Knowledge-Based Systems 150 (2018) 175–197

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Multi-objective spotted hyena optimizer: A Multi-objective optimization algorithm for engineering problems

Gaurav Dhiman*, Vijay Kumar

Computer Science and Engineering Department, Thapar University, Patiala, Punjab, India

ARTICLE INFO

Article history:
Received 9 August 2017
Revised 1 March 2018
Accepted 8 March 2018
Available online 8 March 2018

Keywords:
Multi-objective optimization
Spotted hyena optimizer
Constrained optimization
Constrained engineering design problems

ABSTRACT

This paper proposes a multi-objective version of recently developed Spotted Hyena Optimizer (SHO) called Multi-objective Spotted Hyena Optimizer (MOSHO). It is used to optimize the multiple objectives problems. In the proposed algorithm, a fixed-sized archive is employed for storing the non-dominated Pareto optimal solutions. The roulette wheel mechanism is used to select the effective solutions from archive to simulate the social and hunting behaviors of spotted hyenas. The proposed algorithm is tested on 24 benchmark test functions and compared with six recently developed metaheuristic algorithms. The proposed algorithm is then applied on six constrained engineering design problems to demonstrate its applicability on real-life problems. The experimental results reveal that the proposed algorithm performs better than the others and produces the Pareto optimal solutions with high convergence.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

In recent years metaheuristic techniques have become a popular research area to solve real-life problems. Due to computationally inexpensiveness and ease to understand, these have been widely used in various engineering design applications [1].

The most challenging characteristic of these algorithms is multi-objectivity. The multi-objective optimization deals with optimization of multiple and competing objective functions. The main challenge in multi-objective optimization is to address the multiple objectives which are frequently in conflict. There are two main approaches for multi-objective optimization such as *priori* and *posteriori* [2,3].

In *priori* approaches, a multi-objective problem is converted to a single-objective with a set of weights that defines the significance of each objective to solve the problem. A *posteriori* method permits to explore the performance of problem and selects one of the obtained solutions based on their obligations [4].

In contrast to single-objective, there is no single solution in multi-objective optimization and represents various trade-offs between the objectives. Therefore, it is more time consuming to obtain the Pareto fronts because it usually requires to produce many points on the Pareto front for good approximations. Even accurate solutions on a Pareto front, there is still no guarantee that these solution points will distribute uniformly on the front [5]. The higher dimensional problems can have extremely complex hypersurface as its Pareto front [3,6–8]. Thus, it is more challenging to solve such high-dimensional problems.

The concept of multi-objective optimization using stochastic techniques was proposed by David Schaffer [9]. The advantages of these techniques are local optima avoidance and gradient-free mechanism that made them applicable to real world problems. The application of multi-objective optimization techniques can be found in various fields such as civil engineering [10], mechanical engineering [11,12], system engineering [13], bioinformatics [14], software engineering [15], artificial intelligence [16], and other fields [17–19].

Some of the well-known optimization techniques are: Non-dominated Sorting Genetic Algorithm 2 (NSGA-2) [20], Multi-objective Particle Swarm Optimization (MOPSO) [21], and Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) [22]. These are able to estimate the true Pareto optimal solutions. However, these are unable to solve all types of optimization problems [23]. The motivation behind this work is to propose a novel multi-objective optimization algorithm called Multi-objective Spotted Hyena Optimizer (MOSHO) which is based on Spotted Hyena Optimizer (SHO) [1]. The main contributions of this paper are:

- An archive component has been incorporated into SHO that is responsible for storing all non-dominated Pareto optimal solutions which are obtained so far.

* Corresponding author.
E-mail address: gaurav.dhiman@thapar.edu (G. Dhiman).

https://doi.org/10.1016/j.knosys.2018.03.011
0950-7051/© 2018 Elsevier B.V. All rights reserved.

- A group selection approach has been proposed to decide whether the solutions are integrated in group corresponding to the position of prey from the archive.
- A grid mechanism has been incorporated in SHO to omit one of the most crowded sections and improves the non-dominated solutions in the archive.

The performance of the proposed MOSHO has been evaluated on seven bi-objective and three tri-objective multi-objective test problems taken from the CEC (Congress on Evolutionary Computation 2009) competition and special session on multi-objective optimization algorithms [24]. The proposed algorithm has also been evaluated on five ZDT [25] and nine DTLZ (Scalable test problems) [26] multi-objective test problems. The results have been compared and verified with six recently developed optimization techniques such as MOPSO [21], NSGA-2 [20], MOEA/D [22], PESA-2 [27], SPEA-2 [28], and MOACO [29]. The four well-known performance metrics have been used to assess these algorithms. The results show that MOSHO can prove to be a very promising metaheuristic technique in the field of multi-objective optimization.

The rest of this paper is discussed as follows: Section 2 presents the basic concepts of multi-objective optimization and related work done in the field of multi-objective optimization. Section 3 describes the concepts of SHO and then proposes the MOSHO algorithm. The results and discussions are given in Section 4. In Section 5, the performance of MOSHO is tested on six constrained multi-objective engineering design problems and compared with other well-known optimizers. The conclusion and outlines of some future works are drawn in Section 6.

2. Background

This section provides the basic concepts of multi-objective optimization techniques.

2.1. Basic concepts of multi-objective optimization

Multi-objective optimization refers to the optimization with more than one objective (criterion) function of a given problem. It can be formulated as [30,31]:

$$Minimize : F(\vec{z}) = [f_1(\vec{z}), f_2(\vec{z}), \dots, f_n(\vec{z})] \tag{1}$$

Subject to:

$$g_i(\vec{z}) \geq 0, \quad i = 1, 2, \dots, m \tag{2}$$

$$h_i(\vec{z}) = 0, \quad i = 1, 2, \dots, p \tag{3}$$

where $\vec{z} = [z_1, z_2, \dots, z_k]^T$ is the vector of decision variables, m is the number of inequality constraints, p is the number of equality constraints, g_i is the *ith* inequality constraints, h_i is the *ith* equality constraints, and *obj* is the number of objective functions $f_i : \mathbb{R}^{obj} \to \mathbb{R}$.

In multi-objective, the solutions in a search space cannot be compared by relational operators due to multi-criterion comparison metrics. The comparison of two solutions was first proposed by Edgeworth [32] and further extended by Pareto [33]. The mathematical formulation of Pareto dominance is described as [30]:

Definition 1. Pareto dominance.

Assume there are two vectors $\vec{x}=(x_1, x_2, \dots, x_r)$ and $\vec{y}=(y_1, y_2, \dots, y_r)$. Vector $\vec{x}$ is said to dominate vector $\vec{y}$ (such as $x \prec y$) if and only if:

$$\forall i \in \{1, 2, \dots, r\} : \; f_i(\vec{x}) \leq f_i(\vec{y}) \; \wedge \exists i \in \{1, 2, \dots, r\} : f_i(\vec{x}) < f_i(\vec{y}) \tag{4}$$

Definition 2. Pareto optimality.

A solution $\vec{x} \in X$ is called Pareto optimal if and only if:

$$\nexists \, \vec{y} \; \in X \mid \vec{y} \prec \vec{x} \tag{5}$$

Definition 3. Pareto optimal set.

The set of all Pareto optimal solutions including all non-dominated solutions of a problem is called Pareto optimal set if and only if:

$$P_s = \{\vec{x}, \vec{y} \in X \mid \exists \vec{y} \succ \vec{x}\} \tag{6}$$

Definition 4. Pareto optimal front.

A set containing the objective values corresponding to Pareto optimal solutions in Pareto optimal set is called Pareto optimal front. It is defined as:

$$P_f = \{f(\vec{x}) \;\mid\; \vec{x} \in P_s\} \tag{7}$$

2.2. Related works

In the last few decades, a wide range of multi-objective techniques have been developed. Multi-objective metaheuristic based techniques deal with many difficulties such as infeasible solutions, diversity of solutions, and optimum separation [4]. To deal with these difficulties, two conditions are to be satisfied. First, the information is exchanged between the search space and search agents. Second, the multi-objective techniques should help to estimate the whole true Pareto optimal front in a single simulation run.

The popular multi-objective metaheuristic is Non-dominated Sorting Genetic Algorithm 2 (NSGA-2) [20]. This algorithm uses a fast non-dominated sorting method, elitist technique, and a niching operator. The population of NSGA-2 algorithm begins with randomized in which each individual is grouped based on the non-dominated sorting technique. To assist with the selection, mutation, and recombination operators, the second population is created. Both of these populations create a new one and then sorted again by the non-dominated sorting technique. The probability to select a new individual depends on the non-domination level for the final population. Finally, the whole process is run until the satisfactory result is found.

The second popular multi-objective metaheuristic technique is Multi-objective Particle Swarm Optimization (MOPSO) [21]. It uses the concept of PSO algorithm. MOPSO uses the external archive for storing and retrieving the Pareto optimal solutions. Additionally, a mutation operator is integrated in MOPSO to enhance the randomness and diversity in the distribution of Pareto optimal solutions.

Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) [22] is another popular multi-objective metaheuristics which decomposes a problem into scalar subproblems to optimize simultaneously. These subproblems are usually equal to the population size in which each subproblem is assigned a weighting vector to combine the objective functions into a single objective function. During search process, two tasks namely co-operation and completion, are performed. In co-operation task, the neighbouring members can collaborate the subproblem to generate solution. Furthermore, the solution of neighbouring subproblem(s) is replaced with the subproblem solution if it is better than the current one. This task is known as competition task. MOEA/D has less computational complexity and fast convergence speed than NSGA-2.

Angus and Woodward [29] proposed the enhanced version of Ant Colony Optimization (ACO) to solve multi-objective optimization problems. A new classification technique is proposed in Multi-objective Ant Colony Optimization (MOACO) which provides nearby solutions for intended problem domains. MOACO is based on the basic concepts of ACO. These are selection of pheromone model, construction process, solution estimation, and updating process. From an optimization viewpoint, this technique is able to balance the exploratory (diversity preserving) behavior with exploitative (quality enhancing) behavior.

Gong et al. [34] proposed a Non-dominated Neighbor Immune Algorithm (NNIA) for multi-objective optimization. NNIA used the non-dominated neighbor-based selection, immune inspired operator, heuristic search operators, and elitism. NNIA algorithm might lose the diversity as it refuses all the dominated antibodies when non-dominated antibodies are selected for proportional cloning. To overcome this issue, Yang et al. [35] developed an enhanced version of NNIA named as NNIA2, by integrating an adaptive ranks clone and K-nearest neighbour approaches to improve the diversity.

There are different types of Multi-objective algorithms have been proposed in recent years. These are Multi-objective Cat Swarm Optimization (MOCSO) [36], Multi-objective Teaching-Learning based Optimization algorithm (MO-TLBO) [37], Multi-objective Artificial Bee Colony algorithm (MOABC) [38], Multi-objective Gravitational Search Algorithm (MOGSA) [39], Multi-objective network clustering algorithm (GMOEA-net) [40], Multi-objective framework for SAR image segmentation (IMIS) [41], Quantum Particle Swarm Optimization (QPSO) [42], Multi-agent Genetic Algorithm (MAGA) [43], Immune Genetic Algorithm (IGA) [44], RDS-NSGA-II [45], Multi-objective Flower Pollination Algorithm (MOFPA) [5]. All of the above-mentioned algorithms are not able to solve all types of optimization problems according to NFL theorem [23]. It is probable that a new algorithm should be able to solve a problem that cannot be solved by the existing algorithms. A novel multi-objective version of the recently developed SHO is proposed in the preceding section for finding optimal solutions of multi-objective problems.

3. Proposed algorithm

In this section, we first describe the basic concepts of SHO followed by brief description of multi-objective version of SHO.

3.1. Inspiration

Social relationships are dynamic in nature. These are affected by the changes in relationship among comprising the network and individual leaving or joining the population. The animal behavior has been classified into three categories [46]:

- The first category includes environmental factors such as resource availability and competition with other animal species.
- The second category focuses on social preferences based on individual behavior.
- The third category has less attention from scientists which includes the social relations of species itself.

The social relation between animals is the inspiration of our work and correlates this behavior to spotted hyena which is scientifically named as Crocuta.

Hyenas are large dog-like carnivores. They live in savannas, grasslands, sub-deserts, and forests of both Africa and Asia. They live 10–12 years in the wild and up to 25 years in imprisonment. There are four known species of hyena such as spotted, striped, brown, and aardwolf. These differ in size, behavior, and type of diet. All of these species have a bear-like attitude.

Spotted hyenas are skillful hunters and largest of three other hyena species (i.e., striped, brown, and aardwolf). Spotted Hyena is also known as laughing hyena because its sounds is much similar to a human laugh. There are spots on their fur reddish brown in color with black spots. Spotted hyenas are complicated, intelligent, and highly social animals with really dreadful reputation. They have the ability to fight endlessly for territory and food.

In spotted hyenas, female members are dominant and live in their clan. However, male members leave their clan when they become adults and join a new clan. In a new family, they are lowest ranking members to get their share of meal. A male member who has joined the clan always stays with the same members (friends) for a long time. Whereas, a female is always assured of a stable place. An interesting fact about the spotted hyena is that they produce sound to communicate with each other during the searching of food source.

According to Ilany et al. [46], spotted hyenas usually rely on a network of trusted friends that have more than 100 members. They usually tie up with another spotted hyena that is a friend of a friend or linked in some way through kinship rather than any unknown spotted hyena. Spotted hyenas are social animals that can communicate with each other through specialized calls such as postures and signals. They use multiple sensory procedures to recognize their kin and other individuals. They can also recognize third party kin and rank the relationships between their clan mates during social decision making. The spotted hyena track prey by sight, hearing, and smell. Cohesive clusters are helpful for an efficient co-operation between spotted hyenas. In this work, the hunting technique and social relation of spotted hyenas are mathematically modeled to design the multi-objective SHO algorithm.

3.2. Spotted Hyena Optimizer (SHO)

To propose the multi-objective version of SHO [1], firstly the essential concept of this algorithm is discussed. The social relation and hunting behaviors of spotted hyenas are the main inspiration of this algorithm. SHO algorithm mimics the cohesive clusters between the trusted spotted hyenas. The four main steps of SHO are searching, encircling, hunting, and attacking. In SHO algorithm, the hunting behavior is guided by the group of trusted friends (so far solutions) towards the best search agent and saves the best optimal solutions.

In order to simulate the encircling behavior of spotted hyenas, the following equations are used [1]:

$$\vec{D}_h = | \vec{B} \cdot \vec{P}_p(x) - \vec{P}(x) | \tag{8}$$

$$\vec{P}(x+1) = \vec{P}_p(x) - \vec{E} \cdot \vec{D}_h \tag{9}$$

where $\vec{D}_h$ represents the distance between prey and spotted hyena. x indicates the current iteration. $\vec{P}_p$ and $\vec{P}$ represents the position vector of prey and spotted hyena, respectively. || and $\cdot$ are the absolute value and multiplication vector, respectively. $\vec{B}$ and $\vec{E}$ are the co-efficient vectors.

$\vec{B}$ and $\vec{E}$ are computed as:

$$\vec{B} = 2 \cdot \vec{rd}_1 \tag{10}$$

$$\vec{E} = 2\vec{h} \cdot \vec{rd}_2 - \vec{h} \tag{11}$$

$$\vec{h} = 5 - \left(Iteration \times \frac{5}{Max_{Iterations}} \right) \tag{12}$$

where, $Iteration = 0, 1, 2, \ldots, Max_{Iterations}$

Here, $\vec{h}$ is linearly decreased from 5 to 0 over the course of iterations. $\vec{rd}_1$ and $\vec{rd}_2$ are the random vectors that lie in the range [0,1]. There are different number of places which can be reached with respect to the current position by adjusting the value of vectors $\vec{B}$ and $\vec{E}$. This algorithm saves the optimal solution achieved so far and compels other search agents to update their positions.

In order to simulate the hunting behavior of spotted hyenas and find the promising regions of search space, the following equations are defined as:

$$\vec{D}_h = | \vec{B} \cdot \vec{P}_h - \vec{P}_k | \tag{13}$$

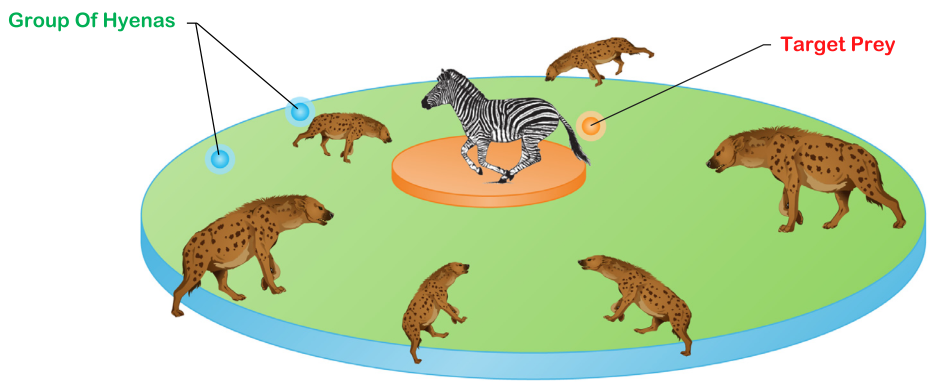


Fig. 1. Searching Prey ($|\vec{E}| > 1$).

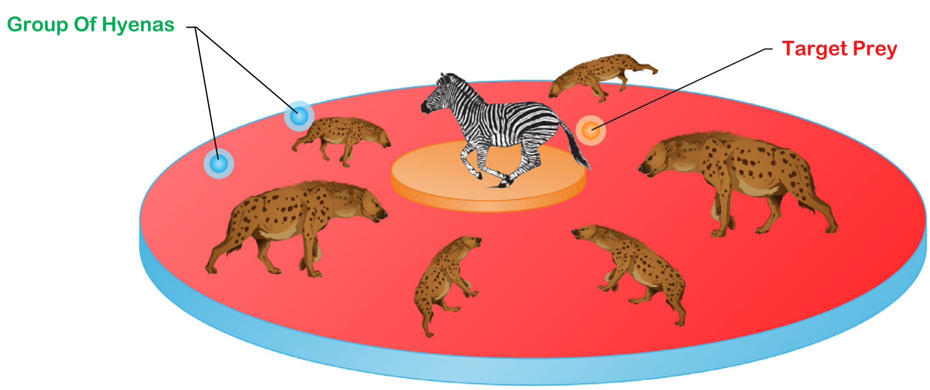


Fig. 2. Attacking Prey ($|\vec{E}| < 1$).

$$\vec{P}_k = \vec{P}_h - \vec{E} \cdot \vec{D}_h \tag{14}$$

$$\vec{C}_h = \vec{P}_k + \vec{P}_{k+1} + \ldots + \vec{P}_{k+N} \tag{15}$$

where N represents the number of iterations which can be computed as:

$$N = count_{nos}(\vec{P}_h, \vec{P}_{h+1}, \vec{P}_{h+2}, \ldots, (\vec{P}_h + \vec{M})) \tag{16}$$

$$\vec{P}(x+1) = \frac{\vec{C}_h}{N} \tag{17}$$

where *nos* indicates the number of solutions which are far similar to best optimal solution in the given search space, $\vec{C}_h$ represents a group or cluster of optimal solutions, $\vec{M}$ indicates a random vector in [0.5, 1]. $\vec{P}(x+1)$ saves N best optimal solutions and updates the positions of other search agents according to the position of best search agent.

