,6908
	IGDX	1,3849 ± 0,7426	**0,9164** ± 0,3935	0,9569 ± 0,5053	1,5746 ± 0,8482	2,4209 ± 0,8490	1,4149 ± 0,5492
	IGDF	**0,0029** ± 0,0004	0,0090 ± 0,0066	0,0061 ± 0,0071	0,0226 ± 0,0093	0,0108 ± 0,0047	0,0084 ± 0,0070
MPMOP8	1/HV	0,3721 ± 0,0377	**0,3143** ± 0,0143	0,3268 ± 0,0099	0,3471 ± 0,0342	0,3967 ± 0,0807	0,3467 ± 0,0068
	1/PSP	**0,0587** ± 0,0032	0,1040 ± 0,0077	0,0974 ± 0,0086	0,1007 ± 0,0124	0,2942 ± 0,1636	0,1243 ± 0,0141
	IGDX	**0,0586** ± 0,0032	0,1040 ± 0,0077	0,0974 ± 0,0086	0,1007 ± 0,0124	0,2349 ± 0,0833	0,1243 ± 0,0141
	IGDF	**0,0662** ± 0,0019	0,1041 ± 0,0077	0,0912 ± 0,0030	0,0905 ± 0,0054	0,2087 ± 0,0324	0,1042 ± 0,0072
MPMOP9	1/HV	0,2436 ± 0,0167	**0,2322** ± 0,0133	0,2358 ± 0,0092	0,2375 ± 0,0115	0,2696 ± 0,0312	0,2396 ± 0,0078
	1/PSP	**0,0575** ± 0,0032	0,0984 ± 0,0107	0,0896 ± 0,0076	0,0902 ± 0,0092	0,1770 ± 0,0318	0,1045 ± 0,0117
	IGDX	**0,0573** ± 0,0032	0,0984 ± 0,0107	0,0896 ± 0,0076	0,0902 ± 0,0092	0,1569 ± 0,0168	0,1045 ± 0,0117
	IGDF	**0,0998** ± 0,0041	0,1637 ± 0,0139	0,1444 ± 0,0134	0,1564 ± 0,0159	0,2786 ± 0,0462	0,1573 ± 0,0131

The statistical analysis showed that the crowding distance-based ranking method improved ABC-NSGAII performance in the archiving process. Table 3 shows the ABC-NSGAII and its competitors' 1/HV, IGDX, IGDF, and 1/PSP metrics for the 15 CEC 2024 benchmark suite problems. The 24 problems were solved using the algorithms' PFe and PSe sets from 21 independent trials, and the four metrics' mean values and standard deviations were calculated. Table 4's dark-coloured cells with bold text indicate the algorithms' best values for 60 cases (4 metrics and 15 test functions, 4*15). Last row of Table 4 shows algorithm scores for all cases. ABC-NSGAII performed best in 34 of 60

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 12, No.3, December : 2024

cases. The NSGAII outperformed the ABC-NSGAII in 12 of 60 cases. Approximation graphs showed algorithm convergence, and box-plot graphs showed all algorithm results' error values. We created approximation graphs for 7 of the 15 CEC 2024 benchmark suite problems and box-plot graphs for the other 12. Fig. 3 shows that the ABC-NSGAII, MOPSO, and MOEA/D algorithms found feasible solutions using Pareto front approximations. In problems with more than two objective functions, the ABC-NSGAII had better PFe/PFt overlap than its competitors. No algorithm achieved 100% PFe/PFt overlap.

