Mathematics

AIMS Mathematics, 8(12): 30244–30268.
DOI: 10.3934/math.20231545
Received: 29 July 2023
Revised: 12 October 2023
Accepted: 17 October 2023
Published: 07 November 2023

http://www.aimspress.com/journal/Math

Research article

MOJMA: A novel multi-objective optimization algorithm based Java Macaque Behavior Model

Dinesh Karunanidy[1], Rajakumar Ramalingam[2,*], Shakila Basheer[3], Nandhini Mahadevan[4] and Mamoon Rashid[5,*]

[1] Department of Computer Science & Technology, Madanapalle Institute of Technology & Science, Madanapalle 517325, India
[2] Centre for Automation, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, TamilNadu, India
[3] Department of Information Systems, College of Computer and Information Science, Princess Nourah bint Abdulrahman University, P.O. BOX 84428, Riyadh 11671, Saudi Arabia
[4] Department of Computer Science and Engineering (Data Science), Madanapalle Institute of Technology & Science, Madanapalle 517325, India
[5] Department of Computer Engineering, Faculty of Science and Technology, Vishwakarma University, Pune, 411048, India

Correspondence: Email: ramukshare@gmail.com; mamoon873@gmail.com.

Abstract: We introduce the Multi-objective Java Macaque Algorithm for tackling complex multi-objective optimization (MOP) problems. Inspired by the natural behavior of Java Macaque monkeys, the algorithm employs a unique selection strategy based on social hierarchy, with multiple search agents organized into multi-group populations. It includes male replacement strategies and a learning process to balance intensification and diversification. Multiple decision-making parameters manage trade-offs between potential solutions. Experimental results on real-time MOP problems, including discrete and continuous optimization, demonstrate the algorithm's effectiveness with a 0.9% convergence rate, outperforming the MEDA/D algorithm's 0.98%. This novel approach shows promise for addressing MOP complexities in practical applications.

Keywords: multi-objective optimization; Java Macaque algorithm; nature-inspired optimization; continuous optimization problem; discrete optimization problem
Mathematics Subject Classification: 90C23, 90C29

1. Introduction

The optimization algorithm demonstrates its greater assistance in solving real-world problems. A significant set of optimization algorithms, such as the evolutionary algorithm (EA), physical algorithm, bio-inspired algorithm and swarm intelligence (SI), have been highlighted by the Nature-inspired optimization algorithms. These algorithms exhibit their efficacy in addressing a vast class of real-world issues [1,2]. A variety of nature-inspired optimization algorithms such as Ageist Spider Monkey Algorithm (ASMO) [3], Spider Monkey Algorithm (SMO) [4], Monkey Algorithm (MA) [5], Cuckoo Search (CO) [6], Bat Algorithm [7], Firefly Algorithm (FA) [8], Particle Swarm Optimization (PSO) [9], Artificial Bee Colony (ABC) [10], Ant Colony Optimization (ACO) [11], Differential Evolution (DE) [12], Genetic Algorithm (GA) [13], etc. uses the simple local search based learning procedure to address the complex problem. The improved African Vultures Optimization Algorithm (AVOA) with Quantum Rotation Gate (QRG) and Association Strategy (AS) mechanisms enhances diversity and avoids local optima, showing promise for large-scale problems compared to other optimization algorithms [14]. The study in [15] enhances the Cuckoo Search Optimization (CSO) algorithm by integrating Genetic Algorithms (GA) to address CSO's convergence issues, improving community detection in complex networks. The review highlights in [16,17] their impact on accelerating convergence and explores their applications in various fields. The Spider Monkey algorithm (SMA) [18], inspired by slime mold behavior, is a novel meta-heuristic approach with adaptive weights and strong exploration exploitation capabilities.

Multi-objective optimization (MOP) is the process of finding an optimum solution to an optimization problem with many competing objectives. [19–22]. The Multi-group population-based algorithms have been verified to be significant in investigating and exploring the search space. Community detection in complex networks aims to locate nodes clustered together with strong internal links but weak external connections. Various Harris Hawk Optimization (HHO) algorithm methods can address challenges in social network community detection by balancing exploitation and exploration [23]. The operations, such as the number of sub-populations, communication and strategy among the sub-populations, are having serious impacts by utilizing the multi-group populations [23,24]. First, the quantity of sub-populations contributes to the diverse occupation of the entire search space. In reality, a limited number of sub-populations leads to local optima, whereas a larger number of sub-populations wastes computational resources and delays convergence [25]. The subsequent essential factor is the sub-population communication management procedure, including interaction frequency and strategy [26–28]. The interaction rate specifies the number of individuals who are part of a sub-population that must interact with individuals of other sub-populations, and the rule of communication is subject to the replacement of individuals with individuals of other sub-populations.

The majority of optimization algorithms encounter adaptive search operation to maintain diversity in the population. However, the algorithm's reliance on multiple populations illustrates its effectiveness in preserving a diverse population. However, it also depends on consideration of the sub-population's communication strategy. The [29] introduces the MAMH method, which combines metaheuristic algorithms within a multi-agent system to address complex high-dimensional problems, enhancing population diversity and convergence rates. The constraints of the existing literature inspire us to formulate the well-balanced exploration and exploitation strategy to solve the multi-objective optimization problem. Java-macaque monkeys are widespread primitives in South-Asian countries and exhibit dominant properties that suit well for directing real-world problems via the exploration and exploitation phases.

The proposed optimization algorithm that leverages the behavioral patterns observed in Java macaque monkeys is used to enhance the search operation's balance [30]. These macaques exhibit intricate social behaviors, which we have successfully modeled. Our algorithm incorporates elements such as cooperation among groups, competition among search agents, a selection strategy based on social hierarchy, breeding, male rehabilitation and learning. To address the complexity of multi-objective problems, our algorithm utilizes a multi-group population approach, ensuring diversity among the generated solutions. The algorithm employs a mating process that takes into account the dominance hierarchy, allowing it to explore complex search spaces effectively. In resolving communication and adaptive search issues, our algorithm draws inspiration from the male replacement mechanism observed in Java macaque monkeys. The exploitation phase of our proposed algorithm is achieved through a learning process. To maintain an elite solution within the population, our Java macaque optimization algorithm employs a multi-leader mechanism, utilizing the alpha male and female of each group. This multileader strategy facilitates the transition from exploration to exploitation, driven by a selection strategy based on social hierarchy.

The major contribution of this work is highlighted below:

• Develop and implement the Multi-objective Java Macaque Algorithm, a nature-inspired optimization approach, to address the complexities of multi-objective optimization problems.

• Investigate the effectiveness of the proposed algorithm in handling local optima and imbalanced selection strategies by analyzing its performance on real-time optimization problems in both multi-objective discrete and continuous optimization scenarios.

• Evaluate the unique selection strategy based on a social hierarchy process, male replacement strategies and learning processes incorporated in the algorithm to strike a well-balanced trade-off between intensification and diversification within the search boundary.

• To assess the impact of multiple decision-making parameters introduced in the algorithm to manage trade-offs among various potential solutions, aiming for enhanced convergence rates and diversified outcomes in comparison to existing algorithms.

Furthermore, section 2 of this paper deliberates relevant information on MOP, and section 3 briefly discuss the behavior of java macaque monkeys towards MOP. Then, in section 4, the model of java macaque algorithms of multi-objective problems are briefly discussed. In section 5, the experimentation of discrete optimization is illustrated in detail, whereas section 6 provides a concise demonstration over the continuous problems. Finally, section 7 provides the paper's conclusion.

2. Related works

The real-world optimization problems require the optimal fine tuning between two or more contradictory objectives. Then, the problem of multi-objective optimization is represented as:

$$Minimize \quad F(\overleftarrow{\varphi}) = [f_1(\overleftarrow{\varphi}), \dots, f_m(\overleftarrow{\varphi})]^T \tag{2.1}$$

$$s.t \quad \overleftarrow{\varphi} \in \mathbb{X}$$

In the MOP, let us consider the following notation and description: $F(\overleftarrow{\varphi})$ signifies the objective function of a solution in relation to the multi-objective optimization issue. $\mathbb{X}$ denotes the decision space, which encompasses the possible solutions or decision variables. $F: \mathbb{X} \to \mathbb{R}^m \text{or} \mathbb{Z}^m: \overleftarrow{lb_\iota} \leq \overleftarrow{\varphi_\iota} \leq \overleftarrow{ub_\iota}$, $\forall i = 1,2,..,m$ represents the set of m real numbered objective functions.

Let the dominance between the two solutions φ and φ^* described as,

$$(f_i(\overleftarrow{\varphi^*})) \leq (f_i(\overleftarrow{\varphi})), \forall_{i\in I} \tag{2.2}$$

$$s.t \quad \varphi^*, \varphi \in \mathbb{X}, I = \{1,2,\ldots,m\} \tag{2.3}$$

and there is at least one $i \in I$ such that,

$$(f_i(\overleftarrow{\varphi^*})) < (f_i(\overleftarrow{\varphi})) \tag{2.4}$$

When $\overleftarrow{\varphi^*}$ is dominated by $\overleftarrow{\varphi}$, the fitness value of each individual must be $\geq$ one objective to another objective. Conflicting objectives are a common feature of multi-objective optimization problems, and simultaneously achieving all objectives results in the best solution. In particular, one solution might outperform another on one objective but not on another. As a result, both solutions are considered dominant.

In MOP, the primary indention is not to determine a single optimal solution, but to explore and present the trade-off between conflicting objectives. The Pareto front consists of all the solutions that do not simultaneously dominate any other solution in terms of all the objectives. The Pareto optimal solution is denoted as:

$$P^* := \{\varphi \in \mathbb{X} | \neg\exists\varphi' \in \mathbb{X}, (F(\overleftarrow{\varphi'})) \preceq (F(\overleftarrow{\varphi}))\} \tag{2.5}$$

The traditional method used by evolutionary algorithms (EA) to select the dominant solution is random selection. However, this approach faces significant challenges due to the proportional increase in objectives. Moreover, a large portion of the population is occupied by dominated solutions, thus circuitously limits the search space for yielding new and novel solutions [31,32].

In order to address these challenges, diversity preservation has emerged as a crucial aspect of EA for multi-objective optimization problems (MOP) [33,34]. The exploration process acts a key role in locating the optimal solution, which generated the distributed solution over the problem space. Various techniques have been proposed for conserving biodiversity, including dominance-based methods [35], indicator-based methods [36] and decomposition-based methods [37]. However, these techniques often come with the drawback of increased computational complexity.