The exploration is assurance by vector $\vec{E}$ with random values which are greater than 1 or less than 1 which compel the search agents to move away from the prey as shown in Figs. 1 and 2. Another constituent of SHO is $\vec{B}$ which is responsible for exploration. It contains random values in the range of [0, 5] which provides the weight of prey. To demonstrate the effect of distance and random behavior, assume $\vec{B}>1$ precedence than $\vec{B}<1$ as seen in Eq. (10) that will helpful for exploration and local optima avoidance.

The exploitation of SHO algorithm commences when $|\vec{E}|<1$ with random values of $\vec{E}$ are in $[-1, 1]$. Therefore, Fig. 5 shows the next position of a search agent lies between its current position and the position of the prey which will helpful to meet towards an estimated position of prey.

In order to perform the optimization, SHO algorithm commences with generating a set of random solutions as the population. During optimization, the search agents make a cluster towards the best search agent for updating their positions using Eqs. (15) - (17). In the meantime, parameters h and E are linearly decreased during the course of iterations. Finally, the positions of search agents that make a cluster are considered as optimal best solutions when the termination criteria is satisfied. From [1], it has been confirmed that SHO is able to solve optimization problems efficiently. The high exploration capability of SHO provides better results as compared to the existing metaheuristics. Due to this, the solutions produced by SHO do not stuck in local optima. This motivates us to propose the multi-objective version of SHO in the preceding subsection.

3.3. Multi-objective Spotted Hyena Optimizer (MOSHO)

To develop a multi-objective version of SHO, we have introduced two components (see Fig. 8). These components are very similar to those of MOPSO [47]. The first component is an archive which is accountable for storing the best non-dominated Pareto optimal solutions. The second component is a group selection approach to choose nearby solutions corresponding to the position of prey from the archive. These components are described in the preceding subsections.

3.3.1. Archive

The archive is a storage part of all non-dominated Pareto optimal solutions obtained so far. This archive component has an ability to evenly spread around the Pareto front if it is concave, convex, and disconnected. It consists of two main parts such as archive controller and grid.

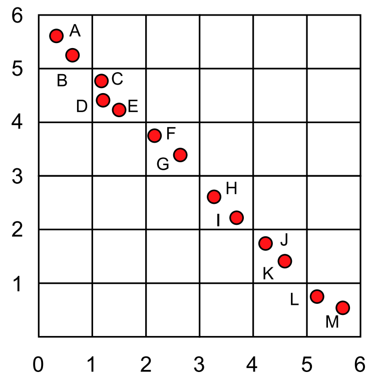


Fig. 3. The adaptive grid mechanism of individuals.

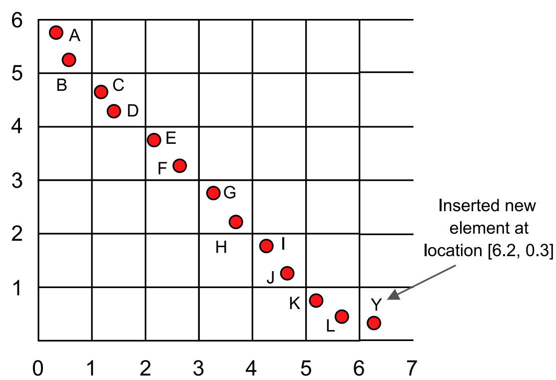


Fig. 4. Relocate the individuals if the new element lies outside the boundary of the grid.

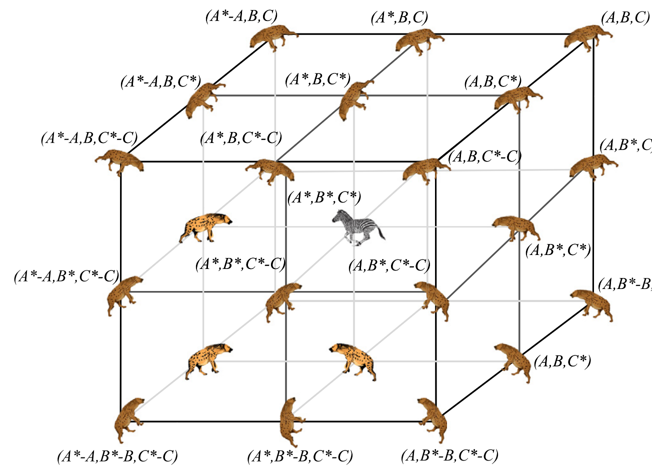


Fig. 5. 3D positions of the spotted hyenas.

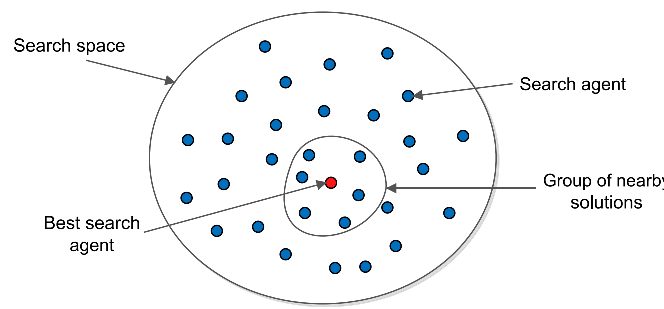


Fig. 6. Group selection mechanism.

3.3.1.1. The archive controller. The main function of this controller is to decide whether the solution should be added or not in the archive. There are some important points regarding the archive update mechanism which are given below:

- The solution should not be allowed to enter the archive when it is dominated by at least one of archive members.
- When a new solution dominates one or more members in the archive, the new solution will be able to enter the archive after omitting the dominated solution(s) in the archive.
- The new member should be added in the archive when neither the new solution nor archive solutions dominate each other.
- The grid method should be run to omit one of the most crowded solution section and insert new solutions to improve the diversity corresponding to Pareto optimal front when the archive is full.

3.3.1.2. The grid. In this work, the adaptive grid mechanism is used to produce the distributed Pareto fronts [48]. The objective function space is divided into different regions as shown in Fig. 3. The grid has to be recalculated and relocate each individual if the inserted individual into population lies outside the current bounds of the grid [47] (see Fig. 4). The adaptive grid is a space formed by hypercubes and is used to distribute in a uniform way.

3.3.1.3. Group selection mechanism. In multi-objective search space, the challenging task is to compare the solutions with archive members. To overcome this issue a group selection mechanism is designed. The group selection strategy chooses the least crowded section of search space and offers one of its non-dominated solutions to group of nearby solutions as shown in Fig. 6. The selection is done on the basis of roulette-wheel method with probability which is defined as:

$$H_k = \frac{f}{S_k} \tag{18}$$

where f is a constant number that should be greater than one and S is the number of Pareto optimal solutions which are obtained so far in the kth segment. The roulette-wheel selection method is a classic selection operator in the proportional type. The fitness value of each individual in the population corresponds to the range on the roulette wheel proportion. In Eq. (17), it can be seen that choosing the number of solutions is directly proportional to vector $\vec{M}$. The proposed MOSHO algorithm (see Algorithm 1) includes all the characteristics of SHO algorithm. The main difference between MOSHO and SHO is that MOSHO searches in a set of archive members, while SHO saves the group of optimal solutions.

The foremost difference between MOSHO and NSGA-2 is archive. The motivation behind the use of archive is to reduce the possibility of deterioration of non-dominated solutions. MOSHO provides optimal solutions as compared to other archive-based algorithms. The archive-based algorithms usually utilize selection operators such as crossover, mutation, etc. These operators bias the search towards the members of archive. In MOSHO, the variables are exchanged among a solution in the search space. This exchange operation increases the exploration capability of MOSHO. Whereas, it reduces the convergence property. To overcome this problem, MOSHO uses a group selection approach to select at least one non-dominated solution in the given search space.

3.4. Computational complexity

In this subsection, the computational complexity of proposed algorithm is discussed. The time and space complexities of MOSHO is described below.

3.4.1. Time complexity

1. Initialization of MOSHO population needs $\mathcal{O}(n_o \times n_p)$ time where n_o represents the number of objectives and n_p represents the number of population size.

Algorithm 1 Multi-objective Spotted Hyena Optimizer (MOSHO).

```
Input: Spotted hyenas population P_i (i = 1, 2, ..., n)
Output: Archive of non-dominated optimal solutions
procedure MOSHO
Initialize the vectors h, B, E, and N
Calculate the objective values of each search agent
Find all the non-dominated solutions and initialize these solutions to archive
P_h = best search agent from archive
C_h = group or cluster of all far optimal solutions with respect to P_h(archive)
    while (x < Max_iterations) do
        for each search agent do
            Update the position of current search agent by Eqs. (13)-(17)
        end for
        Update h, B, E, and N
        Calculate the objective values of all search agents
        Find the non-dominated solutions from updated search agents
        Update the obtained non-dominated solutions to archive
        if archive is full then
            Grid method should be run to omit one of the most crowded archive members
            Add new solution to the archive
        end if
        Check if any search agent goes beyond the search space and then adjust it
        Calculate the objective values of each search agent
        Update P_h if there is a better solution than the previous optimal solution from archive
        Update the group C_h with respect to P_h(archive)
        x = x + 1
    end while
return archive
end procedure
```

2. The fitness calculation of each search agent requires $\mathcal{O}(Max_{iterations} \times n_o \times n_p)$ time where $Max_{iterations}$ is the maximum number of iterations to simulate the proposed MOSHO algorithm.
3. The algorithm requires $\mathcal{O}(M)$ time to define the group of spotted hyenas where M indicates the counting value of spotted hyenas.
4. It requires $\mathcal{O}(n_o \times (n_{ns} + n_p))$ time to update the archive of non-dominated solutions.
5. Repeat Steps 2 to 4 until the termination criteria is satisfactory.

Therefore, the overall time complexity of proposed MOSHO algorithm is $\mathcal{O}(Max_{iterations} \times n_o \times (n_p + n_{ns}) \times M)$.

The computational complexities corresponding to MOPSO, NSGA-2, PESA-2, SPEA-2, and MOACO are $\mathcal{O}(Max_{iterations} \times n_o \times n_p^2)$, $\mathcal{O}(Max_{iterations} \times n_o \times n_p^2)$, $\mathcal{O}(Max_{iterations} \times n_o \times (n_p + n_{ns})^2)$, $\mathcal{O}(Max_{iterations} \times n_o \times n_p^2)$, and $\mathcal{O}(Max_{iterations} \times n_o \times n_p^2)$, respectively. Whereas, the computational complexity of MOEA/D is $\mathcal{O}(Max_{iterations} \times n_o \times n_p \times T)$, where T represents the Tchebycheff approach to decompose the optimization problem into subproblems.

3.4.2. Space complexity

The space complexity of MOSHO algorithm is considered during its initialization process which requires space at any one time. Hence, the total space complexity of MOSHO algorithm is $\mathcal{O}(n_o \times n_p)$.

4. Experimental results and discussions

This section compares the performance of MOSHO with six recently developed algorithms and validates over 24 benchmark test functions.

4.1. Benchmark test functions

The twenty four well-known benchmark test functions are applied on MOSHO algorithm to demonstrate its efficiency. The CEC-2009 special session benchmark test suit [24] is used which consists of ten unconstrained test functions as described in Appendix A. These unconstrained functions are able to solve the bound constrainted multi-objective problems.

The popular five ZDT [25] and nine DTLZ [26] benchmark test suites are also utilized which are described in the Appendices B and C, respectively. These unconstrained functions involve a feature to converge the Pareto optimal front very efficiently.

In Appendix A, the characteristics of CEC-2009 benchmark test functions are described. There are ten unconstrained test functions ($UF1 - UF10$) which are included in this test suite.

The detailed characteristics of ZDT and DTLZ unconstrained test functions are described in Appendices B and C, respectively. There are five ZDT test functions ($ZDT1 - ZDT6$) and nine DTLZ test functions ($DTLZ1 - DTLZ9$).

4.2. Experimental setup

To validate the performance of MOSHO algorithm, it is compared with six well-known optimization algorithms such as Multi-objective Particle Swarm Optimization (MOPSO) [21], Non-dominated Sorting Genetic Algorithm 2 (NSGA-2) [20], Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) [22], Pareto Envelope-based Selection Algorithm (PESA-2) [49], Strength Pareto Evolutionary Algorithm (SPEA-2) [28], and Multi-objective Ant Colony Optimization (MOACO) [29].

The parameter settings of these algorithms are set as they are recommended in their original papers. The following initial parameters for MOPSO are chosen as [21]:

- $\phi_a = \phi_b = 2.05$
- $\phi_f = \phi_a + \phi_b$
- Inertia weight: $w = \dfrac{2}{\phi_f - 2 + \sqrt{\phi_f^2 - 4\phi_f}}$
- Personal coefficient: $c_1 = \chi * \phi_a$
- Social coefficient: $c_2 = \chi * \phi_b$
- Grid inflation parameter: $\alpha = 0.1$
- Leader selection pressure parameter: $\beta = 4$
- Number of grids: $Grid_{number} = 10$

For NSGA-2, the following initial parameters are chosen as [34]:

- Population size $(X) = 100$
- Cross over probability $P_c = 0.8$
- Mutation probability $P_m = 0.1$

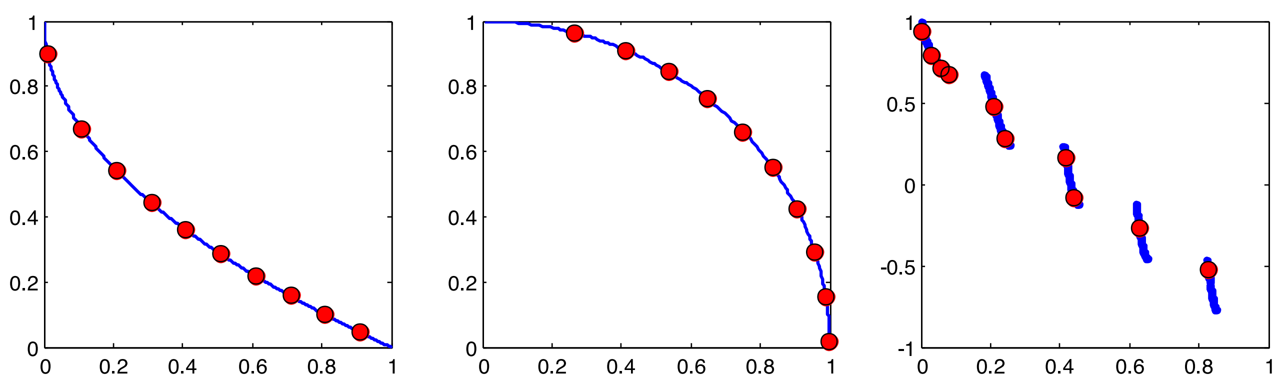
For MOEA/D, the following initial parameters are decided as [22]:

- Subproblems: $N = 100$
- Number of neighbours: $T = 0.1{*}N$
- Updated new child maximal copies: $M = 0.01{*}N$
- Probability of selecting parents: $P_p = 0.9$
- Mutation rates: $M_r = 0.5$
- Distribution index: $D_i = 30$

The following initial parameters are chosen for PESA-2 and SPEA-2 [34]:

Table 1
Archive state in different generations for ZDT1, DTLZ6, and ZDT3 test functions using MOSHO.

Iterations	ZDT1 (*Concave*)				DTLZ6 (*Convex*)				ZDT3 (*Disconnected*)			
	Archive		Objective value		Archive		Objective value		Archive		Objective value	
	x	y	f_1	f_2	x	y	f_1	f_2	x	y	f_1	f_2
1	0.999	0.015	0.861	0.937	0.256	3.616	0.648	0.382	0.999	0.017	0.459	0.045
	0.987	0.156			0.005	5.989			0.987	0.170		
	0.955	0.294			0.005	5.989			0.956	0.290		
	0.904	0.425			0.030	5.337			0.907	0.421		
	0.835	0.549			0.030	5.337			0.842	0.535		
	0.750	0.661			0.005	5.989			0.753	0.655		
	0.649	0.760			0.090	4.112			0.643	0.751		
	0.535	0.844			0.030	5.337			0.436	0.840		
	0.411	0.911			0.256	3.616			0.407	0.900		
	0.263	0.964			0.030	5.337			0.269	0.960		
50	0.842	0.010	0.762	0.753	0.136	2.025	0.590	0.233	0.825	0.011	0.301	0.040
	0.862	0.140			0.023	3.234			0.762	0.267		
	0.823	0.118			0.007	4.023			0.882	0.243		
	0.843	0.346			0.039	4.237			0.752	0.376		
	0.705	0.413			0.039	3.007			0.755	0.341		
	0.639	0.519			0.085	4.462			0.697	0.528		
	0.556	0.431			0.128	3.096			0.763	0.428		
	0.448	0.746			0.025	3.039			0.340	0.774		
	0.401	0.831			0.223	2.219			0.386	0.841		
	0.248	0.763			0.010	4.414			0.202	0.741		
100	0.018	0.900	0.801	0.566	0.999	0.157	0.568	0.967	0.004	0.934	0.284	0.914
	0.117	0.670			0.987	0.156			0.032	0.793		
	0.215	0.541			0.955	0.294			0.055	0.708		
	0.316	0.442			0.904	0.425			0.082	0.669		
	0.418	0.359			0.835	0.549			0.209	0.479		
	0.519	0.285			0.750	0.661			0.240	0.278		
	0.612	0.218			0.649	0.760			0.415	0.164		
	0.714	0.157			0.535	0.844			0.439	−0.081		
	0.818	0.100			0.411	0.911			0.627	−0.273		
	0.916	0.045			0.263	0.964			0.826	−0.524		



(a) (b) (c)

Fig. 7. Convergence analysis of proposed archive on (a) Concave, (b) Convex, and (c) Disconnected Pareto fronts.

- Cross over probability $P_c = 0.8$
- Distribution index for SBX = 15
- Mutation probability $P_m = 1/n$
- Polynomial mutation of distribution index = 20

For MOACO, the following initial parameters are chosen as [50]:

- Initial pheromone = 1.0E-06
- Pheromone update constant $Q = 20$
- Exploration constant $q_0 = 1$
- Global pheromone decay rate = 0.9
- local pheromone decay rate = 0.5
- $\alpha = 0.5$
- $\beta = 2.5$

The population size and maximum number of iterations for above-mentioned algorithms are set to 100 and 1000, respectively.

Due to stochastic nature of these algorithms, the results are averaged over 30 independent runs under 30 different random seeds. The mean best-of-run solution and standard deviation of best solution in the last iteration are reported in tables.

4.3. Performance metrics

The four well-known performance metrics are used to investigate the performance of above-mentioned algorithms. These are Hypervolume (HV) [51,52], $\Delta_p (p = 1)$ [53–55], Spread [52,56], and Epsilon (ε) [52,57].