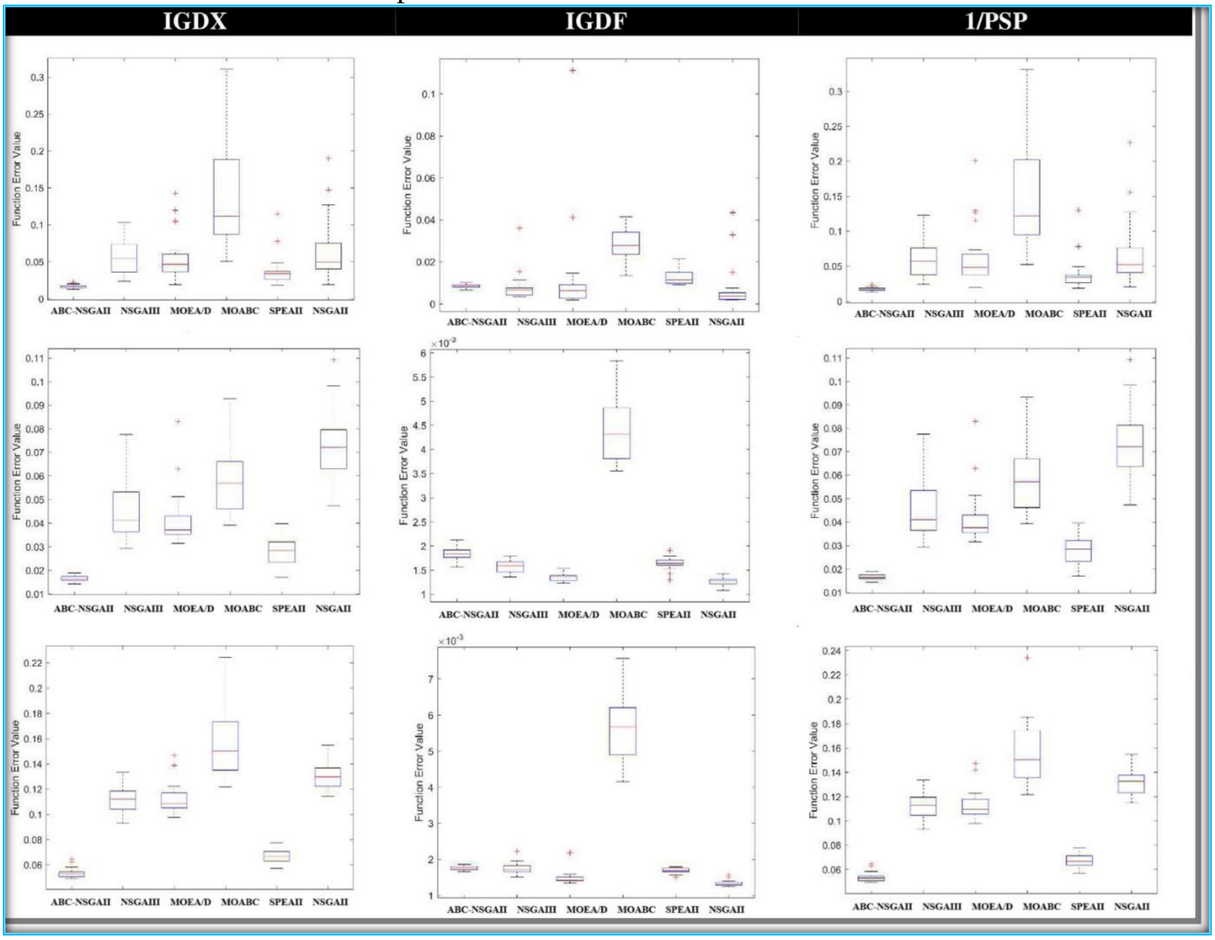
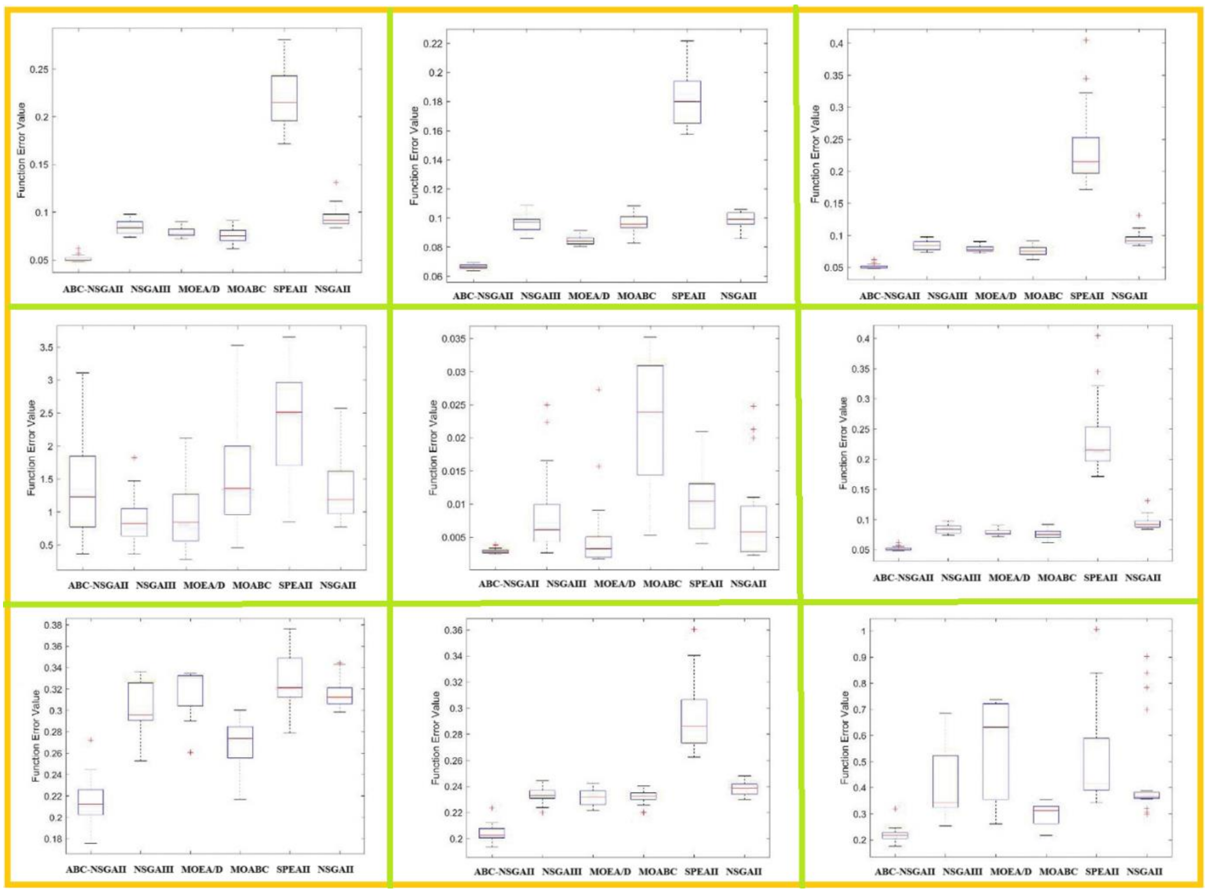