Among the popular techniques for maintaining genetic diversity in the population are domination-based approaches like NSGA II [38] and decomposition-based approaches like MOEA/D [39]. However, the MOEA/D-generated approximation set of Pareto fronts can become complex due to the fronts' sensitive geometries. Tchebychef (TCH), Penalty-based Boundary Intersection (PBI) and Weighted Sum (WS) methods are prevalent decay techniques for handling multi-objective problems.

Several techniques, such as Constrained Decomposition Approach with Grids [40], reference vector guided evolutionary algorithm [41], NSGA III utilizing reference point [42] and Decomposition based reference points [36,43] have been introduced in the literature to address this issue. However, the majority of evolutionary algorithms struggle to reconcile exploration and exploitation. The Java macaque algorithm, on the other hand, has been extensively investigated and shown to efficiently maintain this equilibrium.

The optimization algorithm demonstrates its greater assistance in solving real-world problems. A significant set of optimization algorithms, such as the evolutionary algorithm (EA), physical algorithm, bio-inspired algorithm and swarm intelligence (SI) have been highlighted by the Nature-inspired optimization algorithms. These algorithms exhibit their efficacy in addressing a vast class of real-world issues [1,2].

Recent researchers [36,43,42,40] in multi-objective optimizations (MOP) have proposed a technique that utilizes ideal and nadir points to subdivide each objective and enhance the performance of the search process.

In Figure 1, the ideal point, denoted as Z^{ideal}, and the nadir point, denoted as Z^{nadir}, on the objective vector are represented as:

$$Z^{ideal} = min\{f_i(\overleftarrow{\varphi})\}, \quad \forall\overleftarrow{\varphi} \in \mathbb{X}, i = 1,2,..,m \tag{2.6}$$

$$Z^{nadir} = max\{f_i(\overleftarrow{\varphi})\}, \quad \forall\overleftarrow{\varphi} \in PS, i = 1,2,..,m \tag{2.7}$$

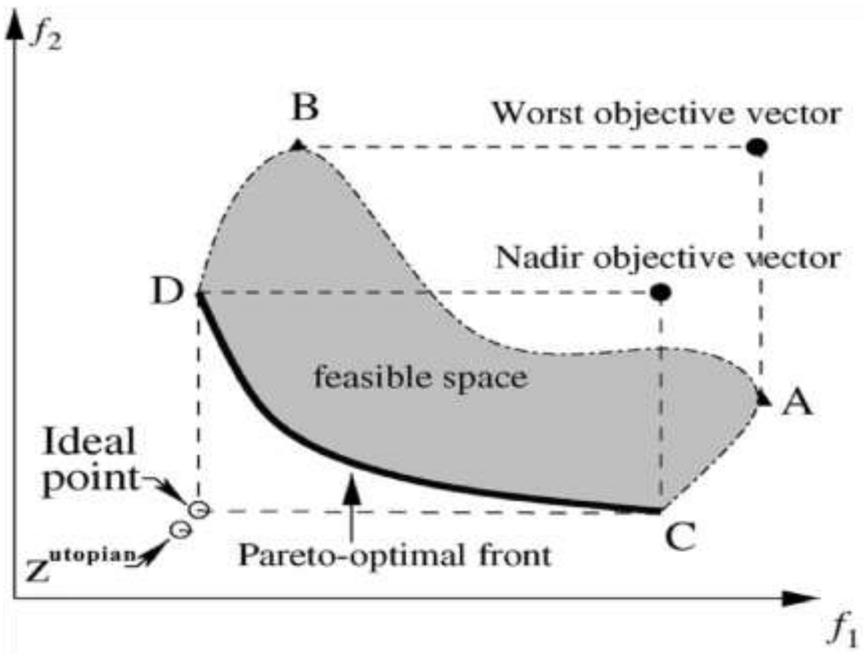

Figure 1. Ideal, Nadir and Utopian Points [11].

Hence, the search space is divided into multiple sub-spaces using reference points. In a two-objective optimization problem, for illustration, there would be 10 reference points used for decomposition. The Pareto fronts prevailed using the TCH decomposition approach for both the convention and scaled objective functions are shown in Figure 2. However, when dealing with the inseparable scaled objective function, it is evident from the figure that the existing TCH decomposition approach fails to explore the entire search space effectively. In particular, when f2 function is scaled five times associated with f1 function, the TCH-based decomposition approach is impotent to adequately delve into the search limit. To address this issue, the grid-based decomposition technique, as depicted in Figure 3, is employed. Therefore, in order to maintain the quality of Pareto fronts, the JMA also needs to incorporate this significant technique.

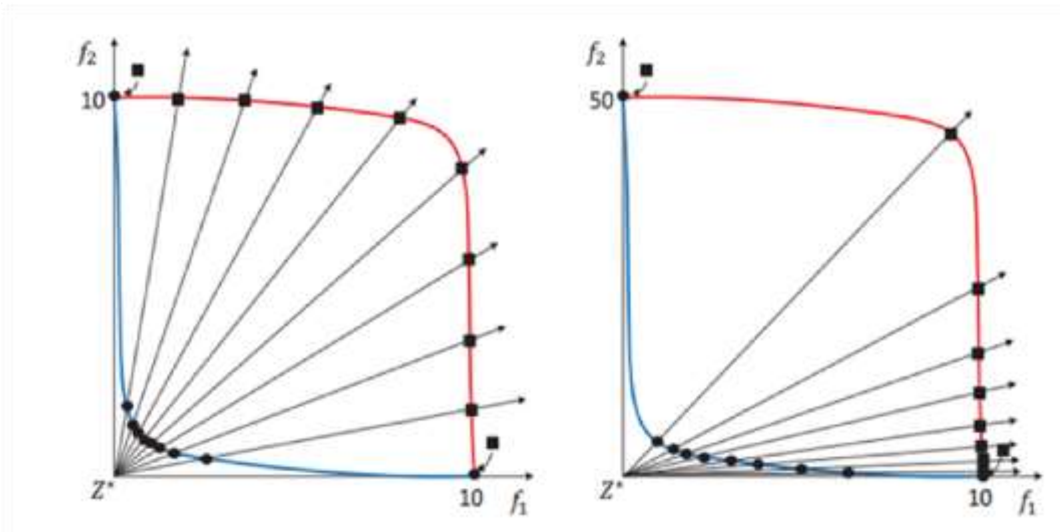

Figure 2. Standard and scaled objective of TCH decomposition [43].

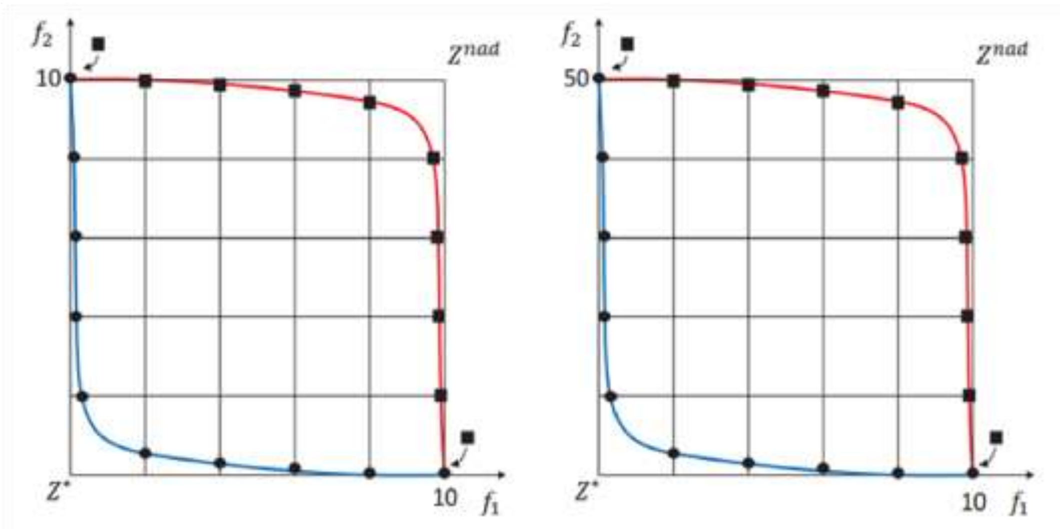

Figure 3. Standard and scaled objective of Grid based decomposition Approach [40].

3. Multi-objective JMA approach

The Java Macaque method effectively tackles the optimization problem illustrated in [30] using the social and biological behavior of Java macaques. Further, the java macaque algorithm shows the dominance in solving the optimization problem by ensuring a diverse population and consistently generating high-quality solutions. We may utilize Java monkey tactics to tackle multi-objective optimization issues by analyzing their behavior, which entails a careful balance of intensification and diversification. As a result, Java monkey behavior may be used as a paradigm for handling multiobjective optimization issues, allowing us to adopt their effective strategy of balancing intensification and diversity into our problem-solving strategies.

In the context described, each individual in an optimization problem with various decision variables of vector space n represented as follows:

$$\overleftarrow{\varphi} = \{\varphi_1, \varphi_2, \dots, \varphi_n\}, \quad \overleftarrow{\varphi} \in \mathbb{X} \tag{3.1}$$

Thus, the $\overleftarrow{\varphi}$ is a vector with n decision parameters in the problem space $\mathbb{X}$. Each decision variable φ can take on real values $\mathbb{R}^m$ or integer values $\mathbb{Z}^m$.

The initial population is rendered by selecting a set of female and male solution from every group as follows:

$$POP = \{\overleftarrow{\varphi_1}, \overleftarrow{\varphi_2}, \dots, \overleftarrow{\varphi_{NOI}}\} = \{G_i\}_{i=1,2,\dots,g} = \{M_j\}_{j=1,2,\dots,M_{size}} \cup \{F_k\}_{k=1,2,\dots,F_{size}} \tag{3.2}$$

Here, POP represents the population, $\overleftarrow{\varphi_i}$ represents an individual, G_i represents group i and M_{size} and F_{size} represent the number of male and female individuals, respectively. The condition $(G_i \supset M, F: M \notin F)$ ensures that each group contains both males and females.

For each individual, the fitness value is determined by its performance over the problem space. The fitness value of a solution $\overleftarrow{\varphi}$ is computed using multiple objectives, as shown below:

$$Minimize \quad F(\overleftarrow{\varphi}) = [f_1(\overleftarrow{\varphi}), \dots, f_m(\overleftarrow{\varphi})]^T \tag{3.3}$$

where, the fitness value is a vector containing m objective, where the objective function $f_i(\overleftarrow{\varphi})$ represents a different aspect of the problem. The problem restraint is denoted as $\overleftarrow{\varphi} \in \mathbb{X}$, indicating that the individual $\overleftarrow{\varphi}$ must satisfy certain conditions specified by the problem's feasible space $\mathbb{X}$.