The best Pareto optimal solutions set which is obtained by each algorithm is compared with the proposed technique. The above-mentioned algorithms run 30 times on 24 benchmark test problems.

Table 2
Results of the multi-objective algorithms on the CEC-2009 (UF1 - UF10) test functions.

F	Performance	MOSHO		MOPSO		NSGA-2		MOEA/D		PESA-2		SPEA-2		MOACO	
	Metrics	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
UF1	*Hypervolume*	**4.69E−02**	**7.30E−04**	4.80E−01	2.04E−02	5.10E−01	4.36E−02	6.61E−01	2.80E−03	4.83E−01	5.97E−02	2.75E−01	1.20E−01	1.46E−01	7.81E−02
	Δ_p	**4.07E−09**	**8.11E−11**	3.21E−04	1.11E−04	4.44E−03	1.35E−03	2.67E−04	1.04E−04	6.42E−03	2.73E−03	1.60E−02	1.69E−03	3.67E−03	2.91E−03
	Spread	**1.27E−01**	**3.12E−02**	9.99E−01	2.13E−01	1.50E+00	1.24E−01	4.14E−01	1.85E−01	1.41E+00	1.65E−01	1.05E+00	1.05E−01	1.01E+00	2.03E−01
	Epsilon	**5.90E−03**	**1.11E−03**	2.06E−01	3.10E−02	2.18E−01	5.50E−02	1.19E−02	1.17E−02	2.76E−01	1.04E−01	6.58E−01	1.69E−01	2.83E−01	1.20E−01
UF2	*Hypervolume*	**1.07E−01**	**1.93E−03**	6.12E−01	2.95E−03	6.10E−01	7.45E−03	6.51E−01	3.89E−03	5.94E−01	1.65E−02	5.27E−01	1.07E−01	4.16E−01	2.28E−01
	Δ_p	**4.13E−04**	**2.10E−04**	1.51E−03	3.86E−05	6.37E−03	5.63E−04	3.45E−04	2.75E−04	5.63E−03	5.86E−04	3.21E−03	2.11E−03	3.10E−03	2.98E−03
	Spread	**4.09E−02**	**2.93E−02**	4.98E−01	1.23E−01	6.04E−01	4.70E−02	5.43E−01	6.76E−02	8.86E−01	8.12E−02	1.04E+00	1.53E−01	2.88E−01	1.49E−01
	Epsilon	**2.05E−02**	**5.19E−04**	9.23E−02	9.73E−03	1.41E−01	3.68E−02	6.85E−02	1.51E−02	1.79E−01	5.61E−02	3.35E−01	1.50E−01	4.71E−02	3.43E−03
UF3	*Hypervolume*	5.09E−01	3.70E−02	3.96E−01	3.51E−02	3.24E−01	4.40E−02	6.14E−01	2.24E−02	3.55E−01	3.36E−02	**3.00E−01**	3.75E−02	4.74E−01	**1.71E−02**
	Δ_p	2.63E−03	1.61E−03	5.88E−03	**4.31E−04**	8.96E−03	1.60E−03	**1.97E−03**	5.01E−04	1.12E−02	1.89E−03	1.69E−02	1.99E−03	2.88E−02	1.40E−03
	Spread	8.02E−01	6.99E−02	**6.86E−01**	1.05E−01	1.02E+00	5.47E−02	7.03E−01	2.04E−01	1.10E+00	1.03E−01	1.09E+00	9.40E−02	3.48E+00	**2.77E−02**
	Epsilon	2.50E−01	4.11E−02	1.95E−01	**2.10E−02**	3.84E−01	5.94E−02	**1.48E−01**	6.18E−02	4.78E−01	6.32E−02	5.55E−01	8.06E−02	3.96E−01	4.48E−02
UF4	*Hypervolume*	**1.00E−01**	**2.32E−03**	2.40E−01	7.71E−03	2.46E−01	3.19E−03	2.37E−01	1.01E−02	2.40E−01	3.13E−03	1.52E−01	6.50E−02	2.41E−01	2.97E−02
	Δ_p	**4.15E−04**	**2.69E−05**	2.66E−03	1.04E−04	1.31E−03	4.45E−05	3.89E−03	5.69E−04	2.39E−03	1.32E−04	5.60E−03	3.64E−03	4.99E−03	2.81E−04
	Spread	**1.00E−01**	**3.13E−02**	4.01E−01	6.83E−02	5.15E−01	5.11E−02	5.02E−01	9.46E−02	8.24E−01	5.32E−02	1.09E+00	1.20E−01	2.74E−01	4.24E−02
	Epsilon	**1.10E−02**	**3.33E−03**	7.48E−02	6.03E−03	6.99E−02	9.11E−03	7.80E−02	9.14E−03	8.39E−02	8.61E−03	3.63E−01	3.10E−01	4.71E−02	6.85E−03
UF5	*Hypervolume*	**2.62E−03**	**1.23E−02**	3.51E−03	1.57E−02	2.70E−02	4.06E−02	4.72E−02	5.74E−02	6.20E−02	6.01E−02	3.96E−02	4.77E−02	4.82E−02	2.09E−02
	Δ_p	**1.05E−01**	**1.08E−02**	2.63E−01	1.50E−01	2.89E−01	3.89E−02	1.07E−01	2.91E−02	3.10E−01	1.38E−02	1.35E−01	2.21E−02	2.42E−01	3.85E−02
	Spread	**1.90E−01**	**4.78E−02**	8.71E−01	9.50E−02	1.26E+00	9.95E−02	1.20E+00	5.02E−02	1.14E+00	1.34E−01	1.13E+00	8.40E−02	1.21E+00	5.72E−02
	Epsilon	**1.11E−01**	**1.00E−01**	1.28E+00	5.28E−01	6.93E−01	1.89E−01	6.41E−01	1.43E−01	7.82E−01	1.76E−01	9.71E−01	2.21E−01	4.90E−01	3.47E−01
UF6	*Hypervolume*	**1.60E−02**	**1.38E−02**	2.02E−02	2.56E−02	1.96E−01	6.99E−02	2.08E−01	8.26E−02	2.09E−01	6.55E−02	1.15E−01	8.61E−02	2.87E−01	3.75E−02
	Δ_p	**4.31E−03**	**1.60E−03**	2.56E−03	1.56E−03	6.78E−03	2.11E−03	4.14E−03	1.67E−03	1.94E−02	3.79E−03	2.21E−02	2.45E−03	1.71E−02	8.29E−03
	Spread	**7.09E−01**	**4.40E−02**	9.79E−01	6.74E−02	1.19E+00	1.24E−01	1.14E+00	9.47E−02	1.27E+00	2.03E−01	1.10E+00	8.48E−02	2.11E+00	4.66E−02
	Epsilon	**2.02E−01**	**1.20E−01**	5.95E−01	1.82E−01	4.22E−01	1.49E−01	3.55E−01	1.48E−01	5.48E−01	1.91E−01	8.70E−01	2.67E−01	5.39E−01	2.09E−01
UF7	*Hypervolume*	2.33E−01	6.36E−02	3.95E−01	3.57E−02	3.29E−01	8.89E−02	4.91E−01	**3.34E−03**	2.49E−01	8.82E−02	**1.50E−01**	8.80E−02	2.86E−01	3.99E−02
	Δ_p	3.61E−03	1.45E−03	6.22E−03	2.67E−03	7.78E−03	2.06E−03	**1.51E−04**	**2.44E−05**	6.32E−02	2.71E−03	3.67E−02	3.30E−03	4.82E−02	1.75E−03
	Spread	**2.81E−01**	**2.17E−02**	8.52E−01	6.63E−02	1.07E+00	2.11E−01	3.49E−01	1.75E−01	1.18E+00	8.77E−02	1.01E+00	2.58E−02	4.88E−01	2.67E−02
	Epsilon	**2.05E−02**	**1.78E−02**	1.82E−01	1.07E−01	4.28E−01	2.42E−01	4.40E−02	3.17E−02	6.08E−01	2.19E−01	8.01E−01	1.37E−01	4.73E−01	2.81E−01
UF8	*Hypervolume*	**1.60E−02**	**2.47E−02**	3.05E−02	2.69E−02	1.16E−01	4.07E−02	1.81E−01	1.01E−01	1.93E−02	3.32E−02	1.29E−01	8.96E−02	1.00E−01	2.79E−02
	Δ_p	3.34E−03	6.11E−04	2.31E−03	2.01E−04	**2.81E−03**	**4.45E−04**	3.16E−01	5.17E−03	5.11E−03	4.86E−04	4.87E−03	9.09E−04	5.95E−03	2.86E−03
	Spread	6.07E−01	5.13E−01	8.27E−01	8.20E−02	7.77E−01	**8.05E−02**	**4.69E−01**	7.90E−01	7.07E−01	1.11E−01	8.62E−01	2.29E−01	5.79E−01	1.58E−01
	Epsilon	8.03E−01	3.14E−01	7.69E−01	8.51E−02	**6.24E−01**	1.11E−01	7.01E−01	9.54E−01	9.70E−01	**5.22E−02**	8.05E−01	1.31E−01	6.97E−01	3.66E−01
UF9	*Hypervolume*	4.09E−02	6.33E−01	6.89E−02	5.70E−02	2.54E−01	6.17E−02	**2.76E−02**	4.47E−01	9.43E−02	3.51E−02	4.57E−01	4.78E−02	3.39E−01	**2.11E−02**
	Δ_p	4.39E−03	6.10E−04	2.68E−03	3.27E−04	5.35E−03	2.51E−04	6.90E−03	4.48E−03	4.78E−03	3.31E−04	**1.20E−03**	**1.04E−04**	3.23E−03	1.76E−03
	Spread	**1.01E−01**	7.37E−02	8.21E−01	5.05E−02	8.52E−01	6.83E−02	6.63E−01	6.41E−02	7.55E−01	7.45E−02	7.36E−01	6.10E−02	4.59E−01	**3.46E−02**
	Epsilon	**3.03E−01**	5.15E−02	8.21E−01	1.42E−01	5.14E−01	6.46E−02	6.56E−01	5.77E−02	8.95E−01	4.16E−02	4.81E−01	**2.92E−02**	6.85E−01	5.68E−02
UF10	*Hypervolume*	4.60E−02	1.31E−02	0.00E+00	0.00E+00	**4.53E−03**	**1.16E−02**	6.90E−02	5.11E−01	4.05E−02	2.97E−02	3.38E−02	2.86E−02	4.90E−02	3.00E−02
	Δ_p	6.78E−03	4.91E−03	3.79E−02	6.73E−03	3.13E−03	3.00E−03	4.01E−03	2.81E−03	**1.83E−03**	4.18E−04	6.50E−03	**4.47E−04**	4.22E−03	1.46E−03
	Spread	**2.62E−01**	5.82E−02	6.62E−01	**5.07E−02**	7.66E−01	6.28E−02	5.21E−01	8.60E−01	9.18E−01	9.13E−02	1.24E+00	2.23E−01	4.53E−01	1.86E−01
	Epsilon	**4.61E−01**	5.53E−01	1.82E+00	2.45E−01	1.05E+00	1.55E−01	5.33E−01	7.17E−01	8.86E−01	1.07E−01	8.68E−01	**1.03E−01**	6.00E−01	2.85E−01

Table 3
Results of the multi-objective algorithms on the ZDT benchmark (ZDT1 – ZDT6) test functions.

F	Performance	MOSHO		MOPSO		NSGA-2		MOEA/D		PESA-2		SPEA-2		MOACO	
	Metrics	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
ZDT1	*Hypervolume*	**6.56E−01**	3.82E−04	6.62E−01	**1.40E−04**	6.60E−01	2.62E−04	6.64E−01	3.56E−04	6.56E−01	8.58E−04	6.60E−01	3.05E−04	6.60E−01	2.46E−04
	Δ_p	3.34E−04	4.04E−05	2.33E−04	**5.00E−07**	1.18E−04	3.37E−06	**6.32E−05**	2.44E−06	4.03E−04	1.44E−04	2.11E−04	3.21E−06	4.66E−04	2.00E−06
	Spread	**2.30E−02**	**1.11E−03**	6.87E−02	1.04E−02	3.77E−01	2.58E−02	2.85E−01	2.82E−03	6.38E−01	3.78E−02	1.43E−01	1.60E−02	2.31E−01	2.20E−02
	Epsilon	5.08E−02	6.16E−02	5.52E−03	**1.58E−04**	1.34E−02	2.00E−03	**3.01E−03**	3.82E−04	2.58E−02	1.18E−02	8.90E−03	6.68E−04	7.08E−02	3.41E−04
ZDT2	*Hypervolume*	3.29E−01	**3.71E−05**	3.28E−01	8.09E−05	3.26E−01	3.11E−04	3.31E−01	1.84E−04	**3.24E−01**	6.54E−04	3.26E−01	6.14E−04	3.27E−01	2.09E−04
	Δ_p	**2.01E−05**	**1.34E−06**	1.45E−04	2.57E−06	1.37E−04	3.41E−05	4.98E−05	6.00E−06	3.52E−04	2.35E−05	2.22E−04	4.21E−06	1.30E−04	2.76E−06
	Spread	2.02E−01	9.10E−02	**6.68E−02**	**7.85E−03**	3.70E−01	4.46E−02	1.45E−01	1.21E−02	6.76E−01	4.20E−02	1.49E−01	1.46E−02	2.56E−01	3.87E−02
	Epsilon	**1.20E−03**	2.19E−04	5.46E−03	**1.37E−04**	1.27E−02	2.30E−03	2.68E−03	3.62E−04	3.29E−02	1.88E−02	8.49E−03	6.16E−04	3.07E−03	8.73E−04
ZDT3	*Hypervolume*	5.11E−01	2.22E−04	5.15E−01	3.87E−04	5.15E−01	1.58E−04	5.16E−01	**3.09E−05**	**5.10E−01**	2.80E−03	5.14E−01	2.31E−04	5.13E−01	5.80E−04
	Δ_p	**4.13E−05**	3.31E−06	2.57E−04	**2.08E−06**	1.33E−04	4.49E−06	4.14E−05	5.55E−06	1.00E−03	5.29E−04	3.21E−04	2.88E−05	2.40E−04	4.99E−05
	Spread	**1.00E−01**	**2.17E−03**	7.05E−01	3.38E−03	7.42E−01	1.32E−02	1.03E+00	2.47E−03	9.02E−01	2.62E−02	7.09E−01	4.74E−03	6.06E−01	2.53E−03
	Epsilon	**4.91E−03**	6.74E−04	5.71E−03	5.49E−04	8.87E−03	1.56E−03	5.33E−03	**1.27E−04**	1.30E−01	1.54E−01	9.60E−03	1.14E−03	5.94E−03	2.48E−03
ZDT4	*Hypervolume*	6.60E−01	4.44E−04	**0.00E+00**	**0.00E+00**	6.54E−01	3.53E−03	6.65E−01	4.95E−05	6.50E−01	4.64E−03	6.49E−01	8.60E−03	6.45E−01	3.46E−02
	Δ_p	4.30E−04	1.80E−04	3.41E−01	2.38E−02	3.78E−04	5.99E−05	**4.18E−05**	**3.04E−05**	3.10E−04	1.10E−04	4.71E−03	1.20E−03	1.69E−04	1.00E−04
	Spread	**2.02E−01**	**1.98E−03**	8.86E−01	7.18E−02	3.98E−01	3.61E−02	2.85E−01	2.00E−03	9.10E−01	2.88E−01	3.06E−01	1.36E−01	2.01E−01	1.07E−02
	Epsilon	**2.09E−03**	**1.00E−04**	5.14E+00	1.91E+00	1.76E−02	8.12E−03	2.94E−03	1.03E−04	2.87E−02	1.06E−02	6.37E−02	5.29E−02	3.28E−02	2.02E−02
ZDT6	*Hypervolume*	**3.19E−01**	4.33E−05	4.01E−01	5.40E−05	3.88E−01	1.46E−03	4.05E−01	**6.11E−08**	3.91E−01	1.34E−03	3.78E−01	2.70E−03	2.26E−01	1.86E−03
	Δ_p	2.70E−04	4.11E−05	2.31E−04	2.09E−06	4.81E−04	7.07E−05	**3.61E−05**	**6.41E−08**	3.07E−04	3.80E−05	4.95E−04	3.37E−05	3.27E−04	1.30E−05
	Spread	**1.11E−01**	**1.24E−04**	8.48E−01	4.79E−01	3.66E−01	3.99E−02	1.50E−01	1.86E−04	7.59E−01	1.93E−01	2.38E−01	3.08E−02	3.47E−01	1.94E−02
	Epsilon	**1.20E−03**	3.13E−04	4.86E−03	5.16E−04	1.58E−02	2.06E−03	1.64E−03	1.43E−01	1.89E−02	3.66E−03	2.56E−02	**5.35E−06**	3.00E−02	2.17E−04

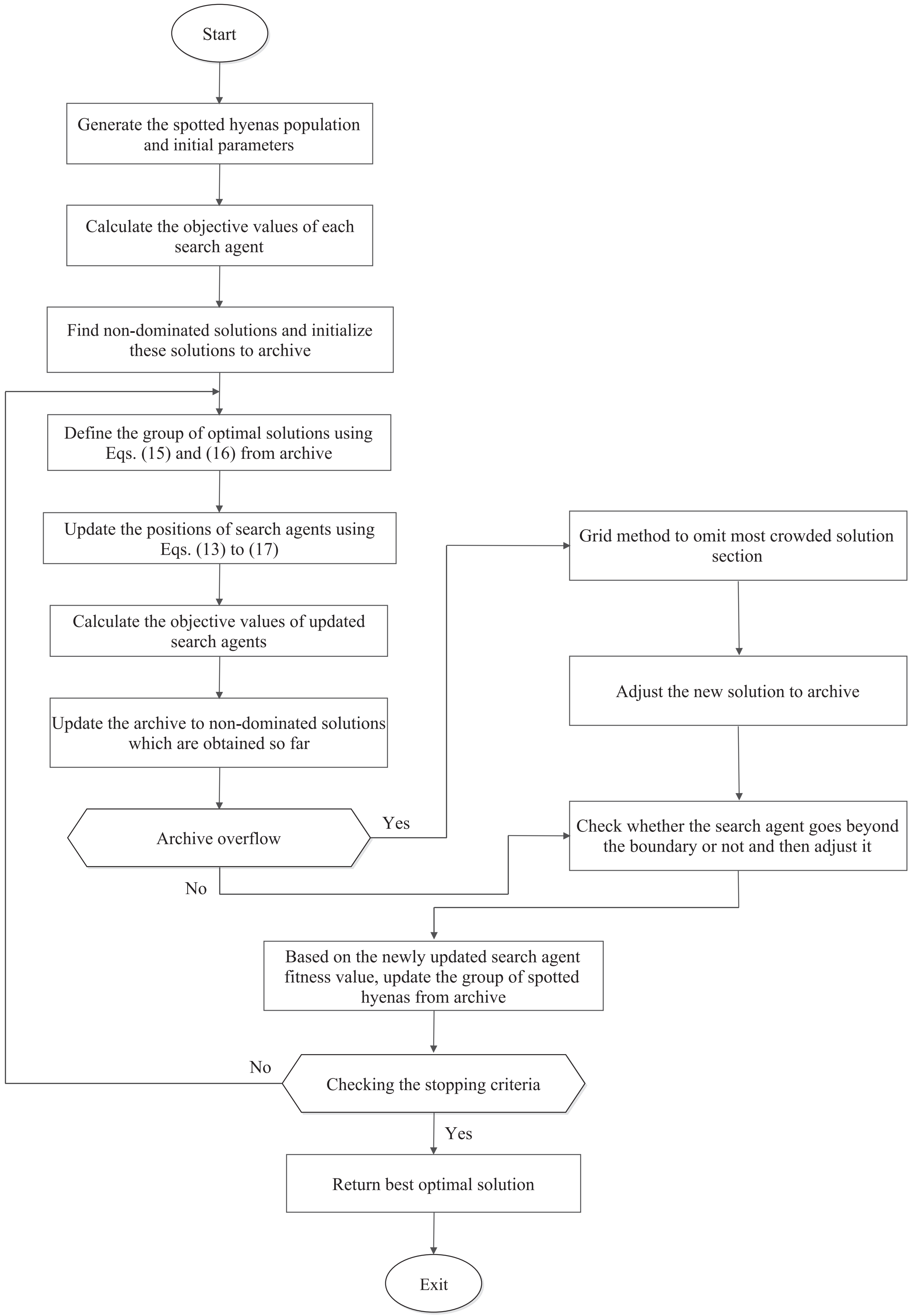


Fig. 8. Flowchart of the proposed MOSHO algorithm.