Fig 4: Box Plots of CEC 2024 Benchmark Test Functions

This Fig shows that multi-objective optimisation algorithms need new exploitation and exploration methods. Box-plot graphs show algorithm quality and convergence stability well. Above are approximate graphs for 7 of 15 problems. For the other 12 problems with different PF and PS geometries, Fig. 4 shows the IGDX, IGDF, and 1/PSP algorithm performances. The box-plot graphs in Fig. 4 show that no algorithm outperformed its competitors in all problems. This showed that algorithm performance varied by problem. This showed that these seven problems had unique algorithms. ABC-NSGAII was more stable and had better convergence performance for IGDX, IGDF, and 1/PSP metrics than its competitors in these unique problems. Here we considers 15 problems, 4 metrics, have score for ABC-NSGAII (29/60), NSGAIII (8/60), MOEA/D(1/60), MOABC(0/60), SPEAII(7/60), NSGAII(17/60).



6.1.2. Algorithm complexity

This section discusses algorithm computational complexity. Everything in this process followed CEC 2024 experimental study standards. In CEC 2024, the F1 problem was used for calculations. Number of decision variables (var = 2), local and global Pareto sets (N_ops = 2), convex Pareto front geometry, non-linear Pareto set geometry. Trials ended with maximal fitness evaluations (10000 * N_ops). Every algorithm was tested 21 times. After these explanations, Table 5 shows the average F1 algorithm calculation time in seconds. Table 5 classifies algorithms by computational complexity into two groups. MOPSO, NSGAII, MOABC, and NSGAIII are computationally easy. However, SPEA2 seems computationally difficult. Clustering makes SPEA2's archive handling algorithm more complicated.