The fitness function $F(\overleftarrow{\varphi_j})$ is assessed for all individual $\overleftarrow{\varphi_j}$ in the male group $\{M\}_i$ of group $\{G_i\}$, and the alpha male AM_i is determined as the solution with the minimum fitness value among all males in the group. Similarly, the fitness function is measured for all individual $\overleftarrow{\varphi_j}$ in the female group $\{F\}_i$ of group $\{G_i\}$, and the alpha female AF_i is determined as the solution with the minimum fitness value among all females in the group. Additionally, the global best solutions from the set of $\{AF\}$ and $\{AM\}$ are chose by taking the minimum fitness value among all alpha males and alpha females in the population (POP):

$$[GM, GF] = \min\{\{AM\}, \{AF\}\} \tag{3.4}$$

where, the global best solutions GM and GF among all individuals in the population $\mathbb{X}$. Furthermore, male and female solutions within each group are divided into DS and NDS based on their fitness values. The decomposition is performed as follows:

$$\overleftarrow{\varphi^*}, \overleftarrow{\varphi} \in \{M_i\}, \text{then} = \begin{cases} \overleftarrow{\varphi^*} \in \{NDS\}, & if F(\overleftarrow{\varphi^*}) < F(\overleftarrow{\varphi}) \\ \overleftarrow{\varphi^*} \in \{DS\}, & otherwise. \end{cases} \tag{3.5}$$

Further, the mechanism is opted for female population in the group. The process of mating is a crucial search mechanism employed by the JMA to explore the problem space denoted as $\mathbb{X}$. Mating occurs between a male $\overleftarrow{\varphi^m}$ and a female solution $\overleftarrow{\varphi^f}$, both of which belong to either the set {NDS} or {DS}. The objective of mating is to generate new offspring $\overleftarrow{\varphi_{off}}$ according to the following equation:

$$\overleftarrow{\varphi^{off}} = \text{Mating}(\overleftarrow{\varphi^m}, \overleftarrow{\varphi^f}) \quad \text{s.t.} \quad \overleftarrow{\varphi_m}, \overleftarrow{\varphi_f} \in G_i \tag{3.6}$$

For discrete optimization problems, the uniform crossover operator is employed. This operator was introduced by Syswerda in 1989 and has been widely used in the field [44,45]. On the other hand, for continuous optimization problems, the simulated binary crossover operator is utilized. This operator was proposed by Deb and has since been utilized in various studies and algorithms [46,47,48].

After the learning process, the age of the Infant Female (IF) and Male (IM) is set to 0, indicating that they have completed their learning and are now ready to improve their fitness. When individuals in each group $\{G_i\}$ reach sexual maturity $\mathbb{S}$, certain changes occur. The subadult male, referred to as the Stray Male, is compelled to impart the own group and seek another group within the same environment. The subadult female, on the other hand, remains in the original group. For the Stray Male, the adaptive search process involves finding a suitable group and establishing an optimal position within that group. Thus, the Stray Male denoted as $\overleftarrow{\varphi_{sm}}$ replaces the week solution from the non-dominated set through a process called male replacement. The male replacement is defined as follows:

$$\text{M}aleReplacement = \begin{cases} REPLACE(\overleftarrow{\varphi^{sm}}, \overleftarrow{\varphi^{m}}), & \text{if} F(\overleftarrow{\varphi^{sm}}) < F(\overleftarrow{\varphi^{m}}) \\ \overleftarrow{\varphi^{sm}} \in \{SM\}, & \text{otherwise.} \end{cases} \tag{3.7}$$

In the above equations, $F(\overleftarrow{\varphi^{sm}})$ and $F(\overleftarrow{\varphi^{m}})$ represent the fitness values of the Stray Male and the non-dominated male, respectively. If the fitness of the Stray Male is lower than the non-dominated male, the non-dominated male is replaced by the Stray Male using the REPLACE operation. Otherwise, the Stray Male remains in the set $\{SM\}$.

Algorithm 1: Population Sorting using Utopian-based Grid Approach.

Input: $\{G_i\}$: Group i of the population, Act_{size}: denotes the umber of solutions in Group i
Z^{ideal}, Z^{nadir}, $Z^{utopian}$: Ideal, nadir and Utopian points, m: Total objectives
Output: $\{G_i\}$, POP
BEGIN
/* Determine the total of solutions in $\{G_i\}$ does not exceed the limit.*/
Redo
if $|G_i| > Act_{size}$ **then**
/* Processing the Grid Approach */
$j \leftarrow 1$
Redo

$$I_j \leftarrow (Z_j^{nadir} - Z_j^{utopian} + 2 \times v)/\,k$$

Till $j \leq m$
$j \leftarrow 1$
Repeat

$$g_j(\varphi) \leftarrow (f_j(\varphi) - Z_j^{utopian} + v)/I_j$$

Till $j \leq Act_{size}$
$j \leftarrow 1$
Redo
$Z^{ideal} \leftarrow \min\{f_i(\overleftarrow{\varphi})\}$
$Z^{nadir} \leftarrow \max\{f_i(\overleftarrow{\varphi})\}$

$Z^{utopian} \leftarrow Z^{ideal} - c$

Till $j \leq m$

$S(\overleftarrow{\varphi}) \leftarrow (s_1(\varphi), \ldots, s_m(\varphi))$

$j \leftarrow 1$

Redo

$S^*(\overleftarrow{\varphi}) \leftarrow \text{SortAscending}(S(\overleftarrow{\varphi}))$

Till $j \leq m$

$j \leftarrow 1$

Redo

$\{NG_i\} \leftarrow \text{LexicographicSort}(R(\overleftarrow{\varphi}))$

Till $j \leq Act_{size}$

$\{G_i\} \leftarrow NG_i(1{:}N)$

END

Till $i \leq g$

$POP \leftarrow \{G_i\}_{i=1,2,\ldots,g}$

END

It is important to note that $\overleftarrow{\varphi^{sm}}$ belongs to one of the groups $\{G_i\}_{i=1,2,\ldots g}$, and $\overleftarrow{\varphi^{m}}$ belongs to the set of non-dominated males $\{NDS\}$. However, the group to which $\overleftarrow{\varphi^{m}}$ belongs, denoted as $\{G_j\}_{j=1,2,\ldots g}$, must be different from the group to which $\overleftarrow{\varphi^{sm}}$ belongs.

Learning is an important aspect of JMA that aids in the search for the discrete problem referred to as below.

$$Learning = \{PoP, \mathbb{G}, \overleftarrow{\delta}, L(\overleftarrow{\varphi^k}), F(\overleftarrow{\varphi^k}), \mathbb{X}\} \tag{3.8}$$

Here, PoP indicates the population, $\mathbb{G}$ is the problem space, $\overleftarrow{\delta}$ denotes the learning rate of an solutions (decreasing linearly from 1 to 0), $L(\overleftarrow{\varphi^k})$ represents learning process of an individuals.

The learning process for an individual $\overleftarrow{\varphi^k}$ in a discrete optimization problem is defined using four cases:

$$-[i, j] = \text{sort}[\lceil x * \text{rand}(\delta, 2)\rceil] \tag{3.9}$$

In this equation, x is a linearly decreasing value generated between the maximum value (max $\leftarrow$ $\mathbb{X}$) and 3. The values u and v are randomly selected.

The four variants for the learning process are as follows:

$$Learning_1(\varphi^k, [u, v]) = Learning(\varphi^k, [u, v]) \tag{3.10}$$

$$Learning_2(\varphi^k, u, v) = Learning(\varphi^k, v{:} -1{:} u) \tag{3.11}$$

$$Learning_3(\varphi^k, u, v) = Learning(\varphi^k, [u + 1{:} v]) \tag{3.12}$$

$$Learning_4(\varphi^k, u, v) = Learning(\varphi^k, u, v) \tag{3.13}$$

The best solution produced from these learning processes is denoted as φ^* and is replaced in the population PoP. Similarly, to solve the continuous optimization problem involves the following steps:

$$\overleftarrow{L_1}(\overleftarrow{\varphi^{GM}}, \overleftarrow{\varphi^k}) = \overleftarrow{\varphi^{GM}} - (2 \cdot \overleftarrow{\delta} \cdot \overleftarrow{r_1} - \overleftarrow{\delta})|2 \cdot \overleftarrow{r_2} \cdot \overleftarrow{\varphi^{GM}} - \overleftarrow{\varphi^k}| \tag{3.14}$$

$$\overleftarrow{\varphi^k} = \frac{\overleftarrow{L_1}+\overleftarrow{L_2}+\overleftarrow{L_3}+\overleftarrow{L_4}}{4} \tag{3.15}$$

In these equations, $\overleftarrow{\varphi^{GM}}$ represents the global best female, $\overleftarrow{\varphi^{GF}}$ represents the global best female, $\overleftarrow{\varphi^{AM}}$ represents the alpha male, $\overleftarrow{\varphi^{AF}}$ represents the alpha female and $\overleftarrow{r_1}$ and $\overleftarrow{r_2}$ are random vectors between 0 and 1. After performing the learning process, the individual $\overleftarrow{\varphi^k}$ is modified and substituted in the population PoP.

Population sorting using Utopian based Grid Process

The JMA is a powerful approach utilized for tackling MOPs, aiming to achieve a balance between exploration (diversity) and exploitation (convergence) during the search process. The algorithm's effectiveness relies on its mating and male replacement processes, which play pivotal roles in the search procedure. Moreover, the JMA incorporates a technique for preserving non-dominated solutions, known as the Pareto front, within the population. The optimization process removes solutions when the number of non-dominated solutions exceeds the population size to maintain a constant population size (i.e., |POP|>NOI).