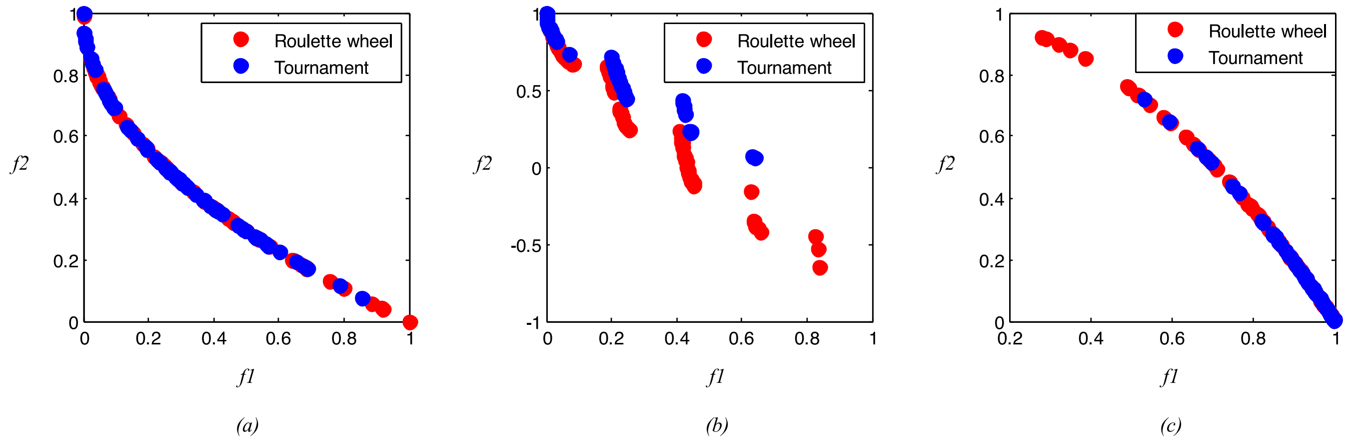


Fig. 9. Convergence analysis of selection approaches on (a) ZDT1, (b) ZDT3, and (c) ZDT6 test functions.

Table 4
Results of the multi-objective algorithms on the DTLZ benchmark (DTLZ1 - DTLZ9) test functions.

F	Performance	MOSHO		MOPSO		NSGA-2		MOEA/D		PESA-2		SPEA-2		MOACO	
	Metrics	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
DTLZ1	*Hypervolume*	4.60E-01	4.33E-02	3.63E-01	5.56E-02	6.62E-01	3.95E-01	**3.56E-01**	**5.94E-03**	5.93E-01	3.73E-02	7.56E-01	1.97E-02	4.11E-01	2.00E-02
	Δ_p	**3.45E-04**	1.11E-05	2.53E-03	2.49E-04	3.78E-03	3.81E-04	4.24E-04	5.40E-05	5.00E-03	2.00E-04	4.46E-04	**7.21E-06**	5.79E-04	4.11E-04
	Spread	**1.33E-01**	3.17E-01	5.95E-01	2.93E-01	2.02E+00	**2.66E-01**	2.85E-01	4.24E-01	3.72E+00	3.89E-01	9.15E-01	2.97E-01	4.97E-01	1.08E-01
	Epsilon	3.60E-03	5.83E-02	3.96E-02	3.93E-02	2.10E-02	5.23E-01	4.35E-02	3.73E-02	**2.04E-03**	2.67E-01	5.39E-02	**1.49E-02**	2.07E-02	2.00E-02
DTLZ2	*Hypervolume*	**1.09E-02**	**1.07E-03**	1.94E-01	2.63E-03	2.63E-02	3.96E-03	3.08E-01	5.23E-02	5.70E-02	1.12E-02	4.04E-01	1.87E-03	3.97E-01	2.05E-03
	Δ_p	**3.54E-04**	**3.11E-05**	3.48E-03	8.06E-04	3.67E-04	2.00E-05	6.90E-04	6.57E-05	3.19E-02	5.39E-03	4.73E-04	4.11E-05	4.95E-04	4.06E-05
	Spread	**3.30E-01**	**5.78E-03**	6.71E-01	2.73E-01	4.83E-01	7.54E-03	7.40E-01	3.30E-02	7.56E-01	5.73E-01	5.46E-01	3.05E-02	4.83E-01	6.28E-02
	Epsilon	**3.69E-02**	6.16E-02	7.73E-02	**8.33E-03**	2.37E-01	8.08E-02	5.05E-02	3.57E-01	3.73E-01	4.73E-01	8.41E-02	1.13E-02	6.05E-02	4.74E-02
DTLZ3	*Hypervolume*	2.09E-02	8.37E-03	3.63E-01	2.63E-02	6.24E-02	**5.25E-04**	3.74E-01	2.74E-02	5.23E-01	2.35E-03	**0.00E+00**	**0.00E+00**	4.03E-01	4.46E-03
	Δ_p	**3.47E-04**	**2.10E-04**	5.22E-03	3.14E-03	5.05E-04	3.33E-04	2.49E-03	6.41E-04	3.03E-01	3.90E-02	1.00E-01	4.59E-02	3.04E-02	2.41E-02
	Spread	**1.39E-02**	3.33E-02	5.73E-01	3.73E-01	1.32E+00	6.34E-02	4.63E-01	**2.63E-02**	2.43E-02	4.03E-01	1.20E+00	1.09E-01	2.92E-01	1.63E-01
	Epsilon	**2.09E-02**	**3.56E-02**	3.74E-01	4.38E-02	5.73E-01	5.77E-02	3.84E-02	6.63E-02	3.22E-01	8.74E-02	6.37E+00	2.70E+00	3.20E-01	2.83E-02
DTLZ4	*Hypervolume*	2.60E-02	1.31E-02	2.74E-01	**3.73E-03**	3.45E-02	6.89E-02	2.74E-01	3.63E-02	**4.53E-03**	5.23E-03	3.12E-01	9.77E-02	4.96E-02	2.38E-02
	Δ_p	2.33E-04	4.11E-05	4.61E-02	2.00E-04	3.69E-02	5.31E-03	**2.14E-04**	1.09E-04	4.79E-02	6.49E-03	4.35E-03	3.20E-03	3.16E-02	4.92E-03
	Spread	2.78E-02	5.11E-02	3.08E-01	3.53E-02	5.73E-02	6.63E-01	6.64E-01	**3.21E-02**	**4.23E-03**	6.66E-02	4.68E-01	1.29E-01	5.05E-02	4.80E-02
	Epsilon	5.05E-02	8.70E-02	5.13E-02	6.63E-02	**6.73E-03**	9.14E-03	3.63E-02	9.74E-02	3.53E-01	**8.33E-03**	3.23E-01	2.86E-01	4.60E-02	3.94E-02
DTLZ5	*Hypervolume*	**1.49E-03**	**1.32E-04**	1.63E-03	2.83E-04	3.13E-02	5.06E-02	6.05E-01	3.83E-02	5.93E-02	5.23E-02	1.96E-01	2.47E-02	2.42E-02	1.99E-02
	Δ_p	**1.46E-03**	**2.81E-04**	1.35E-01	7.36E-02	3.21E-01	1.73E-02	5.64E-03	6.88E-04	4.73E-02	2.32E-02	3.77E-01	2.31E-02	2.90E-02	1.04E-03
	Spread	3.88E-01	5.00E-02	5.63E-01	5.73E-02	1.36E+00	7.96E-03	6.27E-01	5.22E-02	1.04E+00	**4.14E-04**	**2.43E-02**	5.47E-02	8.33E-01	4.29E-02
	Epsilon	**2.03E-01**	2.78E-01	3.73E+00	5.28E-01	5.23E-01	1.69E-01	6.94E-01	5.92E-01	5.82E-01	**1.02E-01**	7.75E-01	3.20E-01	4.06E-01	1.95E-01
DTLZ6	*Hypervolume*	**2.63E-03**	**1.11E-03**	3.02E-03	2.56E-02	1.92E-01	7.83E-02	5.04E-02	3.16E-03	3.09E-01	5.55E-02	1.05E-01	6.61E-02	2.53E-02	2.00E-02
	Δ_p	**3.00E-03**	**2.67E-03**	2.33E-02	3.66E-03	3.50E-02	5.12E-03	5.21E-03	2.51E-03	3.67E-02	6.36E-03	3.11E-02	3.10E-03	2.10E-02	4.96E-03
	Spread	3.45E-01	6.77E-02	6.09E-01	**3.32E-02**	1.53E+00	1.94E-01	4.10E-01	5.07E-01	**1.26E-01**	2.63E-01	2.10E+00	7.18E-02	3.53E-01	4.75E-02
	Epsilon	**2.47E-03**	7.16E-03	6.95E-01	1.82E-03	4.63E-03	3.63E-01	3.65E-02	**3.48E-04**	3.43E-01	6.36E-01	7.72E-01	2.73E-01	4.34E-02	6.80E-03
DTLZ7	*Hypervolume*	2.09E-02	1.32E-02	3.53E-01	3.63E-02	6.29E-03	**3.09E-04**	4.11E-01	3.83E-03	**3.49E-03**	5.82E-01	2.51E-01	8.38E-02	5.03E-02	7.72E-03
	Δ_p	4.10E-03	3.77E-04	5.50E-03	**1.13E-03**	6.98E-03	3.48E-04	**3.38E-03**	3.27E-03	4.91E-02	6.94E-03	1.89E-02	2.81E-03	6.42E-03	3.92E-03
	Spread	4.67E-02	7.17E-02	6.31E-01	8.04E-02	1.83E+00	2.53E-03	4.35E-01	3.37E-01	**3.73E-03**	**3.27E-04**	1.03E-01	3.91E-02	2.28E-02	5.40E-03
	Epsilon	7.78E-02	5.18E-02	8.63E-01	1.07E-01	**3.28E-02**	**3.74E-04**	3.63E-02	3.35E-02	3.08E-01	3.33E-02	4.04E-01	1.93E-01	2.28E-01	4.32E-02
DTLZ8	*Hypervolume*	7.60E-03	2.47E-02	**4.50E-03**	**2.69E-04**	1.84E-01	3.94E-02	4.88E-03	2.01E-02	2.03E-02	4.23E-02	2.98E-01	6.93E-02	1.39E-01	2.76E-03
	Δ_p	6.46E-03	2.81E-04	**3.18E-04**	1.00E-03	5.03E-03	2.68E-03	4.75E-02	2.30E-03	4.69E-03	**6.68E-04**	3.11E-03	1.47E-03	4.53E-03	8.95E-04
	Spread	3.87E-02	7.77E-02	3.27E-01	3.20E-02	**2.37E-03**	8.53E-03	3.39E-01	3.90E-01	5.07E-02	3.14E-02	5.02E-02	2.99E-02	1.37E-02	**4.05E-03**
	Epsilon	1.01E-02	6.19E-02	3.69E-01	3.51E-02	**6.25E-03**	**1.24E-03**	3.01E-01	4.23E-01	3.24E-02	2.22E-02	6.09E-01	1.82E-01	2.23E-02	6.97E-03
DTLZ9	*Hypervolume*	**1.60E-03**	**2.36E-02**	6.89E-02	5.70E-02	2.35E-03	6.17E-02	3.76E-02	4.47E-01	5.43E-02	3.51E-02	3.83E-01	2.94E-02	1.98E-01	8.76E-02
	Δ_p	**1.47E-03**	**3.62E-04**	5.47E-03	3.58E-03	6.77E-02	4.33E-03	3.24E-03	2.87E-03	6.89E-03	4.30E-04	5.47E-03	3.91E-04	7.04E-03	9.43E-04
	Spread	9.87E-02	3.78E-02	4.21E-01	8.88E-02	**4.52E-03**	3.83E-01	6.76E-01	6.31E-02	3.55E-02	**8.45E-03**	4.16E-01	4.94E-01	5.09E-02	9.81E-03
	Epsilon	9.58E-02	4.99E-02	4.21E-01	8.42E-01	**7.73E-03**	**6.93E-03**	6.06E-01	7.77E-02	7.67E-02	6.96E-01	3.41E-02	2.44E-02	4.38E-02	3.79E-02

4.4. Performance evaluation

This subsection discuss the results of proposed MOSHO algorithm on CEC-2009, ZDT, and DTLZ benchmark test suites.

4.4.1. Results on CEC-2009 ($UF1 - UF10$) benchmark test functions

Table 2 shows the performance comparison of the proposed MOSHO with the above-mentioned existing techniques for CEC-2009 benchmark test problems. The results reveal that MOSHO is able to determine the optimal solutions for *UF*1, *UF*2, *UF*4, *UF*5, and *UF*6 test functions. For *UF*3 test function, MOPSO provides better value of Spread than the others. MOEA/D provides better results than the other competitive algorithms in terms of Δ_p and Epsilon. For *UF*7, *UF*9, and *UF*10 test functions, MOSHO provides competitive results in comparison with the others. For *UF*8 test function, NSGA-2 gives the better value of Δ_p and Epsilon as compared with other algorithms. The Pareto optimal solutions of MOSHO on *UF* benchmark problems are illustrated in Fig. 10.

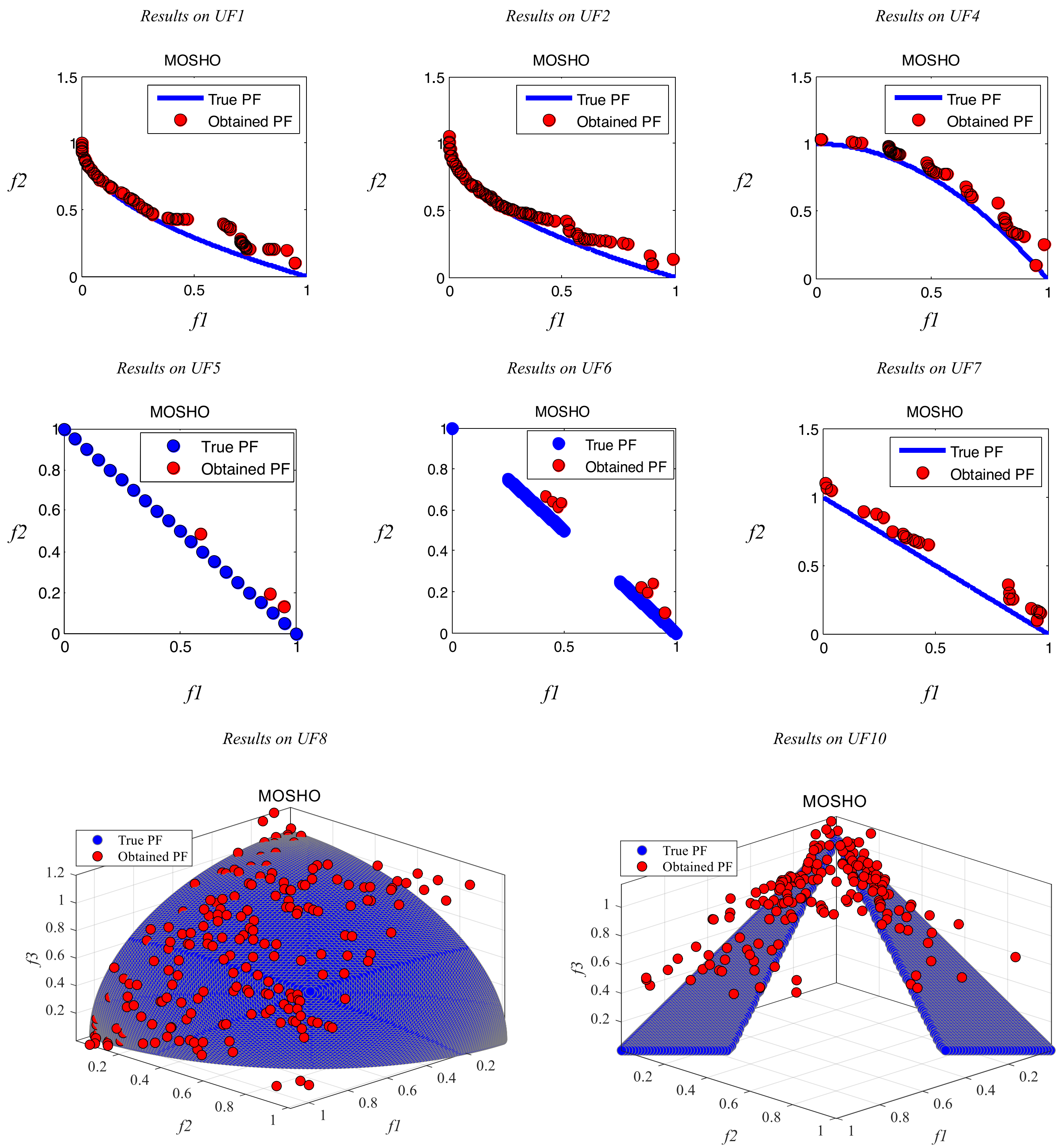


Fig. 10. Best Pareto optimal solutions obtained by the proposed MOSHO algorithm on CEC–2009 benchmark test problems.

4.4.2. Results on ZDT ($ZDT1 - ZDT6$) benchmark test functions

The performance comparison of MOSHO with various algorithms on *ZDT* benchmark test problems are tabulated in Table 3. It has been observed from table that MOSHO outperforms the other algorithms on the majority of *ZDT* test functions in terms of Δ_p, Spread, and Epsilon. The best Pareto optimal solutions obtained from the above-mentioned algorithms is shown in Fig. 11. This figure illustrates that the MOSHO provides better true Pareto optimal front. It is also observed that NSGA-2, MOEA/D, and SPEA-2 perform poorer results on *ZDT*2 test problem. MOSHO is able to provide efficient results on *ZDT*2, *ZDT*3, and *ZDT*6 test problems while all of the other algorithms fail to achieve Pareto optimal solutions. Fig. 12 shows the boxplot of MOSHO is considerably narrower than those of MOPSO, NSGA-2, MOEA/D, PESA-2, SPEA-2, and MOACO.