Table 5: Average computation times of various algorithms for problem F1 (in seconds)

Algorithm	MOABC	NSGAIII	ABC-NSGAII	NSGAII	MOPSO	SPEA2
Time (s)	7.90	8.57	8.69	13.64	14.95	308.71

7. Conclusions

This study offers two notable contributions to the domain of multi-dimensional meta-heuristic optimisation. The development of the robust ABC-NSGAII algorithm represents a significant enhancement in search performance through the utilisation of a crowding distance-based design. This algorithm's effectiveness was demonstrated through comprehensive experimental trials on 24 benchmark problems from the CEC 2024 suite, recognised for its rigour in multi-objective optimisation. The findings indicated that ABC-NSGAII surpassed other algorithms, exhibiting enhanced convergence on problems characterised by diverse geometries of the Pareto Front (PF) and Pareto Set (PS). The performance was validated through statistical analyses using key metrics, including Hypervolume (HV), Pareto Set Proximity (PSP), Inverted Generational Distance X (IGDX), and Inverted Generational Distance F (IGDF).

The evaluation of the ABC-NSGAII algorithm utilised a thorough methodology, applying it to the IEEE CEC 2024 benchmark test instances for multi-objective optimisation. The structure of the hybrid algorithm, which incorporates an employed bee phase and an onlooker bee phase, improved existing solutions, while the following crossover and mutation phases produced new and diverse solutions. The algorithm, implemented in MATLAB, effectively minimised multiple objectives concurrently, surpassing alternative methods by more effectively satisfying equality and inequality constraints with minimal violations. The algorithm demonstrated robustness through its convergence stability and ability to manage optimal functions within practical constraints. The ABC-NSGAII algorithm demonstrated significant efficacy in multi-objective optimisation research. The solution set presented is the most effective in the literature for multi-objective optimisation problems, markedly decreasing the total cost across all test cases in comparison to alternative algorithms. This enhancement highlights the algorithm's capacity for generating substantial profits and its relevance to diverse optimisation problems.

References:

[1] Mohamed A, Hadi A, Mohamed A, Agrawal P, Kumar A, Suganthan P (2021) Problem definitions and evaluation criteria for the cec 2021 special session and competition on single objective bound constrained numerical optimization. In: Tech. Rep

[2] Mohamed AW, Hadi AA, Agrawal P, Sallam KM, Mohamed AK (2021) Gaining-sharing knowledge based algorithm with adaptive parameters hybrid with iMODE algorithm for solving CEC 2021 benchmark problems. In: 2021 IEEE congress on evolutionary computation (CEC). IEEE, pp. 841–848

[3] Agrawal P, Ganesh T, Oliva D, Mohamed AW (2021) S-shaped and v-shaped gaining-sharing knowledge-based algorithm for feature selection. Appl Intell pp. 1–32

[4] Stanovov V, Akhmedova S, Semenkin E (2021) NL-SHADE-RSP algorithm with adaptive archive and selective pressure for CEC 2021 numerical optimization. In: IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 809–816

[5] Mohamed AW, Hadi AA, Fattouh AM, Jambi KM (2021) Gaining-sharing knowledge-based algorithm with adaptive parameters for engineering optimization. IEEE Access 9:934–946

[6] Mohamed AW, Hadi AA, Fattouh AM, Jambi KM (2021) Evaluating the performance of adaptive gaining sharing knowledge based algorithm on CEC 2020 benchmark problems. In: 2020 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 1–8

[7] Mohamed AW, Hadi AA, Fattouh AM, Jambi KM (2021) L-SHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems. In: 2017 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 145–152

[8] Brest J, Maučec MS, Bošković B (2021) Self-adaptive differential evolution algorithm with population size reduction for single objective bound-constrained optimization: algorithm J21. In: IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 817–824

[9] Brest J, Maučec MS, Bošković B (2020) Differential evolution algorithm for single objective bound-constrained optimization: Algorithm j2020. In: IEEE Congress on Evolutionary Computation (CEC).