Maintaining a well-converged and diverse Pareto front in the succeeding generation poses a significant challenge in multi-objective optimization. To address this issue and preserve the population, the conventional lexicographic-based ranking technique has been enhanced by introducing idealistic points. The Utopian point is obtained by subtracting a small value from the ideal point (Z^{ideal}). This expansion of the search space in the bounded region is illustrated in Figure 4 and can be represented by the equation:

$$Z^{utopian} = Z^{ideal} - c \quad \text{where c is a small constant} \tag{3.16}$$

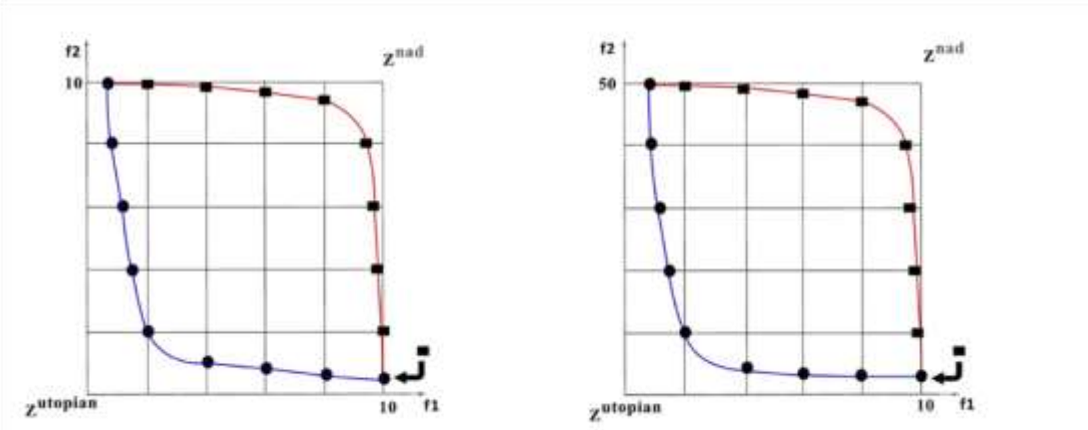

Figure 4. Standard and scaled objective function based on Utopian Points based grid decomposition Approach.

The individual goal (j) is then divided into k portions between the nadir and Utopian ranges. In this case, k denotes the sequence of objectives. The subdivision is calculated as follows:

$$I_j = \frac{(Z_j^{nadir} - Z_j^{utopian} + 2\nu)}{k} \tag{3.17}$$

where ν represents diminished positive value. The grid location $(g_j(\varphi))$ of an individual φ along the jth objective is determined using the equation:

$$g_j(\varphi) = \frac{(f_j(\varphi) - Z_j^{utopian} + \nu)}{I_j} \tag{3.18}$$

The rank-based selection in JMA involves assigning ranks to individuals in the population. Each individual is assigned m ranks, and the rank vector is denoted as:

$$S(\overleftarrow{\varphi}) = (s_1(\varphi), \dots, s_m(\varphi)) \tag{3.19}$$

The solutions of the population are then classified in ascending order as per rank vector:

$$S^*(\overleftarrow{\varphi}) = SortAscending(S(\overleftarrow{\varphi})) \tag{3.20}$$

This ranking procedure serves the purpose of structuring the population and simplifying the retention of superior solutions throughout the optimization process. Ultimately, the population undergoes lexicographic sorting, enabling the selection of essential individuals for the succeeding generation.

4. Proposed JMA for MOP

In this work, a novel approach namely Java Macaque algorithm is introduced to address multi-objective optimization problems by integrating exploration and exploitation strategies to achieve optimal search performance. The detailed procedure of Java Macaque algorithm is presented in ref. [30]. The algorithm initiates by generating an initial population, denoted as POP, through random population seeding, where each individual $\overleftarrow{\varphi}$ is mapped $\mathbb{G}$ to the $\mathbb{X}$. The fitness function $F(\overleftarrow{\varphi})$ is employed to evaluate individuals within each group.

Subsequently, the population is divided into two groups: Alpha Males (AM) and Alpha Females (AF). The best male and female individuals, represented by GM and GF, respectively, are selected based on their fitness values and categorized into DS and NDS individuals. The mating process, a pivotal aspect of the algorithm, is then carried out according to the social hierarchy, where male and female individuals, φ_m and φ_f, respectively, are chosen from either the DS and NDS sets.

The new solution resulting from mating are referred to as infant males (φ^{IM}) or infant females (φ^{IF}) depending on their gender. When a male offspring reaches the age of 1 (pull through one generation), it becomes a juvenile male (φ^{JM}). Similarly, when a female offspring reaches the age of 1, it becomes a juvenile female (φ^{JF}). At the age of 4, the juvenile male reaches sexual maturity and gains the social status of Stray Male (φ^{SM}). Likewise, a female offspring reaches sexual maturity at age 3 and becomes a subadult female (φ^{SF}). The subadult female then matures into an adult female (φ^{AF}), and it remains in the group for 5 generations. However, the φ^{SM} or subadult male is enforced to vacate the natal group. The male replacement procedure is employed, allowing the stray male to explore the search space and replace the dominant male in the group.

The JMA's learning strategy efficiently intensifies the search limit, enabling the identification of the global best solution for continuous optimizing issues. The individual's learning rate, represented as $\overleftarrow{\delta}$, linearly decreases from $(0 \leq \overleftarrow{\delta} \leq 1)$. During the learning process, for an individual $\overleftarrow{\varphi_k}$ with fitness value $F(\overleftarrow{\varphi})$, a compromise between $\overleftarrow{\varphi_k}$ and the global best individual $\overleftarrow{\varphi_{GM}}$ is computed with the aid of two arbitrary vectors ranging from 0 to 1. Similarly, the compromise between the individual and the global best female $\overleftarrow{L_2}(\overleftarrow{\varphi_{GF}}, \overleftarrow{\varphi_k})$ is determined. The same method is applied to calculate the compromise between the individual and the alpha male $\overleftarrow{L_3}(\overleftarrow{\varphi_{AM}}, \overleftarrow{\varphi_k})$, and similarly for $\overleftarrow{L_4}(\overleftarrow{\varphi_{AF}}, \overleftarrow{\varphi_k})$ within the individual's natal group. The individual then adjusts its position using Eq 3.8. Finally, a grid-based lexicographic sorting approach is employed as the primary process for maintaining a well-distributed population in the multi-objective optimization algorithm.

In summary, the Java Macaque algorithm is a multi-objective optimization approach that combines exploration and exploitation strategies. It utilizes social hierarchy, mating processes and a learning strategy to efficiently explore the search space and find global optimal solutions. The algorithm maintains a well-distributed population sorting using grid-based lexicographic approach.

5. Analysis of experiments for a discrete multiobjective problem

The proposed MOJMA is thoroughly investigated in this paper using the famous Traveling Salesman Problem (TSP) from TSPLIB. The investigation is carried out utilizing the existing test-bed stated in [49], in which the effectiveness of MOJMA is compared to that of MEDA/D. Following the standard procedure detailed by Karshenas and colleagues, 50 independent simulations are performed for each method, with a consistent population size of 250 and termination criterion set to 1000 generations. The convergence metrics employed in the MEDA/D algorithm are adopted as performance measures to ascertain the significance of MOJMA's proposed approach. Specifically, in MEDA/D, the parameter "alpha" governing the balance between a priori and learned information, is set to 3, while other parameters are determined based on existing literature. This meticulous evaluation provides valuable insights into MOJMA's capabilities and highlights its potential as a promising alternative for solving discrete optimization issues such as TSP.

5.1. Experimental results

The experiments conducted in this study demonstrate the effectiveness of the proposed MOJMA when compared to the existing Multiobjective Evolutionary Algorithm based on Decomposition (MEDA/D). A random technique was used to generate the population's initial set of individuals. The experimental results, as presented in Table 1, show that the MOJMA consistently outperforms MEDA/D according to various evaluation criteria. For instance, when considering the KroAB100 instance, the MOJMA achieved a convergence rate of 0.999, whereas MEDA/D reached only 0.998. Similarly, in the case of the KroAD100 instance, the MOJMA maintained its superiority with a mean convergence value of 0.97, while MEDA/D obtained 0.999.

In the KroBC100 instance, both MEDA/D and the proposed MOJMA produced identical results for the best and average convergence values. However, the MOJMA outperformed MEDA/D in terms of worst convergence. Conversely, in the KroBE100 instance, MEDA/D dominated the results with a best, worst and average convergence value of 0.997, 0.996 and 0.996, respectively. Although MEDA/D performed better in terms of best and average convergence values (0.998 for both), both methods achieved the same worst convergence value. Specifically, Table MOPTSP reveals that the best, average and worst values for MEDA/D are 1, 1 and 0, respectively, while MOJMA achieved 1, 0.999 and 0.998 for mean, worst and average values. Based on these observations, the results provide substantial evidence of the superiority of the MOJMA over the MEDA/D in terms of convergence performance. The outcome analysis for the best, average, and worst convergence is shown in Figures 5, 6, and 7 respectively.

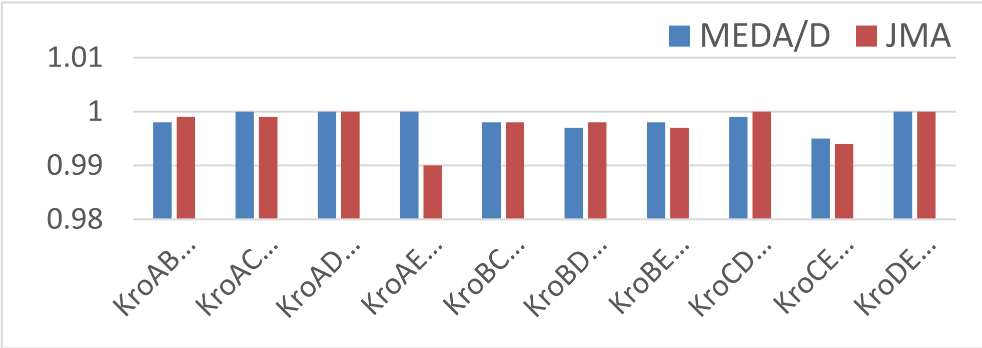

Figure 5. Outcome analysis in terms of Best Convergence.

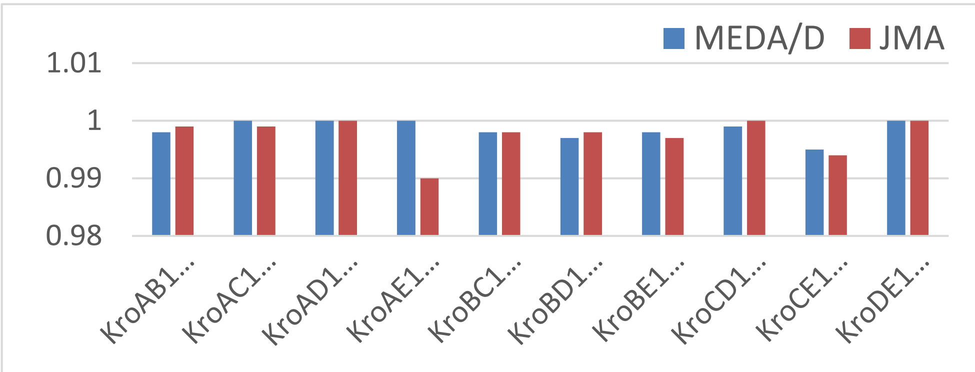

Figure 6. Outcome analysis in terms of average convergence.