4.4.3. Results on DTLZ ($DTLZ1 - DTLZ9$) benchmark test functions

The statistical results of MOPSO, NSGA-2, MOEA/D, PESA-2, SPEA-2, and MOACO on DTLZ benchmark problems are shown in

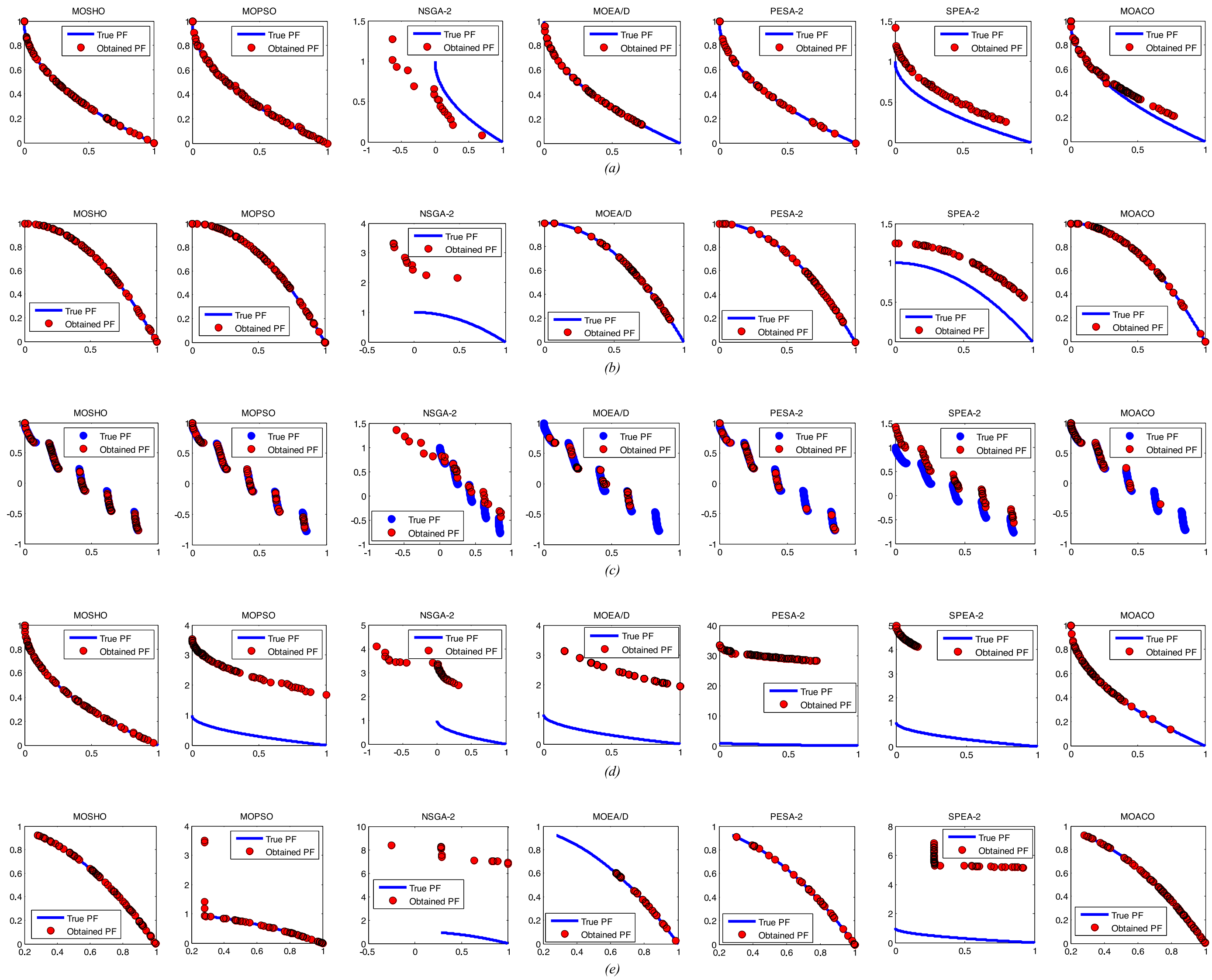


Fig. 11. Best Pareto optimal solutions obtained by the multi-objective algorithms on (a) ZDT1, (b) ZDT2, (c) ZD3, (d) ZDT4, and (e) ZDT6 test problems.

Table 4. For *DTLZ*1, *DTLZ*2, *DTLZ*3, *DTLZ*5, *DTLZ*6, and *DTLZ*9 test functions, MOSHO provides better statistical results than the others for most of the performance measures. For *DTLZ*4 test function, MOEA/D and NSGA-2 give better value of Δ_p and Epsilon over other algorithms. PESA-2 provides better value of Hypervolume and Spread than the others.

The results show that the proposed algorithm has better convergence ability for most of the *DTLZ* benchmark test functions. Fig. 13 shows the shapes of the obtained Pareto optimal fronts of MOSHO which are closer to the true Pareto optimal front. However, these results prove that the durability of the proposed MOSHO algorithm is considerably high on these unconstrained benchmark test problems. Therefore, MOSHO shows superior convergence on all of the above benchmark test problems.

4.5. Sensitivity analysis

4.5.1. Effect of archive on proposed algorithm

The behavior of archive during consecutive generations is illustrated in Table 1. To show the effect of archive on proposed MOSHO, it has been tested on *ZDT*1, *DTLZ*6, and *ZDT*3 benchmark test functions. *ZDT*1, *DTLZ*6, and *ZDT*3 have concave, convex, and disconnected properties respectively. The archive size of these test problems has been fixed at 10. The detailed convergence analysis is depicted in Fig. 7. It has been observed that the proposed MOSHO achieves optimal values of these test functions during the course of iterations.

4.5.2. Effect of selection mechanism on proposed algorithm

The roulette wheel and tournament selection approaches are used to analyze the performance of proposed MOSHO. For experimentation, *ZDT*1, *ZDT*3, and *ZDT*6 test functions are taken into the consideration. Fig. 9 shows the convergence analysis of roulette wheel and tournament selection approaches. From Fig. 9, it can be concluded that the convergence of roulette wheel selection approach is better than the tournament selection approach. It is also observed that the roulette wheel selection approach is able to converge towards the optimal solutions.

5. MOSHO for engineering design problems

In order to investigate the performance of proposed MOSHO, it has been applied on six real-life constrained engineering design problems such as welded beam design, multiple-disk clutch brake design, pressure vessel design, speed reducer design, gear train design, and 25-bar truss design. There are different types of penalty

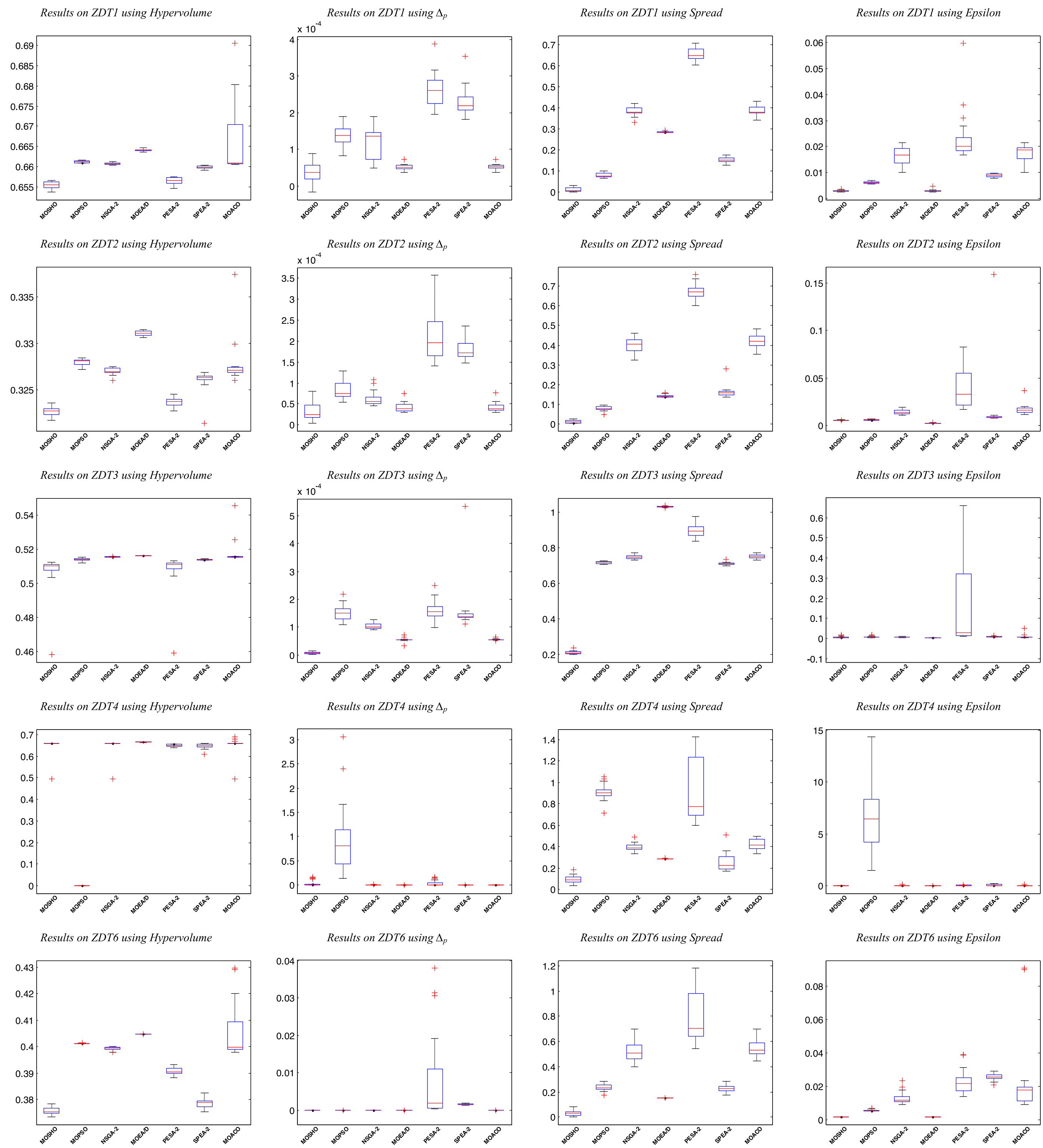


Fig. 12. Boxplot of the statistical results for Hypervolume, Δ_p, Spread, and Epsilon on ZDT benchmark test problems.

functions to handle these multi-constraint problems such as Static penalty, Dynamic penalty, Annealing penalty, Adaptive penalty, Co-evolutionary penalty, and Death penalty [58].

However, death penalty function is used to discard the infeasible solutions and does not employ the information of such solutions which are helpful to solve the dominated infeasible regions. Due to low computational cost and its simplicity, MOSHO algorithm is equipped with death penalty function to handle multiple constraints.

The proposed MOSHO is compared with six well-known algorithms such as MOPSO, NSGA-2, MOEA/D, PESA-2, SPEA-2, and MOACO. The algorithm's parameter settings are same as used in Section 4.2.

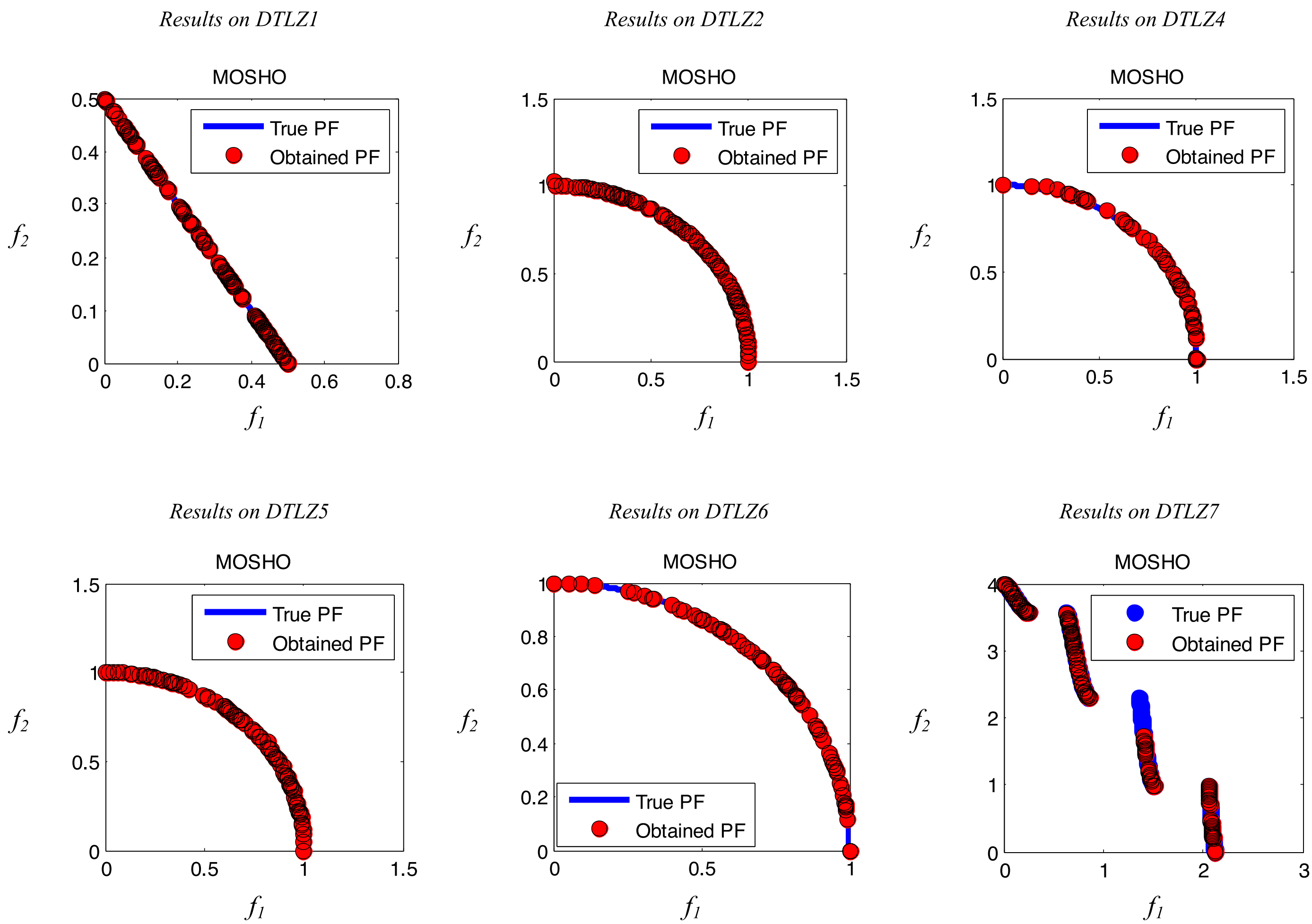


Fig. 13. Best Pareto optimal solutions obtained by the proposed MOSHO algorithm on DTLZ benchmark test problems.

Table 5
Comparison of best solution obtained from different algorithms for welded beam design problem.

Performance	MOSHO		MOPSO		NSGA-2		MOEA/D		PESA-2		SPEA-2		MOACO	
Metrics	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
Hypervolume	**5.18E−01**	**3.35E−01**	8.09E−01	6.93E−01	7.17E−01	5.34E−01	7.94E−01	6.97E−01	9.93E−01	8.96E−01	6.40E−01	4.87E−01	8.59E−01	5.70E−01
Δ_p	**3.71E−02**	**5.07E−03**	2.47E−01	6.55E−02	1.69E−01	2.87E−02	4.45E−02	2.30E−02	6.84E−01	6.87E−02	7.92E−02	5.50E−02	4.34E−02	3.48E−02
Spread	**1.37E−01**	**5.96E−02**	2.14E−01	8.29E−02	9.90E−01	1.78E−01	1.78E+00	1.01E+00	1.81E−01	7.20E−02	6.59E−01	2.18E−01	3.66E−01	2.97E−01
Epsilon	**9.64E−03**	**2.41E−03**	1.04E−01	9.64E−02	9.64E−02	1.36E−02	4.22E−02	9.75E−03	1.15E−01	9.63E−02	1.52E−02	8.76E−03	4.68E−02	3.05E−03

5.1. Welded beam design

The main objective of this problem is to minimize the fabrication cost and simultaneously minimize the vertical deflection at the end of the welded beam as shown in Fig. 14. There are four optimization variables of this problem.

- Thickness of weld (h).
- Length of the clamped bar (l).
- Height of the bar (t).
- Thickness of the bar (b).

The constraints of welded beam problem are:

- Shear stress (τ) and bending stress (θ) in the beam.
- Buckling load (P_c) on the bar.
- End deflection (δ) of the beam.

There are some important points that are considered during the optimization of problem.

- The first constraint confirms that the shear stress of beam is smaller than the shear strength of material.
- The normal stress is smaller than the strength of the material (i.e., 30,000 psi).
- The third constraint confirms that the thickness of beam is not smaller than the thickness of weld.
- The fourth constraint confirms that the buckling load is more than the applied load of beam.

The mathematical formulation of this problem is given in Appendix D. Table 5 shows the comparisons of different algorithms for the best obtained solutions. The results reveal that MOSHO performs better than the other existing algorithms.

By observing Fig. 15, MOSHO achieves the near optimal solution in the initial steps of iterations and durability of the proposed MOSHO algorithm is high on this constrained engineering problem.

Table 6
Comparison of best solution obtained from different algorithms for multiple-disk clutch brake design problem.

Performance	MOSHO		MOPSO		NSGA-2		MOEA/D		PESA-2		SPEA-2		MOACO	
Metrics	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
Hypervolume	**4.53E−01**	**2.49E−01**	7.86E−01	5.32E−01	7.19E−01	5.63E−01	9.64E−01	7.63E−01	9.03E−01	8.14E−01	9.64E−01	7.96E−01	6.40E−01	4.02E−01
Δ_p	**2.73E−02**	**1.48E−03**	5.16E−02	5.99E−03	2.13E−01	2.52E−02	7.43E−02	6.03E−02	2.21E−01	9.91E−02	9.72E−02	3.17E−02	5.13E−02	4.52E−02
Spread	**1.96E−01**	**1.10E−01**	2.36E−01	1.35E−01	8.64E−01	5.24E−01	1.54E+00	1.35E+00	9.52E−01	5.21E−01	8.30E−01	4.40E−01	4.42E−01	2.38E−01
Epsilon	**1.76E−02**	**5.24E−03**	1.14E−01	9.84E−02	7.52E−02	2.44E−02	1.39E−01	8.32E−02	1.10E−01	7.30E−02	2.93E−02	1.31E−02	2.08E−02	1.58E−02

Table 7
Comparison of best solution obtained from different algorithms for pressure vessel design problem.