[10] Biswas S, Saha D, De S, Cobb AD, Das S, Jalaian BA (2021) Improving differential evolution through Bayesian hyperparameter optimization. In: IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 832–840

[11] Warchulski E, Arabas J (2021) A new step-size adaptation rule for CMA-ES based on the population midpoint fitness. In: IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 825–831

[12] Bujok P, Kolenovsky P (2021) Differential evolution with distance-based mutation-selection applied to CEC 2021 single objective numerical optimisation. In: 2021 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 849–856

[13] Mousavirad SJ, Moghadam MH, Saadatmand M, Chakrabortty RK (2021) An ordered and roulette-wheel-based mutation incorporated L-SHADE algorithm for solving CEC 2021 single objective numerical optimisation problems. In: Proceedings of the Genetic and Evolutionary Computation Conference

[14] Kadavy T, Pluhacek M, Viktorin A, Senkerik R (2020) Self-organizing migrating algorithm with clustering-aided migration. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, pp. 1441–1447

[15] Opara KR, Hadi AA, Mohamed AW (2020) Parametrized benchmarking: an outline of the idea and a feasibility study. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, pp. 197–198
[16] Sallam KM, Elsayed SM, Chakrabortty RK, Ryan MJ (2020) Improved multi-operator differential evolution algorithm for solving unconstrained problems. In: IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 1–8
[17] Awad N, Ali M, Liang JJ, Qu B, Suganthan P (2016) Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization. In: Technical Reports
[18] Awad NH, Ali MZ, Suganthan PN, Reynolds RG (2016) An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC 2014 benchmark problems. In: IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 2958–2965
[19] Yue C, Price K, Suganthan P, Liang J, Ali M, Qu B, Awad N, Biswas P (2019) Problem definitions and evaluation criteria for the cec 2020 special session and competition on single objective bound constrained numerical optimization. Comput Intell Lab, Zhengzhou Univ., Zhengzhou, China, Tech.
[20] Wu G, Mallipeddi R, Suganthan PN (2017) Problem definitions and evaluation criteria for the cec 2017 competition on constrained real-parameter optimization. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report
[21] Beiranvand V, Hare W, Lucet Y (2017) Best practices for comparing optimization algorithms. Optim Eng 18(4):815–848
[22] Cicirelli F, Forestiero A, Giordano A, Mastroianni C (2016) Transparent and efficient parallelization of swarm algorithms. ACM Trans Autonom Adap Syst (TAAS) 11(2):1–26
[23] Mersmann O, Preuss M, Trautmann H, Bischl B, Weihs C (2015) Analyzing the bbob results by means of benchmarking concepts. Evol Comput 23(1):161–185
[24] Muñoz MA, Sun Y, Kirley M, Halgamuge SK (2015) Algorithm selection for black-box continuous optimization problems: a survey on methods and challenges. Inf Sci 317:224–245
[25] Hansen N, Auger A, Ros R, Finck S, Posík P (2010) Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In: Proceedings of the 12th annual conference companion on Genetic and evolutionary computation, pp. 1689–1696
[26] Liang J, Qu B, Suganthan P, Hernández-Díaz AG (2013) Problem definitions and evaluation criteria for the cec, (2013) special session and competition on real-parameter optimization. Technical Report
[27] Tanabe R, Fukunaga A (2013) Success-history based parameter adaptation for differential evolution. In: IEEE Congress on Evolutionary Computation. IEEE, pp. 71–78
[28] Performance Comparison of NSGA-II and NSGA-III on Various Many-Objective Test Problems Hisao Ishibuchi, Ryo Imada Significance of NSGA-III over NSGAII. *2016 IEEE Congress on Evolutionary Computation (CEC), pp-3045-3052*
[29] Beume N, Naujoks B, Emmerich M (2007) SMS-EMOA: Multiobjective selection based on dominated hypervolume. Eur J Oper Res 181(3):1653–1669
[30] Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y-P, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. KanGAL Rep 2005005(2005):2005