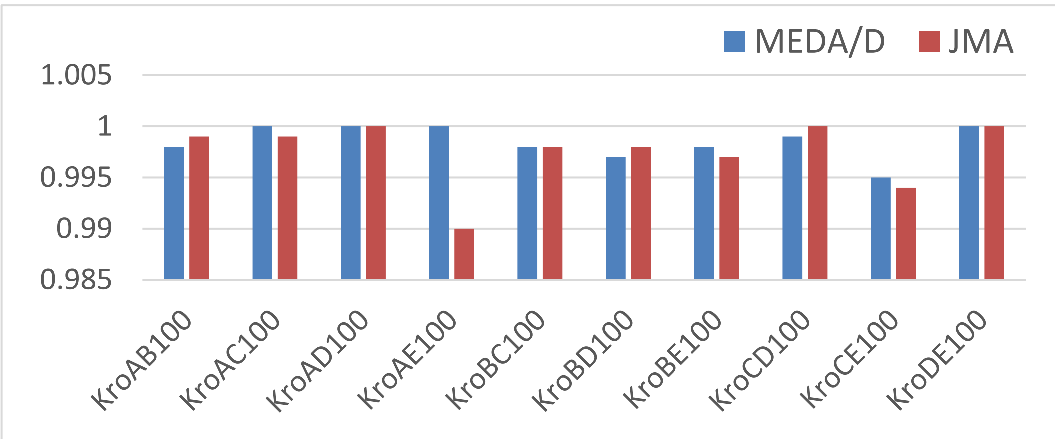

Figure 7. Outcome analysis in terms of Worst Convergence.

The MOJMA as a new approach and compares it with the existing Multiobjective Evolutionary Algorithm with Decomposition (MEDA/D). Convergence is identified as the main evaluation metric for measuring the effectiveness of both algorithms. The results are presented in illustrations and tables, demonstrating the superior performance of MOJMA in terms of optimal convergence values for certain instances like KroAC100, KroAE100 and KroBE100, while MEDA/D performs better in instances such as KroAB100, KroBD100 and KroCD100. For cases where MOJMA and MEDA/D achieve the same convergence values, the analysis suggests that the two algorithms are independent of each other. Moreover, we explore the average convergence of the population and reveal that MOJMA generally outperforms MEDA/D, except for specific instances like KroAE100, KroBE100 and KroDE100. Overall, the experimental results validate the effectiveness of MOJMA over MEDA/D, showcasing its superiority in terms of convergence metrics, for instance, achieving a maximum convergence of 0.999 for KroAB100 compared to MEDA/D's 0.998 and a mean convergence of 0.97 for KroAD100 while MEDA/D reaches only 0.999. The initial population generation used a random technique, and the proposed strategy demonstrates better performance across various evaluation criteria as depicted in Table 1.

Table 1. Result for Multiobjective TSP.

S.NO	Instance	Technique	Best	Mean	Worst
1	KroAB100	MEDA/D	9.98E-01	9.90E-01	9.70E-01
		JMA	**9.99E-01**	**9.98E-01**	**9.90E-01**
2	KroAC100	MEDA/D	**1.00E+00**	**9.99E-01**	9.95E-01
		JMA	9.99E-01	**9.99E-01**	**9.99E-01**
3	KroAD100	MEDA/D	**1.00E+00**	**1.00E+00**	**1.00E+00**
		JMA	**1.00E+00**	**1.00E+00**	**1.00E+00**
4	KroAE100	MEDA/D	**1.00E+00**	**9.99E-01**	**9.94E-01**
		JMA	9.90E-01	9.70E-01	9.53E-01
5	KroBC100	MEDA/D	**9.98E-01**	**9.95E-01**	9.85E-01
		JMA	**9.98E-01**	**9.95E-01**	**9.87E-01**
6	KroBD100	MEDA/D	9.97E-01	9.96E-01	9.86E-01
		JMA	**9.98E-01**	**9.97E-01**	**9.90E-01**
7	KroBE100	MEDA/D	**9.98E-01**	**9.98E-01**	**9.94E-01**
		JMA	9.97E-01	9.96E-01	**9.94E-01**
8	KroCD100	MEDA/D	9.99E-01	9.97E-01	9.84E-01
		JMA	**1.00E+00**	**9.98E-01**	**9.93E-01**
9	KroCE100	MEDA/D	9.94E-01	9.92E-01	9.81E-01
		JMA	**9.95E-01**	**9.93E-01**	**9.85E-01**
10	KroDE100	MEDA/D	**1.00E+00**	**1.00E+00**	**9.98E-01**
		JMA	**1.00E+00**	9.99E-01	**9.98E-01**

6. Experimentation and result analysis for multiobjective continuous problem

Using a typical benchmark function, the proposed JMA was assessed for its potential in addressing multi-objective continuous optimization problems. The experimentation includes employing a set of ten benchmark functions taken from the [50] for the evaluation. The benchmark functions were categorized into bi-objective and tri-objective problems. To assess the performance of the JMA, it was compared with several other algorithms: CDG (Constrained Decomposition Approach with Grids) [40], RVEA (Reference Vector Guided Evolutionary Algorithm) [41], NSGA III [42] and MOEA/D-DE TCH [29].

6.1. Parameter settings

For the bi-objective benchmark functions, the primary population (NOI) was held to 300 individuals. For the tri-objective benchmark functions, the primary population fixed to 600. The JMA and CDG employed a grid distribution with a value of 180 for bi-objectives and a value of 30 for tri-objectives. Sub-problem sizes of 600 for two objectives and 1000 for three objectives were implemented in MOEA/D-DE TCH. The parameters for the neighbor selection probability (δ), crossover rate (CR), mutation probability (p_m), distribution index (η) and differential weight (F) were adjusted to 0.9, 20, 1.0, 1/n (where n represents the number of variables) and 0.5, respectively. For JMA, RVEA and NSGA III, the simulated binary crossover was used with a distribution index (η_c) set to 30. Each benchmark instance was tested with 30 independent runs, and the total number of function evaluations was set to 300,000 for each instance.

6.2. Metrics for performance

Final Generation Distance (GD): GD metric computes convergence calculates the average Euclidean distance between the individuals in the search space and the nearest true Pareto front (PF) approximation point. The GD formula is:

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n}$$

where d_i represents the distance between the individual x_i and the nearest true PF point, and n is the number of approximation PF points.

Inverted Generation Distance (IGD): This metric assesses the diversity and convergence. It calculates the average minimal Euclidean distance between the true Pareto front (PF_{true}) and the dominant solutions in the approximation front (PF_{known}). The IGD formula is that Inverted Generation Distance measures both the convergence and diversity. The PF_{true} is the set of all solutions in true PF with uniform dispersion, and X is the collection of the approximation front's most prominent solutions PF_{known}:

$$GD = \frac{\sum_{v \in PF_{true}} d(v, X)}{|PF_{true}|}$$

where $d(v, X)$ represents the minimal Euclidean distance among the true PF point v and the points in X (the dominant solutions in PF_{known}).

6.3. Result analyses

After conducting the experimentation as described in the setup, the performance of the proposed

JMA algorithm was assessed using various performance metrics. The results for bi-objective benchmark functions UF1 to UF7 are presented in Tables 2 and 3, while Table 4 shows the results for tri-objective benchmark functions. The evaluation involved comparing JMA to other algorithms like RVEA, CDG, NSGA III and MOEA/D-DE TCH, with metrics including best, average and worst of GD and IGD are used for the comparison along with standard deviation.

In general, JMA demonstrated superior performance compared to other algorithms for instances UF1 to UF4, with the best GD and IGD values. However, for instance UF2, it had the lowest standard deviation value for IGD. The efficacy of existing algorithms decreased for all instances, as indicated by the comments in Table 4. In terms of dominance over the search space, JMA outperformed other algorithms in most cases. For instance UF5, NSGA III and CDG had the highest IGD and GD values, respectively, while JMA dominated the remaining metrics. Similarly, for instance UF6, JMA achieved reduced values and its standard deviation for IGD was superior to other compared algorithms. However, RVEA performed better for UF6, and CDG took the lead for UF7.

Table 2. Outcome of Bi-Objective Benchmark Function.

S. No	Instances	Techniques	IGD				GD			
			Best	Mean	SD	Worst	Best	Mean	SD	Worst
1	UF1	JMA	**0.00220**	**0.00229**	**0.00006**	**0.00243**	**0.00200**	**0.00207**	**0.00004**	**0.00216**
		CDG	0.00235	0.00249	0.00008	0.00268	0.00191	0.00196	0.00003	0.00202
		RVEA	0.00298	0.00326	0.00014	0.00354	0.00249	0.00259	0.00006	0.00273
		MOEA/D-DE TCH	0.00698	0.00729	0.00018	0.00772	0.00625	0.00682	0.00027	0.00736
		NSGA 3	0.01398	0.01457	0.00028	0.01504	0.01296	0.01401	0.00050	0.01508
2	UF2	JMA	**0.00439**	**0.00454**	**0.00008**	**0.00472**	**0.00376**	**0.00396**	**0.00013**	**0.00429**
		CDG	0.00552	0.00566	0.00009	0.00586	0.00485	0.00516	0.00014	0.00549
		RVEA	0.00610	0.00638	0.00013	0.00660	0.00549	0.00579	0.00018	0.00622
		MOEA/D-DE TCH	0.00965	0.00989	0.00016	0.01024	0.00840	0.00911	0.00027	0.00970
		NSGA 3	0.01200	0.01241	0.00022	0.01283	0.01065	0.01125	0.00035	0.01215
3	UF3	JMA	**0.01745**	**0.01830**	**0.00035**	**0.01878**	**0.01586**	**0.01658**	**0.00040**	**0.01751**
		CDG	0.02050	0.02144	0.00044	0.02216	0.01760	0.01862	0.00054	0.01977
		RVEA	0.01786	0.01904	0.00040	0.01971	0.01791	0.01912	0.00062	0.02070
		MOEA/D-DE TCH	0.02119	0.02221	0.00048	0.02297	0.01952	0.02069	0.00062	0.02190
		NSGA 3	0.03912	0.04398	0.00222	0.04835	0.03588	0.04268	0.00449	0.05068
4	UF4	JMA	**0.03143**	**0.03329**	**0.00064**	**0.03400**	**0.02891**	**0.03060**	0.00139	**0.03455**
		CDG	0.03389	0.03586	0.00070	0.03683	0.03005	0.03289	**0.00097**	0.03456
		RVEA	0.03742	0.04041	0.00071	0.04101	0.03365	0.03682	0.00146	0.03977
		MOEA/D-DE TCH	0.03218	0.03398	0.00064	0.03454	0.03247	0.03377	0.00070	0.03495
		NSGA 3	0.08162	0.08662	0.00142	0.08751	0.06070	0.06523	0.00191	0.06902

Table 3. Outcome of Bi-Objective Benchmark Function.