Performance	MOSHO		MOPSO		NSGA-2		MOEA/D		PESA-2		SPEA-2		MOACO	
Metrics	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
Hypervolume	**3.03E−01**	**2.00E−01**	7.30E−01	4.23E−01	6.25E−01	4.51E−01	8.95E−01	7.13E−01	8.36E−01	7.01E−01	9.60E−01	8.95E−01	4.00E−01	3.79E−01
Δ_p	**2.67E−03**	**1.00E−03**	6.16E−01	2.19E−02	4.31E−02	4.45E−03	2.59E−01	5.98E−02	4.00E−02	3.22E−02	3.67E−01	9.90E−02	2.28E−02	1.63E−02
Spread	**1.77E−01**	**1.20E−01**	2.01E−01	1.56E−01	5.21E−01	4.03E−01	1.44E+00	1.04E+00	8.45E−01	6.38E−01	9.63E−01	6.35E−01	3.07E−01	2.89E−01
Epsilon	**1.19E−02**	**4.21E−03**	1.04E−01	8.43E−02	9.52E−02	7.84E−02	1.54E−01	1.35E−01	1.40E−01	1.04E−01	3.00E−02	1.93E−02	2.58E−02	2.07E−02

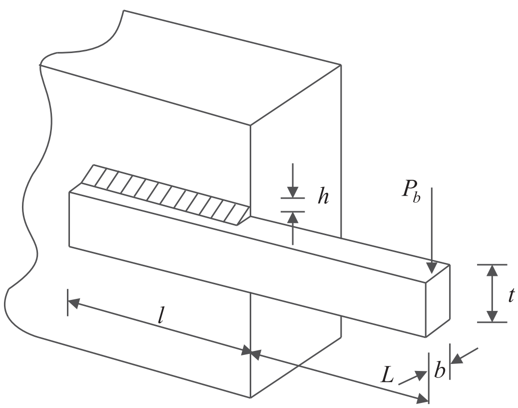


Fig. 14. Schematic view of welded beam problem.

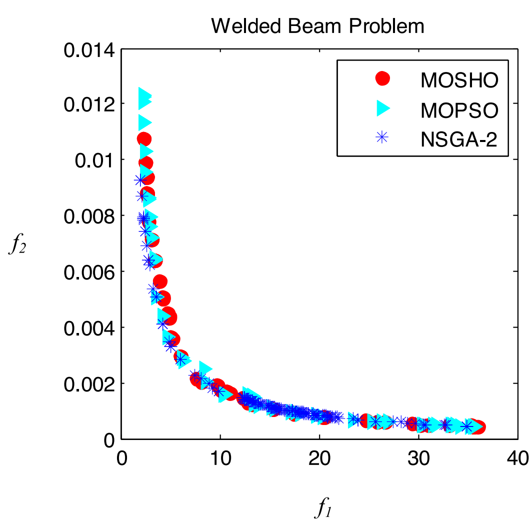


Fig. 15. Best Pareto optimal front obtained by MOSHO, MOPSO, and NSGA-2 on welded beam design problem.

5.2. Multiple-disk clutch brake design problem

The objective of this problem is to minimize the stopping time (f_1) as well as minimize the mass of the brake system (f_2) as shown in Fig. 16. There are five decision variables of this problem.

- Inner radius in mm (r_i).
- Outer radius in mm (r_o).

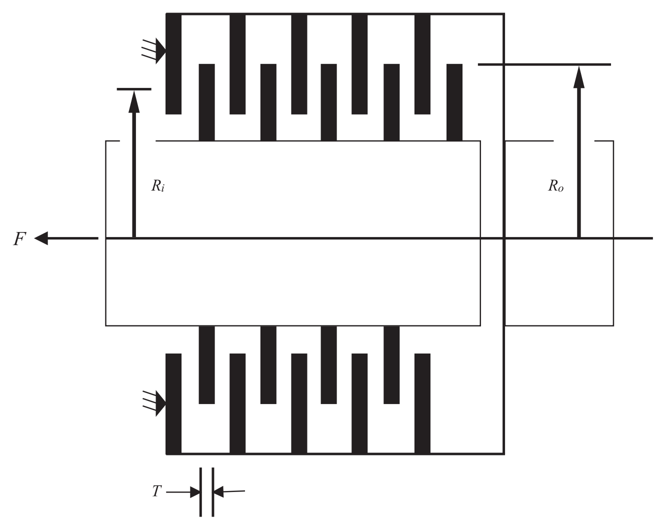


Fig. 16. Schematic view of multiple-disk clutch brake design problem.

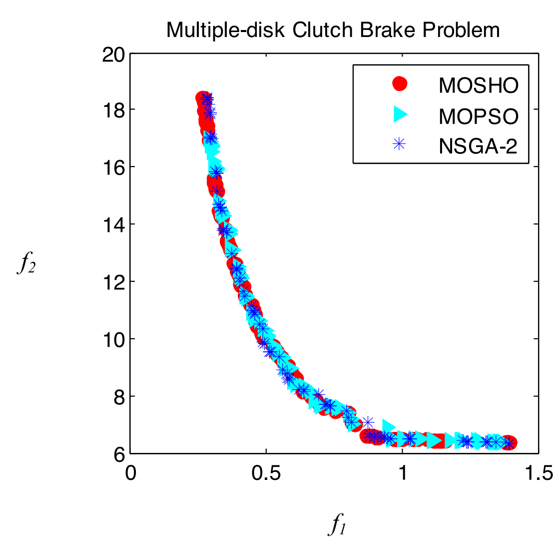


Fig. 17. Best Pareto optimal front obtained by MOSHO, MOPSO, and NSGA-2 on multiple-disk clutch brake design problem.

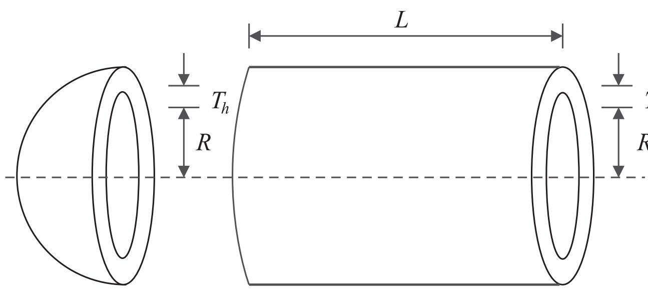


Fig. 18. Schematic view of pressure vessel problem.

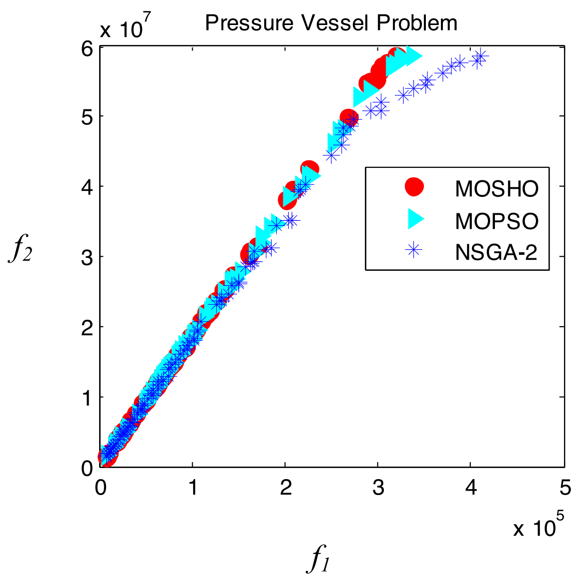


Fig. 19. Best Pareto optimal front obtained by MOSHO, MOPSO, and NSGA-2 on pressure vessel problem.

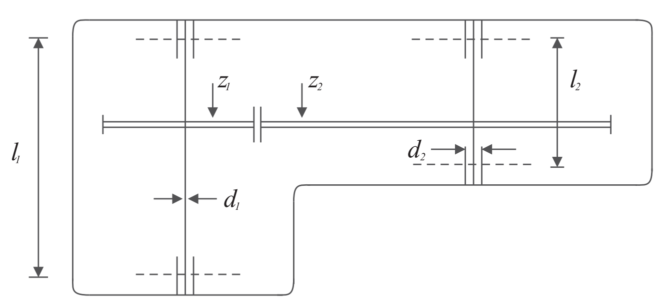


Fig. 20. Schematic view of speed reducer problem.

- Thickness of discs in mm (t).
- Actuating force (F).
- Number of friction surfaces (Z).

The statistical results of best optimal solution among several algorithms are given in Table 6. Among these algorithms, the proposed MOSHO gives optimal value of decision variables. It is worth mentioning that MOSHO provides better statistical results in terms of Hypervolume, Δ_p, Spread, and Epsilon.

Fig. 17 shows the best Pareto optimal fronts obtained from MOSHO, MOPSO, and NSGA-2. From Fig. 17, it can be concluded that the performances of MOSHO, MOPSO, and NSGA-2 are comparable to each other.

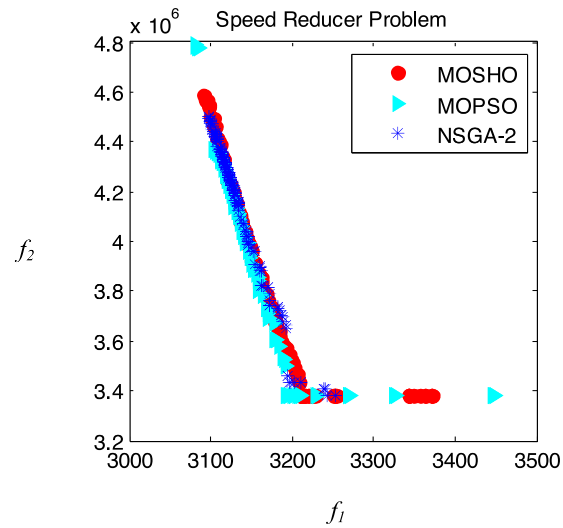


Fig. 21. Best Pareto optimal front obtained by MOSHO, MOPSO, and NSGA-2 on speed reducer problem.

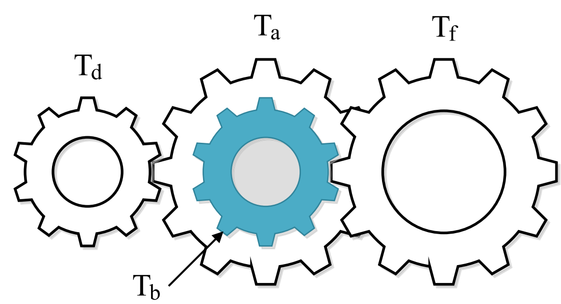


Fig. 22. Schematic view of gear train problem.

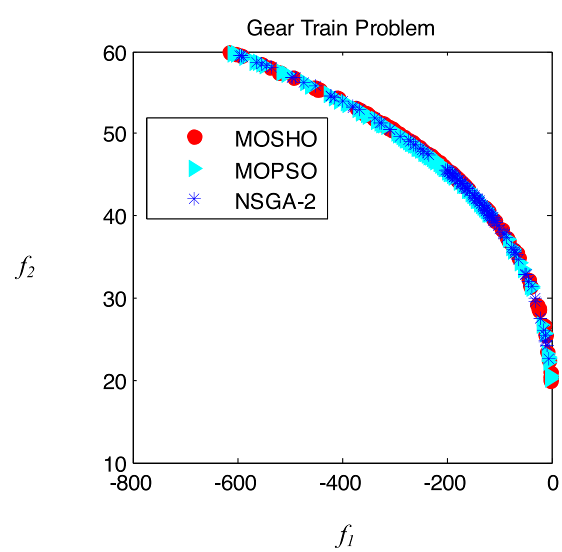


Fig. 23. Best Pareto optimal front obtained by MOSHO, MOPSO, and NSGA-2 on gear train design problem.

5.3. Pressure vessel design

This problem was proposed by Kannan and Kramer [59] to minimize the total cost (f_1) including cost of material, forming and welding of cylindrical vessel which are capped at both ends by hemispherical heads, and to maximize the storage capacity (f_2) as shown in Fig. 18.

Table 8
Comparison of best solution obtained from different algorithms for speed reducer design problem.

Performance	MOSHO		MOPSO		NSGA-2		MOEA/D		PESA-2		SPEA-2		MOACO	
Metrics	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
Hypervolume	**3.22E−01**	**2.20E−01**	6.97E−01	4.35E−01	6.00E−01	5.62E−01	8.63E−01	6.69E−01	9.95E−01	8.03E−01	8.64E−01	6.14E−01	5.11E−01	4.00E−01
Δ_p	**4.13E−03**	**1.57E−03**	5.07E−02	4.84E−03	3.37E−02	3.03E−02	7.68E−02	2.00E−02	5.00E−01	8.93E−02	6.42E−03	4.66E−03	5.99E−02	3.57E−03
Spread	**1.69E−01**	**1.22E−01**	3.36E−01	2.15E−01	7.63E−01	4.32E−01	1.97E+00	1.45E+00	4.52E−01	6.32E−01	6.95E−01	4.44E−01	3.11E−01	2.52E−01
Epsilon	**1.00E−02**	**5.09E−03**	1.09E−01	1.02E−01	1.39E−01	8.62E−02	1.96E−01	7.63E−02	6.53E−02	4.25E−02	1.34E−01	1.96E−01	2.75E−01	4.54E−02

Table 9
Comparison of best solution obtained from different algorithms for gear train design problem.

Performance	MOSHO		MOPSO		NSGA-2		MOEA/D		PESA-2		SPEA-2		MOACO	
Metrics	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
Hypervolume	**3.64E−01**	**2.86E−01**	9.53E−01	6.96E−01	6.11E−01	5.60E−01	6.39E−01	7.97E−01	8.62E−01	7.14E−01	9.86E−01	8.91E−01	6.48E−01	4.34E−01
Δ_p	**2.62E−02**	1.41E−02	5.40E−01	2.89E−02	4.58E−02	**5.65E−03**	6.79E−01	2.33E−02	4.45E−01	4.08E−02	7.00E−02	4.66E−02	4.50E−02	3.32E−02
Spread	**1.26E−01**	**7.63E−02**	2.04E−01	1.03E−01	5.39E−01	3.12E−01	1.25E+00	1.02E+00	7.93E−01	5.00E−01	9.64E−01	5.32E−01	5.20E−01	3.47E−01
Epsilon	**3.45E−03**	**1.51E−02**	1.11E−01	7.62E−02	1.30E−01	1.04E−01	1.20E−01	9.65E−02	9.46E−02	3.49E−02	1.27E−01	1.25E−01	4.62E−02	2.22E−02

Table 10
Member stress limitations for the 25-bar truss design problem.

Element group	Compressive stress limitations Ksi (MPa)	Tensile stress limitations Ksi (MPa)
Group 1	35.092 (241.96)	40.0 (275.80)
Group 2	11.590 (79.913)	40.0 (275.80)
Group 3	17.305 (119.31)	40.0 (275.80)
Group 4	35.092 (241.96)	40.0 (275.80)
Group 5	35.092 (241.96)	40.0 (275.80)
Group 6	6.759 (46.603)	40.0 (275.80)
Group 7	6.959 (47.982)	40.0 (275.80)
Group 8	11.082 (76.410)	40.0 (275.80)

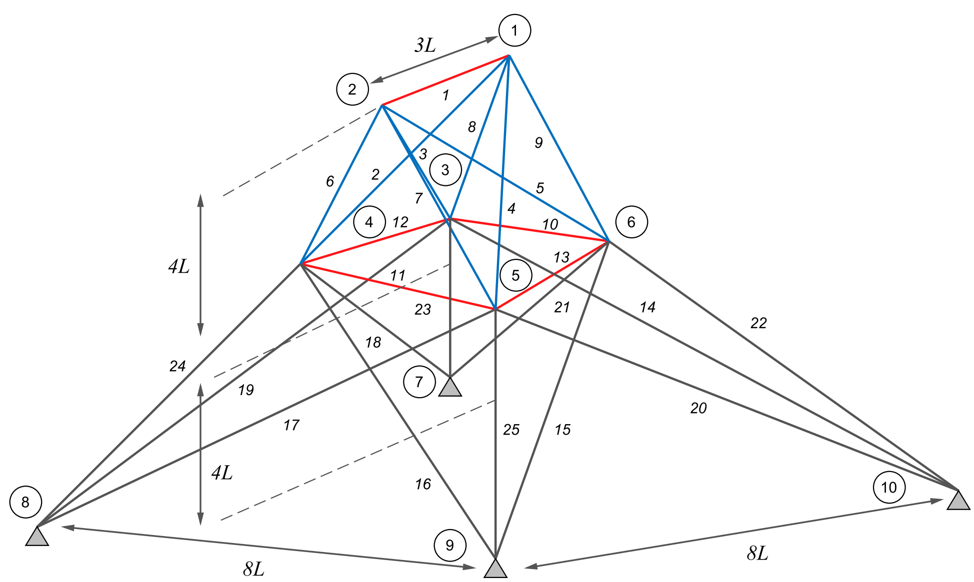


Fig. 24. Schematic view of 25-bar truss design problem.

There are four design variables ($x_1 - x_4$) of this problem.

- T_s (x_1, thickness of the shell).
- T_h (x_2, thickness of the head).
- R (x_3, inner radius).
- L (x_4, length of the cylindrical section without considering the head).

where R and L are continuous variables, while T_s and T_h are integer values which are multiples of 0.0625 *in*. The mathematical formulation of this problem is formulated in Appendix D.

Table 7 shows the comparisons of best optimal solution for MOSHO and other reported algorithms. According to this table, MOSHO is able to find optimal design with minimum cost and maximum storage.

The shape of Pareto optimal fronts obtained from MOSHO, MOPSO, and NSGA-2 are shown in Fig. 19. MOSHO is also able to find a good distribution of points similar to that of NSGA-2 and MOPSO.

5.4. Speed reducer design problem

The design of speed reducer is a most challenging test problem which is associated with seven design variables [60]. In this optimization problem (see Fig. 20), the weight of speed reducer (f_1)

Table 11
Two loading conditions for the 25-bar truss design problem.

Node	Case 1			Case 2		
	$P_x Kips(kN)$	$P_y Kips(kN)$	$P_z Kips(kN)$	$P_x Kips(kN)$	$P_y Kips(kN)$	$P_z Kips(kN)$
1	0.0	20.0 (89)	−5.0 (22.25)	1.0 (4.45)	10.0 (44.5)	−5.0 (22.25)
2	0.0	−20.0 (89)	−5.0 (22.25)	0.0	10.0 (44.5)	−5.0 (22.25)
3	0.0	0.0	0.0	0.5 (2.22)	0.0	0.0
6	0.0	0.0	0.0	0.5 (2.22)	0.0	0.0

Table 12
Comparison of best solution obtained from different algorithms for 25-bar truss design problem.