S. No	Instances	Techniques	IGD				GD			
			Best	Mean	SD	Worst	Best	Mean	SD	Worst
5	UF5	JMA	0.15818	**0.19100**	**0.00872**	**0.19363**	0.14392	**0.15157**	**0.00417**	**0.15489**
		CDG	0.16822	0.23180	0.03386	0.30217	**0.14082**	0.20252	0.04266	0.31026
		RVEA	0.14689	0.21856	0.02779	0.26984	0.14586	0.20253	0.04058	0.28208
		MOEA/D-DE TCH	0.15702	0.20698	0.03319	0.28931	0.15592	0.23648	0.06664	0.44211
		NSGA 3	**0.13599**	0.23904	0.04769	0.32968	0.18719	0.27540	0.04874	0.38700
5	UF6	JMA	0.05205	0.06360	0.00337	0.06689	0.06227	0.06789	0.00297	0.07778
		CDG	0.05426	0.07145	0.01530	0.10996	0.08228	0.07859	0.01353	0.13724
		RVEA	**0.02654**	**0.02878**	0.00060	**0.02908**	**0.04252**	**0.04511**	**0.00089**	**0.04601**
		MOEA/D-DE TCH	0.04087	0.04272	0.00038	0.04284	0.06722	0.07027	0.00137	0.07108
		NSGA 3	0.05442	0.05769	0.00061	0.05781	0.09939	0.10851	0.00478	0.12923
7	UF7	JMA	0.00395	0.00407	**0.00005**	0.00415	0.00302	0.00313	0.00006	0.00327
		CDG	**0.00280**	**0.00292**	0.00006	**0.00308**	**0.00244**	**0.00256**	**0.00006**	**0.00270**
		RVEA	0.00363	0.00376	0.00006	0.00387	0.00319	0.00332	0.00009	0.00350
		MOEA/D-DE TCH	0.00422	0.00440	0.00008	0.00456	0.00390	0.00406	0.00011	0.00437
		NSGA 3	0.00465	0.00488	0.00009	0.00504	0.00423	0.00452	0.00014	0.00478

6.3.1. Discussion based on Best value of IGD

The IGD value serves as a vital evaluation criterion for assessing the optimal solution by striking a balance between convergence and diversity. The results in Figure 8 demonstrate that the JMA algorithm outperforms existing methods, as evidenced by its superior convergence curves for UF3. Additionally, Table 4 reveals that the JMA algorithm consistently surpasses CDG, RVEA, MOEA/DDE TCH and NSGA 3 across various instances of bi-objective benchmark functions. However, for instances UF6 and UF7 (as shown in Table 3), some other methods like RVEA, CGD and NSGA III display better performance than the JMA algorithm. Notably, MOEA/DDE TCH consistently exhibits inferior results.

6.3.2. Discussion based on the Mean value of IGD

The JMA algorithm demonstrated superior performance based on the Mean IGD values when compared to instances UF9 and UF10 from Table 4, as well as one instance from Table 3, and all examples are listed in Table 4. However, it is important to note that there were significant variations in the results when the suggested approach was applied to benchmark examples UF6, UF7 and UF8. For example, the RVEA algorithm assigned a value of 5.20E-02 to the UF8 instance, while the JMA algorithm showed values of 5.05E-02 and 2.19E-01 for the UF9 and UF10 instances, respectively. Furthermore, Table 1 provides an overview highlighting the overall superiority of the JMA algorithm over other methodologies.

Table 4. Outcome of Tri-Objective Benchmark Function.

S. No	Instances	Techniques	IGD				GD			
			Best	Mean	SD	Worst	Best	Mean	SD	Worst
8	UF8	JMA	0.05478	0.05663	0.00080	0.05807	0.01305	0.01402	0.00042	0.01523
		CDG	0.05606	0.05786	0.00097	0.05963	0.01332	0.01400	0.00044	0.01493
		RVEA	**0.05032**	**0.05197**	**0.00072**	**0.05338**	**0.01182**	**0.01246**	**0.00036**	**0.01319**
		MOEA/D-DE TCH	0.06482	0.06698	0.00110	0.07010	0.01661	0.01742	0.00042	0.01836
		NSGA 3	0.05522	0.05685	0.00084	0.05949	0.01337	0.01423	0.00043	0.01517
9	UF9	JMA	**0.04800**	**0.05046**	0.00115	**0.05244**	**0.01622**	**0.01714**	0.00050	**0.01817**
		CDG	0.05284	0.05533	0.00139	0.05840	0.01718	0.01816	0.00072	0.01983
		RVEA	0.05424	0.05622	0.00098	0.05856	0.01731	0.01809	**0.00046**	0.01916
		MOEA/D-DE TCH	0.06028	0.06249	0.00132	0.06514	0.02115	0.02199	0.00056	0.02393
		NSGA 3	0.05359	0.05549	0.00116	0.05758	0.01906	0.02043	0.00070	0.02156
10	UF10	JMA	0.20933	**0.21940**	**0.00439**	**0.22743**	**0.07279**	**0.07689**	**0.00353**	**0.08593**
		CDG	0.25824	0.30991	0.03194	0.37105	0.10525	0.14693	0.02150	0.19296
		RVEA	**0.20595**	0.32023	0.00959	0.32466	0.09074	0.11699	0.00834	0.14632
		MOEA/D-DE TCH	0.42850	0.56255	0.07129	0.76738	0.20086	0.32129	0.09060	0.65955
		NSGA 3	0.27315	0.30325	0.01123	0.31205	0.13413	0.14703	0.01130	0.17827

6.3.3. Discussion based on Standard Deviation of IGD

This metric impacts the evaluation of the multiobjective algorithm as it reflects the overall convergence of the population. The stability and superiority of the proposed Java Macaque Approach (JMA) over previous techniques are demonstrated in the results shown in reference Figure 12 based on the Standard Deviation value of the Inverted Generational Distance (IGD). To illustrate this, let us consider instances UF1 to UF4 from Table 4, which exhibit excellent values of 0.00006, 0.00008, 0.00035 and 0.00064 for the JMA. Upon further examination, it was found that the JMA outperformed other algorithms, such as RVEA and MOEA/DDE, by dominating seven examples. However, these algorithms were dominant in the remaining instances. Therefore, the JMA demonstrates superior performance compared to existing algorithms, establishing its stability and effectiveness.

6.3.4. Discussion based on the worst value of IGD

Tables 2, 3 and 4 present an evaluation of the proposed mechanism based on the worst IGD value across the entire population. Among the evaluated methods, JMA demonstrates superior outcomes in seven cases. However, the methods RVEA and CGD yield IGD values of 2.91E-02, 3.08E-03 and 5.34E-02 in three instances: UF6, UF7 and UF8. On the other hand, according to Table 1, it is evident that the suggested method excels in terms of worst individual as well. In summary, the results indicate the performance of both SOGA and the traditional genetic algorithm.

6.3.5. Discussion based on the best and worst values of GD

The performance of the Java Macaque Algorithm (JMA) is compared to other existing algorithms in Figures 8 to 12. While the JMA achieves the best GD values in most cases, there are six instances (UF1 to UF4, UF9 and UF10) where it consistently has a higher value. The RVEA algorithm performs slightly better than JMA in UF6 and UF8, and the CGD algorithm dominates UF5 and UF7 with superior values. In terms of the worst GD value, except for UF8, JMA outperforms the other algorithms in the majority of occurrences as shown in Tables 4 and 6. Furthermore, RVEA outperforms CGD in two cases (UF6 and UF8), while CGD outperforms RVEA in only one case (UF7). Based on the results of the analysis, it can be concluded that the proposed JMA dominates both the best and worst GD values.

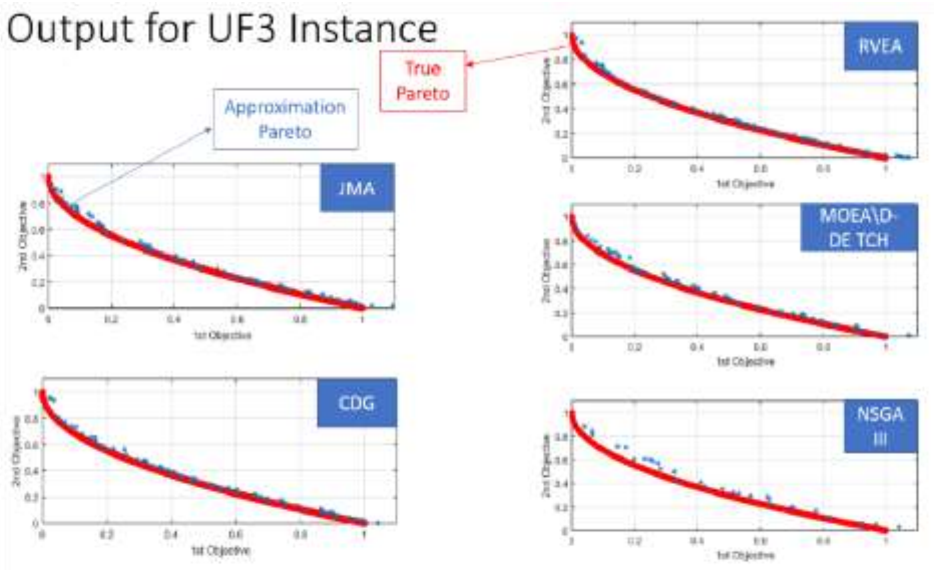

Figure 8. Outcome of Instance UF3.

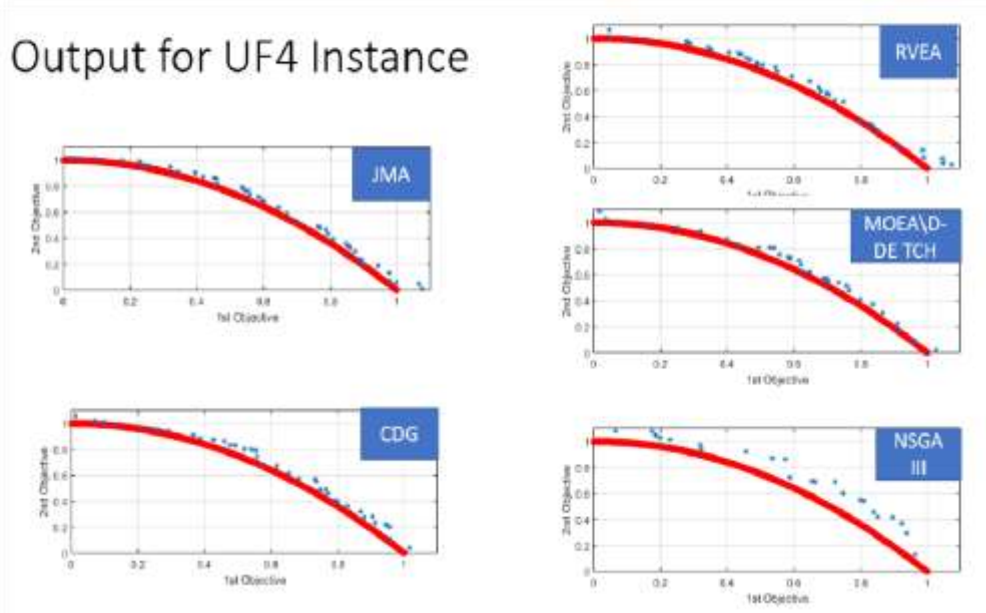

Figure 9. Outcome of Instance UF4.

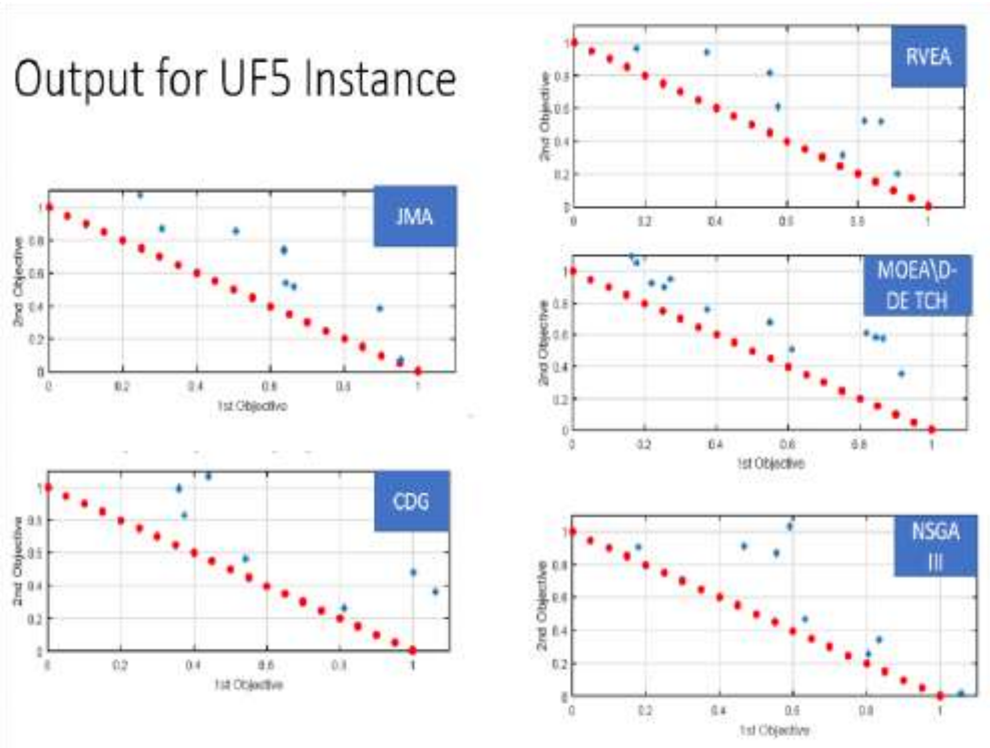

Figure 10. Outcome of Instance UF5.

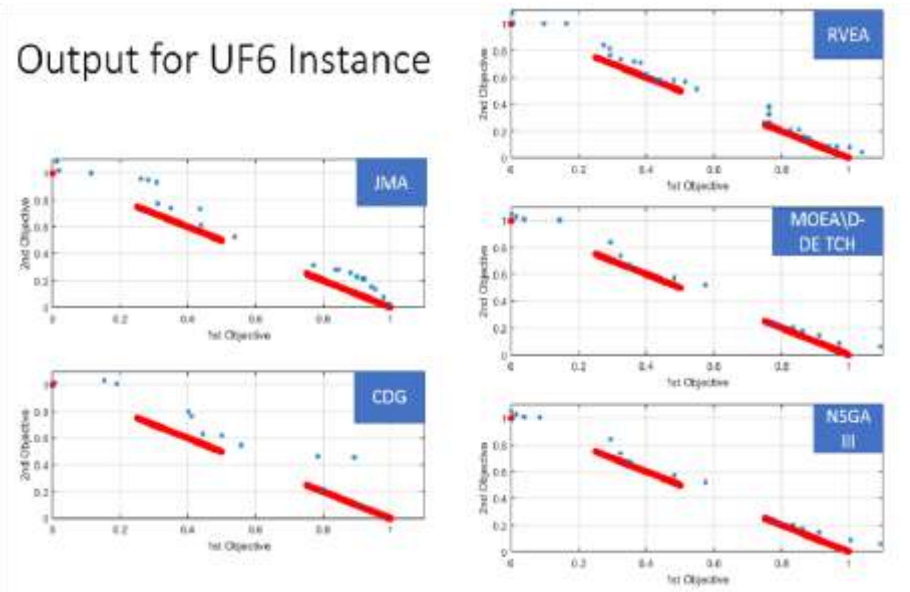

Figure 11. Outcome of Instance: UF6.

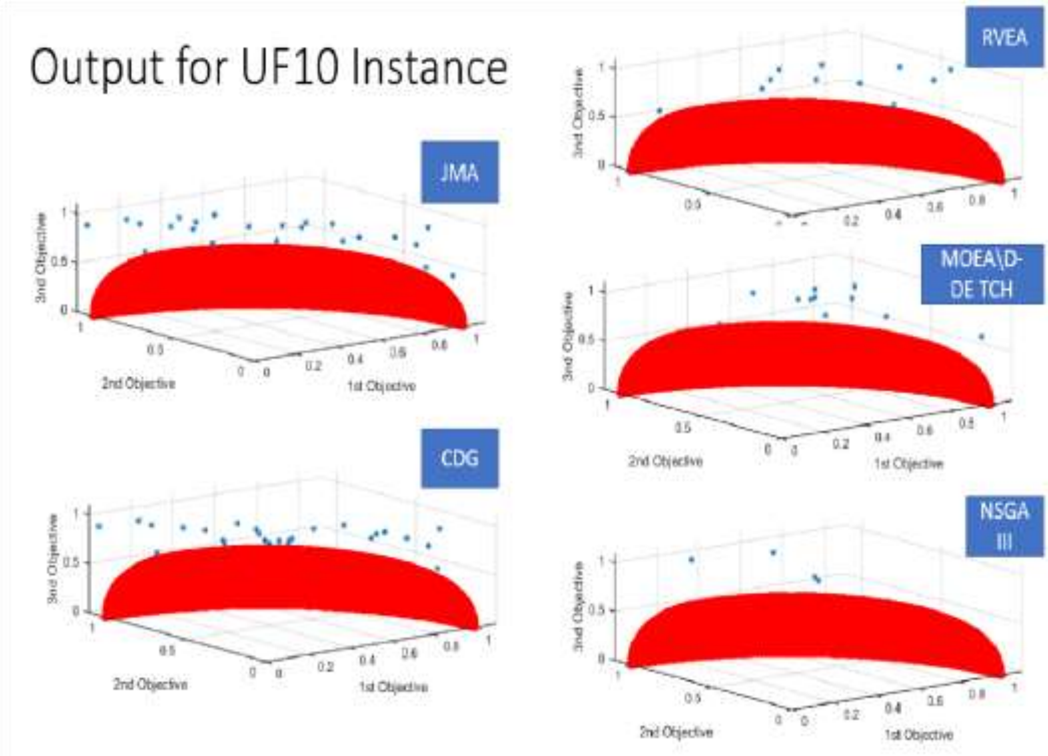

Figure 12. Outcome of Instance UF10.

6.3.6. Discussion based on the mean and SD values of GD

Figures 8 to 12 illustrate a comparison between the performance of the JMA algorithm and current methods. The mean value represents the difference between the converged population and the standard deviation. The JMA algorithm demonstrates superiority over the mean value of the GD method in seven examples: UF1 to UF3, UF5 and UF10. Additionally, the JMA algorithm dominates the SD value of GD in five cases: UF1 to UF3, UF5 and UF10. However, in cases like UF4 and UF7, CGD ranks higher, despite RVEA outperforming the other instances.

7. Conclusion and Future Work

The proposed MOJMA incorporates a Utopian-based lexicographic sorting approach to optimize the exploration and exploitation in multi-objective optimization problems. This method effectively maintains a well-distributed collection of Pareto fronts, resulting in superior performance compared to other methods such as the TSP and the benchmark functions from CEC09. Specifically, it outperforms the bi-objective problems UF1 to UF7 and the tri-objective problems UF8 to UF10. The performance of proposed JMA model is used to experiment on the multi-objective Travelling Salesman Problem. Additionally, the performance results of JMA on the multi-objective continuous optimization problem is noted. The proposed JMA algorithm demonstrates superior performance in optimizing multi-objective problems. It outperforms the compared algorithms in both discrete and continuous multi-objective optimization problems. The results validate the effectiveness of JMA in achieving better convergence and diversity. Further, the JMA will be optimized by incorporating the hyper parameter tuning that suits the algorithm for more complex optimization.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research is supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R195) Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Author contributions

Conceptualization, D.K.; methodology, D.K.; validation, R.R. and M.R.; formal analysis, S.B.; writing—original draft preparation, D.K.; writing—review and editing, M.R. and N.M.; supervision, R.R. and M.R; funding acquisition, S.B. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflicts of interest.

References

1. X. Yang, Nature-inspired metaheuristic algorithms, Luniver Press, second edition. 2010.
2. D. Kumar, S. Kumar, R. Bansal, P. Singla, A survey to nature inspired soft computing, *Int. J. Inf. Syst. Model.*, **8** (2017), 112–133. https://doi.org/10.4018/IJISMD.2017040107
3. A. Sharma, A. Sharma, B. K. Panigrahi, D. Kiran, R. Kumar, Ageist spider monkey optimization algorithm, *Swarm Evol. Comput.*, **28** (2016), 58–77. https://doi.org/10.1016/j.swevo.2016.01.002
4. J. C. Bansal, H. Sharma, S. S. Jadon, M. Clerc, Spider monkey optimization algorithm for numerical optimization, *Memetic Comput.,* **6** (2014): 31–47. https://doi.org/10.1007/s12293-013-0128-0
5. H. Sharma, G. Hazrati, J. C. Bansal, Spider monkey optimization algorithm, *Evol. Swarm Intell. Algorithms*, (2019), 43–59.
6. C. A. G. Santos, P. K. M. M. Freire, S. K. Mishra, Cuckoo search via Lévy flights for optimization of a physically-based runoff-erosion model, *J. Urban Environ. Eng.*, **6** (2012), 123–131. https://www.jstor.org/stable/26203380
7. S. Yılmaz, E. U Kuc¸ uksille, A new modification approach on bat algorithm for solving optimization problems, *Appl. Soft Comput.*, **28** (2015), 259–275. https://doi.org/10.1016/j.asoc.2014.11.029
8. X. S. Yang, Firefly algorithm, Engineering optimization, 2010, 221–230.

9. J. Kennedy, Particle swarm optimization, In: Encyclopedia of machine learning, 760–766. *Springer*, 2011.
10. D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, *J. Global Optim.*, **39** (2007), 459–471. https://doi.org/10.1007/s10898-007-9149-x
11. M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, *IEEE Comput. Intell. M.*, **1** (2006), 28–39. https://doi.org/10.1109/MCI.2006.329691
12. R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.*, **11** (1997), 341–359. https://doi.org/10.1007/s11042-022-12409-x
13. J. H. Holland, Adaptation in natural and artificial systems*, Univ. Mich. Press*. Ann Arbor, 1975.
14. F. S. Gharehchopogh, T. Ibrikci, An improved african vultures optimization algorithm using different fitness functions for multi-level thresholding image segmentation, *Multimed. Tools Appl.*, (2023), 1–47. https://doi.org/10.1007/s11042-023-16300-1
15. S. T. Shishavan, F. S. Gharehchopogh, An improved cuckoo search optimization algorithm with genetic algorithm for community detection in complex networks, *Multimed. Tools Appl.*, **81** (2022), 25205–25231.
16. F. S. Gharehchopogh, Quantum-inspired metaheuristic algorithms: comprehensive survey and classification, *Artif. Intell. Rev.,* **56** (2023), 5479–5543, 2023. https://doi.org/10.1007/s10462-022-10280-8
17. A. Laith, M. Shehab, M. Alshinwan, S. Mirjalili, M. A. Elaziz, Ant lion optimizer: A comprehensive survey of its variants and applications, *Arch. Comput. Method. Eng.,* **28** (2021), 1397–1416. https://doi.org/10.1007/s11831-020-09420-6
18. X. Yang, J. Zou, S. Yang, J. Zheng, Y. Liu, A fuzzy decision variables framework for large-scale multiobjective optimization, *IEEE T. Evolut. Comput.*, **27** (2021), 445–459. https://doi.org/10.1109/TEVC.2021.3118593
19. A. M. Basset, R. Mohamed, M. Abouhawwash, Balanced multi-objective optimization algorithm using improvement-based reference points approach, *Swarm Evol. Comput.*, **60** (2021), 100791. https://doi.org/10.1016/j.swevo.2020.100791
20. M. A. Basset, R. Mohamed, S. Mirjalili, A novel whale optimization algorithm integrated with nelder–mead simplex for multi-objective optimization problems, *Knowledge-Based Syst.*, **212** (2021), 106619. https://doi.org/10.1016/j.knosys.2020.106619
21. B. Xu, G. Zhang, K. Li, B. Li, H. Chi, Y. Yao, et al., Reactive power optimization of a distribution network with high-penetration of wind and solar renewable energy and electric vehicles, *Protect. Contr. Mod. Pow.*, **7** (2022), 51. https://doi.org/10.1016/j.ins.2022.05.123
22. F. S. Gharehchopogh, An improved harris hawks optimization algorithm with multistrategy for community detection in social network, *J. Bionic Eng.*, **20** (2023), 1175–1197. https://doi.org/10.1007/s42235-022-00303-z
23. H. Ma, S. Shen, M. Yu, Z. Yang, M. Fei, H. Zhou, Multi-population techniques in nature inspired optimization algorithms: A comprehensive survey, *Swarm Evol. Comput.*, **44** (2019), 365–387. https://doi.org/10.1016/j.swevo.2018.04.011
24. Z. Li, V. Tam, L. K. Yeung, An adaptive multi-population optimization algorithm for global continuous optimization, *IEEE Access*, **9** (2021), 19960–19989. https://doi.org/10.1109/ACCESS.2021.3054636

25. X. Peng, Z. Shi, Finding informative collaborators for cooperative co-evolutionary algorithms using a dynamic multi-population framework, *In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2016, 1–6. https://doi.org/10.1109/SSCI.2016.7849958
26. K. Dinesh, R. Ramalingam, A. Dumka, R. Singh, I. Alsukayti, D. Anand, et al., An intelligent optimized route-discovery model for IoT-based VANETs, *Processes*, **9** (2021), 2171. https://doi.org/10.3390/pr9122171
27. D. Saravanan, R. Rajakumar, M. Sreedevi, K. Dinesh, S. V. Sudha, D. K. Anguraj, et al., Multi-objective swarm-based model for deploying virtual machines on cloud physical servers, *Distrib. Parallel Dat.*, **41** (2023), 75–93. https://doi.org/10.1007/s10619-021-07341-2
28. Y. Liu, Y. Shi, H. Chen, A. A. Heidari, W. Gui, M. Wang, et al., Chaos-assisted multi-population salp swarm algorithms: Framework and case studies, *Expert Syst. Appl.*, **168** (2021), 114369. https://doi.org/10.1016/j.eswa.2020.114369
29. H. Li, Q. Zhang, Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii, *IEEE T. Evol. Comput.*, **13** (2009), 284–302. https://doi.org/10.1109/TEVC.2008.925798
30. D. Karunanidy, S. Ramalingam, A. Dumka, R. Singh, M. Rashid, A. Gehlot, et al., Jma: Nature-inspired java macaque algorithm for optimization problem, *Mathematics*, **10** (2022), 688. https://doi.org/10.3390/math10050688
31. K. Dinesh, J. Amudhavel, R. Rajakumar, P. Dhavachelvan, R. Subramanian, A novel self-organisation model for improving the performance of permutation coded genetic algorithm, *Int. J. Adv. Intell. Paradigms*, **17** (2020), 299–322. https://doi.org/10.1504/IJAIP.2020.109512
32. D. Kalyanmoy, D Saxena, Searching for pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems, *In*: *Proceedings of the world congress on computational intelligence, (2006)*, 3352–3360.
33. B. Xu, D. Gong, Y. Zhang, S. Yang, L. Wang, Z. Fan, et al., Cooperative co-evolutionary algorithm for multi-objective optimization problems with changing decision variables, *Inf. Sci.*, **607** (2022), 278–296.
34. H. Mohammadzadeh, F. S. Gharehchopogh, A multi-agent system based for solving high-dimensional optimization problems: a case study on email spam detection, *Int. J. Commun. Syst.*, **34** (2021), 4670. https://doi.org/10.1002/dac.4670
35. Y. Yuan, H. Xu, B. Wang, X. Yao, A new dominance relation-based evolutionary algorithm for many-objective optimization, *IEEE T. Evol. Comput.*, **20** (2015), 16–37. https://doi.org/10.1109/TEVC.2015.2420112
36. E. Zitzler, S. Kunzli, Indicator-based selection in multiobjective search, In: International Conference on Parallel Problem Solving from Nature, 832–842. Springer, 2004. https://doi.org/10.1007/978-3-540-30217-9_84
37. Q. Zhang, H. Li, Moea/d: A multiobjective evolutionary algorithm based on decomposition, *IEEE T. Evol. Comput.*, **11** (2007), 712–731. https://doi.org/10.1109/TEVC.2007.892759
38. D. Kalyanmoy, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, *IEEE T. Evol. Comput.*, **6** (2002), 182–197. https://doi.org/10.1109/4235.996017
39. A. Trivedi, D. Srinivasan, K. Sanyal, A. Ghosh, A survey of multiobjective evolutionary algorithms based on decomposition, *IEEE T. Evol. Comput.*, **21** (2017), 440–462. https://doi.org/10.1109/TEVC.2016.2608507

40. X. Cai, Z. Mei, Z. Fan, Q. Zhang, A constrained decomposition approach with grids for evolutionary multiobjective optimization, *IEEE T. Evol. Comput.*, **22** (2017), 564–577. https://doi.org/10.1109/TEVC.2017.2744674
41. R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, *IEEE T. Evol. Comput.,* **20** (2016), 773–791. https://doi.org/10.1109/TEVC.2016.2519378
42. K. Deb, K. Miettinen, S. Chaudhuri, Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches, *IEEE T. Evol. Comput.,* **14** (2010), 821–841. https://doi.org/10.1109/TEVC.2010.2041667
43. X. Ma, Q. Zhang, J. Yang, Z. Zhu, On tchebycheff decomposition approaches for multi-objective evolutionary optimization, *IEEE T. Evol. Comput.*, 2017. https://doi.org/10.1109/TEVC.2017.2704118
44. G. Syswerda, Uniform crossover in genetic algorithms. In: Proceedings of the third international conference on Genetic algorithms, *Morgan Kaufmann Publishers*, **3** (1989), 2–9.
45. H. Ishibuchi, N. Akedo, Y. Nojima, Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems, *IEEE T. Evol. Comput.*, **19** (2015), 264–283. https://doi.org/10.1109/TEVC.2014.2315442
46. K. Deb, Multi-objective genetic algorithms: Problem difficulties and construction of test problems, *Evol. Comput.,* **7** (1999), 205–230. https://doi.org/10.1162/evco.1999.7.3.205
47. J. J. Durillo, A. J Nebro, jmetal: A java framework for multi-objective optimization, *Adv. Eng. Soft.*, **42** (2011), 760–771. https://doi.org/10.1016/j.advengsoft.2011.05.014
48. Q. Lin, J. Chen, Z. H. Zhan, W. N. Chen, C. A. C. Coello, Y. Yin, et al., A hybrid evolutionary immune algorithm for multiobjective optimization problems, *IEEE T. Evol. Comput.*, **20** (2016), 711–729. https://doi.org/10.1109/TEVC.2015.2512930
49. H. Karshenas, R. Santana, C. Bielza, P. Larranaga, Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables, *IEEE T. Evol. Comput.*, **18** (2014), 519–542. https://doi.org/10.1109/TEVC.2013.2281524
50. Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, S. Tiwari, Multiobjective optimization test instances for the cec 2009 special session and competition, University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report, 264, 2008.