Performance Metrics	MOSHO		MOPSO		NSGA-2		MOEA/D		PESA-2		SPEA-2		MOACO	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
Hypervolume	**2.14E−01**	**1.16E−01**	8.59E−01	5.79E−01	4.35E−01	2.42E−01	7.48E−01	5.00E−01	4.05E−01	2.71E−01	6.76E−01	5.85E−01	3.05E−01	2.31E−01
Δ_p	**3.22E−02**	**1.42E−02**	2.56E−01	1.72E−01	3.79E−02	2.47E−02	2.00E−01	1.34E−01	2.68E−01	4.00E−02	3.27E−02	2.86E−02	5.38E−02	3.59E−02
Spread	**2.17E−01**	**4.33E−02**	3.59E−01	3.43E−01	4.30E−01	3.27E−01	2.49E−01	1.19E−01	5.07E−01	4.04E−01	5.38E−01	2.86E−01	3.34E−01	2.02E−01
Epsilon	**2.88E−02**	**1.72E−02**	1.91E−01	3.00E−02	2.80E−01	1.44E−01	1.59E−01	2.51E−02	3.87E−02	2.76E−02	2.42E−01	2.00E−01	3.66E−01	2.58E−01

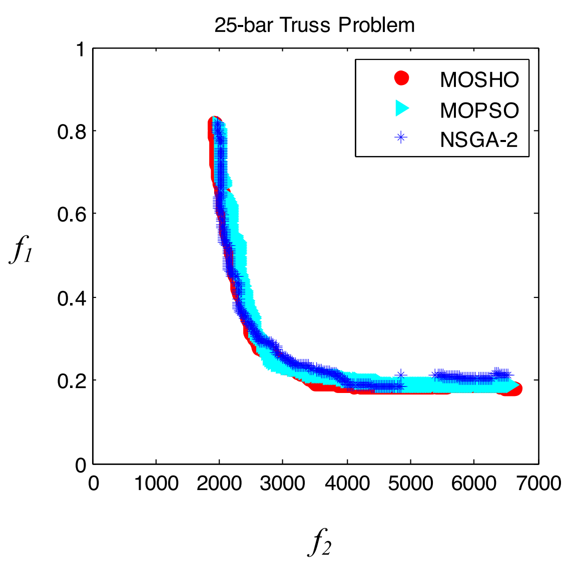


Fig. 25. Best Pareto optimal front obtained by MOSHO, MOPSO, and NSGA-2 on 25-bar truss design problem.

and stress of a speed reducer (f_2) should be minimized with eleven constraints. There are seven optimization variables ($x_1 - x_7$) of this problem.

- Face width (b).
- Module of teeth (m).
- Number of teeth in the pinion (z).
- Length of the first shaft between bearings (l_1).
- Length of the second shaft between bearings (l_2).
- Diameter of first (d_1) shafts.
- The diameter of second shafts (d_2).

The mathematical formulation of this problem is formulated in Appendix D.

Table 8 shows the comparison of best solutions for six optimization algorithms and proposed MOPSO in terms of function values.

From Table 8, it is evident that MOSHO produces best solution with considerable improvement as compared to other algorithms. MOSHO algorithm is able to find a design with minimum weight and stress, simultaneously.

The best Pareto optimal solutions obtained from MOSHO, MOPSO, and NSGA-2 algorithms are shown in Fig. 21. MOSHO obtains best Pareto optimal solution for speed reducer problem during course of iterations. The MOSHO algorithm achieves better results and verified by finding a set of Pareto solutions.

5.5. Gear train design problem

The objective of this problem is to find the number of teeth for each four gears so as to minimize the error between obtained and required gear ratio and minimize the size of four gears as shown in Fig. 22. There are four integer decision variables of this problem.

- Number of teeth on the driving gear (T_d).
- Number of teeth on the driven gear (T_a).
- Number of teeth on the gear attached to the driven gear (T_b).
- Number of teeth on the following gear (T_f).

The mathematical formulation of this problem is formulated in Appendix D.

Table 9 shows the comparison for best solution given by MOSHO and other algorithms. The proposed algorithm provides best optimal solution over the other optimizers.

The best Pareto optimal fronts obtained from MOSHO, MOPSO, and NSGA-2 are shown in Fig. 23. MOSHO is able to find a set of well-distributed points on the entire Pareto optimal front.

5.6. 25-Bar truss design

The truss design problem is very popular optimization problem in the literature [61,62]. As shown in Fig. 24, there are ten nodes which are fixed and twenty five bars cross-sectional members which are grouped into eight categories:

- Group 1: A_1
- Group 2: A_2, A_3, A_4, A_5
- Group 3: A_6, A_7, A_8, A_9
- Group 4: A_{10}, A_{11}
- Group 5: A_{12}, A_{13}
- Group 6: A_{14}, A_{15}, A_{17}
- Group 7: A_{18}, A_{19}, A_{20}, A_{21}
- Group 8: A_{22}, A_{23}, A_{24}, A_{25}

The other variables which effects on this problem are as:

- $p = 0.0272$ N/cm^3 (0.1 lb/in.^{3})
- $E = 68947$ MPa (10000 Ksi)
- Displacement limitation = 0.35 in.
- Maximum displacement = 0.3504 in.
- Design variable set = {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.6, 2.8, 3.0, 3.2, 3.4 }

The member stress limitations for this truss is shown in Table 10.

The leading conditions of 25-bar truss is tabulated in Table 11. Table 12 shows the best optimal weight obtained by MOSHO is better than other competitor algorithms. The statistical results (i.e., average and standard deviation) also show that MOSHO outperforms the other algorithms and converge very efficiently to optimize this problem as shown in Fig. 25.

In summary, the experimental results reveal that MOSHO is the obvious choice for high performance to solve challenging problems. The optimization results point out that MOSHO has the capability to handle various multi-objective optimization problems.

6. Conclusions

This paper presented a multi-objective swarm-based optimization technique called Multi-objective Spotted Hyena Optimizer (MOSHO). The two components are introduced to solve multi-objective problem. The first component is an archive for retrieving and storing the best non-dominated solutions which are obtained so far and the second component is a group selection mechanism of nearby optimal solutions from the archive. MOSHO has been tested on twenty four well-known benchmark test functions. The results reveal that the performance of MOSHO algorithm is better than the MOPSO, NSGA-2, MOEA/D, PESA-2, SPEA-2, and MOACO. The statistical results demonstrate that the proposed method is able to handle many constraints and offers better solutions than other optimizers. The proposed method is applied on six real-life multi-objective optimization problems. The results reveal that MOSHO provides better results than the others in terms of computational cost.

For future work, we are planning to extend this algorithm to solve many-objective problems in the field of bio-informatics and further its applicability on data clustering techniques.

Appendix A. Unconstrained multi-objective test problems

Name	Mathematical formulation	Properties
UF1	$f_1 = x_1 + \frac{2}{\mid J_1 \mid} \sum_{j \epsilon J_1} [x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right)]^2$ $f_2 = 1 - \sqrt{x} + \frac{2}{\mid J_2 \mid} \sum_{j \epsilon J_2} [x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right)]^2$ $J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}, J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$	Bi-objective
UF2	$f_1 = x_1 + \frac{2}{\mid J_1 \mid} \sum_{j \epsilon J_1} y_j^2$ $f_2 = 1 - \sqrt{x} + \frac{2}{\mid J_2 \mid} \sum_{j \epsilon J_2} y_j^2$ $J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}, J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$ $y_j = \begin{cases} x_j - [0.3x_1^2 cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1]cos\left(6\pi x_1 + \frac{j\pi}{n}\right), & if \ \ j\epsilon J_1 \\ x_j - [0.3x_1^2 cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1]sin\left(6\pi x_1 + \frac{j\pi}{n}\right), & if \ \ j\epsilon J_2 \end{cases}$	Bi-objective
UF3	$f_1 = x_1 + \frac{2}{\mid J_1 \mid}\left(4\sum_{j\epsilon J_1} y_j^2 - 2\prod_{j\epsilon J_1} cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right)$ $f_2 = \sqrt{x_1} + \frac{2}{\mid J_2 \mid}\left(4\sum_{j\epsilon J_1} y_j^2 - 2\prod_{j\epsilon J_2} cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right)$ $J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}, J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$ $y_j = x_j - x_1^{0.5(1.0 + \frac{3(j-2)}{n-2})}, j = 2, 3, \ldots, n$	Bi-objective
UF4	$f_1 = x_1 + \frac{2}{\mid J_1 \mid} \sum_{j\epsilon J_1} h(y_j)$ $f_2 = 1 - x_2 + \frac{2}{\mid J_2 \mid} \sum_{j\epsilon J_2} h(y_j)$ $J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}, J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$ $y_j = x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, 3, \ldots, n, \quad h(t) = \frac{\mid t \mid}{1 + e^{2\mid t \mid}}$	Bi-objective
UF5	$f_1 = x_1 + \left(\frac{1}{2N} + \epsilon\right) \mid sin(2N\pi x_1) \mid + \frac{2}{\mid J_1 \mid} \sum_{j\epsilon J_1} h(y_i)$ $f_2 = 1 - x_1 + \left(\frac{1}{2N} + \epsilon\right) \mid sin(2N\pi x_1) \mid + \frac{2}{\mid J_2 \mid} \sum_{j\epsilon J_2} h(y_i)$ $J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}, J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$ $\epsilon > 0, \quad y_j = x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, 3, \ldots, n, \quad h(t) = 2t^2 - cos(4\pi t) + 1$	Bi-objective
UF6	$f_1 = x_1 + max\{0, 2\left(\frac{1}{2N} + \epsilon\right) sin(2N\pi x_1)\} + \frac{2}{\mid J_1 \mid}\left(4\sum_{j\epsilon J_1} y_j^2 - 2\prod_{j\epsilon J_1} cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 1\right)$ $f_2 = 1 - x_1 + max\{0, 2\left(\frac{1}{2N} + \epsilon\right) sin(2N\pi x_1)\} + \frac{2}{\mid J_2 \mid}\left(4\sum_{j\epsilon J_2} y_j^2 - 2\prod_{j\epsilon J_2} cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 1\right)$ $J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}, J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$ $\epsilon > 0, \quad y_j = x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, 3, \ldots, n$	Bi-objective
UF7	$f_1 = \sqrt[5]{x_1} + \frac{2}{\mid J_1 \mid} \sum_{j\epsilon J_1} y_j^2$ $f_2 = 1 - \sqrt[5]{x_1} + \frac{2}{\mid J_2 \mid} \sum_{j\epsilon J_2} y_j^2$	Bi-objective

(continued on next page)

(*continued*)

Name	Mathematical formulation	Properties
	$J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}, J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$ $\epsilon > 0,\quad y_j = x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, 3, \dots, n$	
UF8	$f_1 = cos(0.5x_1\pi)cos(0.5x_2\pi) + \frac{2}{\mid J_1 \mid}\sum_{j\epsilon J_1}\left(x_j - 2x_2 sin\left(2\pi x_1 + \frac{j\pi}{n}\right)^2\right)$ $f_2 = cos(0.5x_1\pi)sin(0.5x_2\pi) + \frac{2}{\mid J_2 \mid}\sum_{j\epsilon J_2}\left(x_j - 2x_2 sin\left(2\pi x_1 + \frac{j\pi}{n}\right)^2\right)$ $f_3 = sin(0.5x_1\pi) + \frac{2}{\mid J_3 \mid}\sum_{j\epsilon J_3}\left(x_j - 2x_2 sin\left(2\pi x_1 + \frac{j\pi}{n}\right)^2\right)$ $J_1 = \{j \mid 3 \le j \le n, \text{ and } j-1 \text{ is a multiplication of } 3\}$ $J_2 = \{j \mid 3 \le j \le n, \text{ and } j-2 \text{ is a multiplication of } 3\}$ $J_3 = \{j \mid 3 \le j \le n, \text{ and } j \text{ is a multiplication of } 3\}$	Tri-objective
UF9	$f_1 = 0.5[max\{0, (1+\epsilon)(1-4(2x_1-1)^2)\} + 2x_1]x_2 + \frac{2}{\mid J_1 \mid}\sum_{j\epsilon J_1}\left(x_j - 2x_2 sin\left(2\pi x_1 + \frac{j\pi}{n}\right)^2\right)$ $f_2 = 0.5[max\{0, (1+\epsilon)(1-4(2x_1-1)^2)\} + 2x_1]x_2 + \frac{2}{\mid J_2 \mid}\sum_{j\epsilon J_2}\left(x_j - 2x_2 sin\left(2\pi x_1 + \frac{j\pi}{n}\right)^2\right)$ $f_3 = 1 - x_2 + \frac{2}{\mid J_3 \mid}\sum_{j\epsilon J_3}\left(x_j - 2x_2 sin\left(2\pi x_1 + \frac{j\pi}{n}\right)^2\right)$ $J_1 = \{j \mid 3 \le j \le n, \text{ and } j-1 \text{ is a multiplication of } 3\}$ $J_2 = \{j \mid 3 \le j \le n, \text{ and } j-2 \text{ is a multiplication of } 3\}$ $J_3 = \{j \mid 3 \le j \le n, \text{ and } j \text{ is a multiplication of } 3\},\quad \epsilon = 0.1$	Tri-objective
UF10	$f_1 = cos(0.5x_1\pi)cos(0.5x_2\pi) + \frac{2}{\mid J_1 \mid}\sum_{j\epsilon J_1}[4y_j^2 - cos(8\pi y_j) + 1]$ $f_2 = cos(0.5x_1\pi)sin(0.5x_2\pi) + \frac{2}{\mid J_2 \mid}\sum_{j\epsilon J_2}[4y_j^2 - cos(8\pi y_j) + 1]$ $f_3 = sin(0.5x_1\pi) + \frac{2}{\mid J_3 \mid}\sum_{j\epsilon J_3}[4y_j^2 - cos(8\pi y_j) + 1]$ $J_1 = \{j \mid 3 \le j \le n, \text{ and } j-1 \text{ is a multiplication of } 3\}$ $J_2 = \{j \mid 3 \le j \le n, \text{ and } j-2 \text{ is a multiplication of } 3\}$ $J_3 = \{j \mid 3 \le j \le n, \text{ and } j \text{ is a multiplication of } 3\}$	Tri-objective

Appendix B. Unconstrained multi-objective test problems

ZDT1:
Minimize: $f_1(x) = x_1$
Minimize: $f_2(x) = g(x) \times h(f_1(x), g(x))$
where,

$$g(x) = 1 + \frac{9}{N-1}\sum_{i=2}^{N} x_i$$
$$h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}}$$
$$0 \le x_i \le 1,\ 1 \le i \le 30$$

ZDT2:
Minimize: $f_1(x) = x_1$
Minimize: $f_2(x) = g(x) \times h(f_1(x), g(x))$
where,

$$g(x) = 1 + \frac{9}{N-1}\sum_{i=2}^{N} x_i$$
$$h(f_1(x), g(x)) = 1 - \left(\frac{f_1(x)}{g(x)}\right)^2$$
$$0 \le x_i \le 1,\ 1 \le i \le 30$$

ZDT3:
Minimize: $f_1(x) = x_1$
Minimize: $f_2(x) = g(x) \times h(f_1(x), g(x))$
where,

$$g(x) = 1 + \frac{9}{29}\sum_{i=2}^{N} x_i$$
$$h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}} - \left(\frac{f_1(x)}{g(x)}\right)sin(10\pi f_1(x))$$
$$0 \le x_i \le 1,\ 1 \le i \le 30$$

ZDT4:
Minimize: $f_1(x) = x_1$
Minimize: $f_2(x) = g(x) \times [1 - (x_1/g(x))^2]$
where,

$$g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}(x_i^2 - 10cos(4\pi x_i))$$
$$0 \le x_1 \le 1,\ -5 \le x_i \le 5,\ i = 1, 2, \dots, n$$

ZDT6:
Minimize: $f_1(x) = 1 - e^{-4x_1} \times \sin^6(6\pi x_1)$
Minimize: $f_2(x) = 1 - \left(\frac{f_1(x)}{g(x)}\right)^2$
where,

$$g(x) = 1 + 9\left[\frac{\left(\sum_{i=2}^{n} x_i\right)}{(n-1)}\right]^{0.25}$$
$$0 \le x_i \le 1,\ i = 1, 2, \dots, n$$

Appendix C. Unconstrained multi-objective test problems

DTLZ1:
Minimize: $f_1(\vec{x}) = \frac{1}{2}x_1(1 + g(\vec{x}))$
Minimize: $f_2(\vec{x}) = \frac{1}{2}(1 - x_1)(1 + g(\vec{x}))$
where,

$$g(\vec{x}) = 100\left[\mid \vec{x} \mid + \sum_{x_i \epsilon \vec{x}}(x_1 - 0.5)^2 - cos(20\pi(x_i - 0.5))\right]$$
$$0 \le x_i \le 1,\ i = 1, 2, \dots, n$$

DTLZ2:
Minimize: $f_1(\vec{x}) = (1 + g(\vec{x}))cos\left(x_1\frac{\pi}{2}\right)$
Minimize: $f_2(\vec{x}) = (1 + g(\vec{x}))sin\left(x_1\frac{\pi}{2}\right)$
where,

$$g(\vec{x}) = \sum_{x_i \epsilon \vec{x}}(x_i - 0.5)^2$$
$$0 \le x_i \le 1,\ i = 1, 2, \dots, n$$

DTLZ3:
Minimize: $f_1(\vec{x}) = (1 + g(\vec{x}))cos\left(x_1\frac{\pi}{2}\right)$
Minimize: $f_2(\vec{x}) = (1 + g(\vec{x}))sin\left(x_1\frac{\pi}{2}\right)$
where,

$$g(\vec{x}) = 100\Big[\,|\,\vec{x}\,| + \textstyle\sum_{x_i \epsilon \vec{x}} (x_i - 0.5)^2 - cos(20\pi\,(x_i - 0.5))\Big]$$
$$0 \le x_i \le 1,\ \ i = 1, 2, \ldots, n$$

DTLZ4:

Minimize: $f_1(\vec{x}) = (1 + g(\vec{x}))cos\left(x_1^{\alpha}\dfrac{\pi}{2}\right)$

Minimize: $f_2(\vec{x}) = (1 + g(\vec{x}))sin\left(x_1^{\alpha}\dfrac{\pi}{2}\right)$

where,

$$g(\vec{x}) = \textstyle\sum_{x_i \epsilon \vec{x}} (x_i - 0.5)^2$$
$$\alpha = 100$$
$$0 \le x_i \le 1,\ \ i = 1, 2, \ldots, n$$

DTLZ5:

Minimize: $f_1(\vec{x}) = (1 + g(\vec{x}))cos\left(\dfrac{1 + 2g(\vec{x})x_1}{4(1 + g(\vec{x}))} \times \dfrac{\pi}{2}\right)$

Minimize: $f_2(\vec{x}) = (1 + g(\vec{x}))sin\left(\dfrac{1 + 2g(\vec{x})x_1}{4(1 + g(\vec{x}))} \times \dfrac{\pi}{2}\right)$

where,

$$g(\vec{x}) = \textstyle\sum_{x_i \epsilon \vec{x}} (x_i - 0.5)^2$$
$$0 \le x_i \le 1,\ \ i = 2, 3, \ldots, n$$

DTLZ6:

Minimize: $f_1(\vec{x}) = (1 + g(\vec{x}))cos\left(\dfrac{1 + 2g(\vec{x})x_1}{4(1 + g(\vec{x}))} \times \dfrac{\pi}{2}\right)$

Minimize: $f_2(\vec{x}) = (1 + g(\vec{x}))sin\left(\dfrac{1 + 2g(\vec{x})x_1}{4(1 + g(\vec{x}))} \times \dfrac{\pi}{2}\right)$

where,

$$g(\vec{x}) = \textstyle\sum_{x_i \epsilon \vec{x}} x_i^{0.1}$$
$$0 \le x_i \le 1,\ \ i = 2, 3, \ldots, n$$

DTLZ7:

Minimize: $f_1(\vec{x}) = x_1$

Minimize: $f_2(\vec{x}) = (1 + g(\vec{x}))h(f_1(\vec{x}), g(\vec{x}))$

where,

$$g(\vec{x}) = 1 + \frac{9}{|\,\vec{x}\,|}\textstyle\sum_{x_i \epsilon \vec{x}} x_i$$
$$h(f_1(\vec{x}), g(\vec{x})) = M - \frac{f_1(\vec{x})}{1 + g(\vec{x})}(1 + sin(3\pi\, f_1(\vec{x})))$$
$$0 \le x_i \le 1,\ \ 1 \le i \le n$$

Appendix D. Constrained engineering design problems

D1. Welded beam design problem

Minimize $f_1(\vec{z}) = C = 1.10471h^2l + 0.04811tb(14.0 + l)$,

Minimize $f_2(\vec{z}) = D = \dfrac{2.1952}{t^3b}$,

Subject to

$g_1(\vec{z}) = 13,600 - \tau(\vec{z}) \ge 0$,

$g_2(\vec{z}) = 30,000 - \sigma(\vec{z}) \ge 0$,

$g_3(\vec{z}) = b - h \ge 0$,

$g_4(\vec{z}) = P_c(\vec{z}) - 6000 \ge 0$,

Variable range

$0.125 \le h, b \le 5.0$in.,

$0.1 \le l, t \le 10.0$in.,

where

$$\tau(\vec{z}) = \sqrt{(\tau')^2 + (\tau'')^2 + (l\tau'\tau'')/\sqrt{0.25(l^2 + (h + t)^2)}},$$
$$\tau' = \frac{6,000}{\sqrt{2}hl},\ \sigma(\vec{z}) = \frac{504,000}{t^2b},$$
$$\tau'' = \frac{6,000(14 + 0.5l)\sqrt{0.25(l^2 + (h + t)^2)}}{2[0.707hl(l^2/12 + 0.25(h + t)^2)]},$$
$$P_c(\vec{z}) = 64,746.022(1 - 0.0282346t)tb^3.$$

D2. Multiple-disk clutch brake design problem

Minimize $f_1(\vec{z}) = M = \pi(r_o^2 - r_i^2)t(Z + 1)p_m$,

Minimize $f_2(\vec{z}) = T = \dfrac{I_z w}{M_h + M_f}$,

Subject to

$g_1(\vec{z}) = r_o - r_i - \Delta R \ge 0$,

$g_2(\vec{z}) = L_{max} - (Z + 1)(t + \delta) \ge 0$,

$g_3(\vec{z}) = p_{max} - p_{rz} \ge 0$,

$g_4(\vec{z}) = p_{max}V_{sr,max} - p_{rz}V_{sr} \ge 0$,

$g_5(\vec{z}) = V_{sr,max} - V_{sr} \ge 0$,

$g_6(\vec{z}) = M_h - sM_s \ge 0$,

$g_7(\vec{z}) = T \ge 0$,

$g_8(\vec{z}) = T_{max} - T \ge 0$,

$60 \le r_i \le 80$ mm,

$90 \le r_o \le 110$ mm,

$1.5 \le t \le 3$ mm,

$0 \le F \le 1000$ N,

$2 \le Z \le 9$

where

$p_m = 0.0000078$ kg/mm^3, $p_{max} = 1$ MPa,

$\mu = 0.5, V_{sr,max} = 10$ m/s,

$s = 1.5, T_{max} = 15$ s, $n = 250$ rpm, $M_s = 40$ Nm, $M_f = 3$ Nm,

$I_z = 55$ kg-m^2, $\delta = 0.5$ mm, $\Delta R = 20$ mm, $L_{max} = 30$ mm,

$M_h = \dfrac{2}{3}\mu FZ\dfrac{r_o^3 - r_i^3}{r_o^2 - r_i^2}$ N-mm, $w = \dfrac{\pi n}{30}$ rad/s, $R_{sr} = \dfrac{2}{3}\dfrac{r_o^3 - r_i^3}{r_o^2 - r_i^2}$ mm

$A = \pi(r_o^2 - r_i^2)$ mm^2, $p_{rz} = \dfrac{F}{A}$ N/mm^2, $V_{sr} = \dfrac{\pi R_{sr} n}{30}$ mm/s,

D3. Pressure vessel design problem

Minimize $f_1(\vec{z}) = 0.6224T_sLR + 1.7781T_hR^2 + 3.1661T_s^2L + 19.84T_s^2R$,

Minimize $f_2(\vec{z}) = -(\pi R^2L + 1.333\pi R^3)$,

Subject to

$g_1(\vec{z}) = 0.0193R - T_s \le 0$,

$g_2(\vec{z}) = 0.00954R - T_h \le 0$,

$g_3(\vec{z}) = 0.0625 - T_s \le 0$,

$g_4(\vec{z}) = T_s - 5 \le 0$,

$g_5(\vec{z}) = 0.0625 - T_h \le 0$,

$g_6(\vec{z}) = T_h - 5 \le 0$,

$g_7(\vec{z}) = 10 - R \le 0$,

$g_8(\vec{z}) = R - 200 \le 0$,

$g_9(\vec{z}) = 10 - L \le 0$,

$g_{10}(\vec{z}) = L - 240 \le 0$,

D4. Speed reducer design problem

Minimize $f_1(\vec{z}) = 0.7854z_1z_2^2(3.3333z_3^2 + 14.9334z_3 - 43.0934) - 1.508z_1(z_6^2 + z_7^2) + 7.4777(z_6^3 + z_7^3) + 0.7854(z_4z_6^2 + z_5z_7^2)$,

Minimize $f_2(\vec{z}) = \dfrac{\sqrt{\dfrac{745z_4}{z_2z_3}^2 + 1.69 \times 10^7}}{0.1z_6^3}$,

Subject to

$$g_1(\vec{z}) = \frac{27}{z_1 z_2^2 z_3} - 1 \le 0,$$
$$g_2(\vec{z}) = \frac{397.5}{z_1 z_2^2 z_3^2} - 1 \le 0,$$
$$g_3(\vec{z}) = \frac{1.93 z_4^3}{z_2 z_6^4 z_3} - 1 \le 0,$$
$$g_4(\vec{z}) = \frac{1.93 z_5^3}{z_2 z_7^4 z_3} - 1 \le 0,$$
$$g_5(\vec{z}) = \frac{[(745(z_4/z_2 z_3))^2 + 16.9 \times 10^6]^{1/2}}{110 z_6^3} - 1 \le 0,$$
$$g_6(\vec{z}) = \frac{[(745(z_5/z_2 z_3))^2 + 157.5 \times 10^6]^{1/2}}{85 z_7^3} - 1 \le 0,$$
$$g_7(\vec{z}) = \frac{z_2 z_3}{40} - 1 \le 0,$$
$$g_8(\vec{z}) = \frac{5 z_2}{z_1} - 1 \le 0,$$
$$g_9(\vec{z}) = \frac{z_1}{12 z_2} - 1 \le 0,$$
$$g_{10}(\vec{z}) = \frac{1.5 z_6 + 1.9}{z_4} - 1 \le 0,$$
$$g_{11}(\vec{z}) = \frac{1.1 z_7 + 1.9}{z_5} - 1 \le 0,$$

where

$$2.6 \le z_1 \le 3.6, \quad 0.7 \le z_2 \le 0.8, \quad 17 \le z_3 \le 28, \quad 7.3 \le z_4 \le 8.3,$$
$$7.3 \le z_5 \le 8.3, \quad 2.9 \le z_6 \le 3.9, \quad 5.0 \le z_7 \le 5.5$$

D5. Gear train design problem

$$\begin{aligned} &\text{Minimize} \quad f_1(T) = \left| 6.931 - \frac{T_a}{T_d}\frac{T_f}{T_b} \right| \\ &\text{Minimize} \quad f_2(T) = max\left(T_d, T_b, T_a, T_f\right) \\ &\text{Subject to} \\ &\frac{f_1(T)}{6.931} \le 0.5, \\ &12 \le T_d, T_b, T_a, T_f \le 60, \\ &T_d, T_b, T_a, T_f \text{ are integers} \end{aligned}$$

References

[1] G. Dhiman, V. Kumar, Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications, Adv. Eng. Software 114 (2017) 48–70, doi:10.1016/j.advengsoft.2017.05.014.
[2] J. Branke, K. Deb, H. Dierolf, M. Osswald, Finding Knees in Multi-objective Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 722–731. doi:10.1007/978-3-540-30217-9_73.
[3] R. Marler, J. Arora, Survey of multi-objective optimization methods for engineering, Struct. Multidiscip. Optim. 26 (6) (2004) 369–395, doi:10.1007/s00158-003-0368-6.
[4] K. Deb, Advances in Evolutionary Multi-objective Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–26. doi:10.1007/978-3-642-33119-0_1.
[5] X.-S. Yang, M. Karamanoglu, X. He, Flower pollination algorithm: a novel approach for multiobjective optimization, Eng. Optim. 46 (9) (2014) 1222–1237.
[6] X.-S. Yang, A.H. Gandomi, Bat algorithm: a novel approach for global engineering optimization, Eng. Comput. (Swansea) 29 (5) (2012) 464–483.
[7] X.-S. Yang, Engineering optimization: An introduction with metaheuristic applications, John Wiley & Sons, 2010.
[8] N.K. Madavan, Multiobjective optimization using a pareto differential evolution approach, in: Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on, 2, 2002, pp. 1145–1150, doi:10.1109/CEC.2002.1004404.
[9] C.A.C. Coello, Evolutionary multi-objective optimization: a historical view of the field, IEEE Comput. Intell. Mag. 1 (1) (2006) 28–36, doi:10.1109/MCI.2006.1597059.
[10] G.-C. Luh, C.-H. Chueh, Multi-objective optimal design of truss structure with immune algorithm, Comput. Struct. 82 (11–12) (2004) 829–844. https://doi.org/10.1016/j.compstruc.2004.03.003.
[11] T. Kipouros, D.M. Jaeggi, W.N. Dawes, G.T. Parks, A.M. Savill, P.J. Clarkson, Biobjective design optimization for axial compressors using tabu search, AIAA J. 46 (3) (2008) 701–711, doi:10.2514/1.32794.
[12] G. Dhiman, V. Kumar, Spotted hyena optimizer for solving complex and non–linear constrained engineering problems, Advances in Intelligent Systems and Computing, Springer, 2018, In press.
[13] M. Chen, O. Hammami, A System Engineering Conception of Multi-objective Optimization for Multi-physics System, Springer International Publishing, Cham, pp. 299–306. doi:10.1007/978-3-319-14532-7_31.
[14] J. Handl, D.B. Kell, J. Knowles, Multiobjective optimization in bioinformatics and computational biology, IEEE/ACM Trans. Comput. Biol. Bioinformatics 4 (2) (2007) 279–292, doi:10.1109/TCBB.2007.070203.
[15] A. Kaur, G. Dhiman, A review on search based tools and techniques to identify bad code smells in object oriented systems, Advances in Intelligent Systems and Computing, Springer, 2018, In press.
[16] R. Maini, G. Dhiman, Impacts of artificial intelligence on real-life problems, Internation Journal Of Advance Research And Innovative Ideas In Education 4 (1) (2018) 291–295.
[17] C.A.C. Coello, G.B. Lamont, D.A.V. Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
[18] R.K. Chandrawat, R. Kumar, B.P. Garg, G. Dhiman, S. Kumar, An Analysis of Modeling and Optimization Production Cost Through Fuzzy Linear Programming Problem with Symmetric and Right Angle Triangular Fuzzy Number, Springer Singapore, Singapore, pp. 197–211. doi:10.1007/978-981-10-3322-3_18.
[19] P. Singh, G. Dhiman, A fuzzy-lp approach in time series forecasting, in: B.U. Shankar, K. Ghosh, D.P. Mandal, S.S. Ray, D. Zhang, S.K. Pal (Eds.), Pattern Recognition and Machine Intelligence, Springer International Publishing, Cham, 2017, pp. 243–253.
[20] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-ii, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, doi:10.1109/4235.996017.
[21] C.A. Coello Coello, M.S. Lechuga, Mopso: A proposal for multiple objective particle swarm optimization, in: Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress - Volume 02, in: CEC '02, IEEE Computer Society, Washington, DC, USA, 2002, pp. 1051–1056.
[22] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731, doi:10.1109/TEVC.2007.892759.
[23] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, Trans. Evol. Comp 1 (1) (1997) 67–82, doi:10.1109/4235.585893.
[24] 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).
[25] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2) (2000) 173–195, doi:10.1162/106365600568202.
[26] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable Test Problems for Evolutionary Multiobjective Optimization, Springer London, London, pp. 105–145. doi:10.1007/1-84628-137-7_6.
[27] D.W. Corne, N.R. Jerram, J.D. Knowles, M.J. Oates, M. J, Pesa-ii: Region-based selection in evolutionary multiobjective optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001, Morgan Kaufmann Publishers, 2001, pp. 283–290.
[28] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Technical Report, 2001.
[29] D. Angus, C. Woodward, Multiple objective ant colony optimisation, Swarm Intell. 3 (1) (2009) 69–85, doi:10.1007/s11721-008-0022-4.
[30] C.A. Coello Coello, Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored, Front. Comput. Sci. China 3 (1) (2009) 18–30, doi:10.1007/s11704-009-0005-7.
[31] O. Schütze, M. Laumanns, C.A. Coello Coello, M. Dellnitz, E.-G. Talbi, Convergence of stochastic search algorithms to finite size pareto set approximations, J. Global Optim. 41 (4) (2008) 559–577, doi:10.1007/s10898-007-9265-7.
[32] F. Edgeworth, Mathematical physics: P.London, England: Keagan (1881).
[33] V. Pareto, Cours d'economie politique: Librairie Droz (1964).
[34] M. Gong, L. Jiao, H. Du, L. Bo, Multiobjective immune algorithm with nondominated neighbor-based selection, Evol. Comput. 16 (2) (2008) 225–255, doi:10.1162/evco.2008.16.2.225.
[35] D. Yang, L. Jiao, M. Gong, J. Feng, Adaptive ranks clone and k-nearest neighbor list-based immune multi-objective optimization, Comput. Intell. 26 (4) (2010) 359–385, doi:10.1111/j.1467-8640.2010.00363.x.
[36] P.M. Pradhan, G. Panda, Solving multiobjective problems using cat swarm optimization, Expert Syst Appl 39 (3) (2012) 2956–2964. https://doi.org/10.1016/j.eswa.2011.08.157.
[37] W. Lin, D. Yu, S. Wang, C. Zhang, S. Zhang, H. Tian, M. Luo, S. Liu, Multi-objective teaching-learning-based optimization algorithm for reducing carbon emissions and operation time in turning operations, Eng. Optim. 47 (7) (2015) 994–1007, doi:10.1080/0305215X.2014.928818.
[38] E. Hancer, B. Xue, M. Zhang, D. Karaboga, B. Akay, A multi-objective artificial bee colony approach to feature selection using fuzzy mutual information, in: 2015 IEEE Congress on Evolutionary Computation (CEC), 2015, pp. 2420–2427, doi:10.1109/CEC.2015.7257185.

[39] H. Hemmatian, A. Fereidoon, E. Assareh, Optimization of hybrid laminated composites using the multi-objective gravitational search algorithm (mogsa), Eng. Optim. 46 (9) (2014) 1169–1182, doi:10.1080/0305215X.2013.832234.
[40] R. Shang, H. Liu, L. Jiao, Multi-objective clustering technique based on k-nodes update policy and similarity matrix for mining communities in social networks, Physica A 486 (Supplement C) (2017) 1–24, doi:10.1016/j.physa.2017.05.026.
[41] D. Yang, L. Jiao, M. Gong, F. Liu, Artificial immune multi-objective sar image segmentation with fused complementary features, Inf. Sci. (Ny) 181 (13) (2011) 2797–2812, doi:10.1016/j.ins.2011.02.025. Including Special Section on Databases and Software Engineering
[42] S. Yang, M. Wang, L. jiao, A quantum particle swarm optimization, in: Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753), 1, 2004, pp. 320–324Vol.1, doi:10.1109/CEC.2004.1330874.
[43] W. Zhong, J. Liu, M. Xue, L. Jiao, A multiagent genetic algorithm for global numerical optimization, IEEE Trans. Syst,, Man, Cybern., Part B (Cybernetics) 34 (2) (2004) 1128–1141, doi:10.1109/TSMCB.2003.821456.
[44] L. Jiao, S. Member, L. Wang, A novel genetic algorithm based on immunity, IEEE Trans. on Systems, Man, and Cybernetics-Part A: Systems and Humans 30 (2000) 552–561.
[45] J.A.H. Mejia, O. Schütze, O. Cuate, A. Lara, K. Deb, Rds-nsga-ii: a memetic algorithm for reference point based multi-objective optimization, Eng. Optim. 49 (5) (2017) 828–845, doi:10.1080/0305215X.2016.1211127.
[46] A. Ilany, A.S. Booms, K.E. Holekamp, Topological effects of network structure on long-term social network dynamics in a wild mammal, Ecol. Lett. 18 (7) (2015) 687–695, doi:10.1111/ele.12447.
[47] C.A. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, Trans. Evol. Comp 8 (3) (2004) 256–279, doi:10.1109/TEVC.2004.826067.
[48] J.D. Knowles, D.W. Corne, Approximating the nondominated front using the pareto archived evolution strategy, Evol. Comput. 8 (2) (2000) 149–172, doi:10.1162/106365600568167.
[49] D.W. Corne, N.R. Jerram, J.D. Knowles, M.J. Oates, Pesa-ii: Region-based selection in evolutionary multiobjective optimization, in: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, Morgan Kaufmann Publishers Inc., 2001, pp. 283–290.
[50] M.N. Ab Wahab, S. Nefti-Meziani, A. Atyabi, A comprehensive review of swarm optimization algorithms, PLoS ONE 10 (5) (2015) 1–36, doi:10.1371/journal.pone.0122827.
[51] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, IEEE Trans. Evol. Comput. 3 (4) (1999) 257–271, doi:10.1109/4235.797969.
[52] C.A. Coello Coello, C. Dhaenens, L. Jourdan, Multi-Objective Combinatorial Optimization: Problematic and Context, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–21. doi:10.1007/978-3-642-11218-8_1.
[53] G. Rudolph, O. Schütze, C. Grimme, C. Domínguez-Medina, H. Trautmann, Optimal averaged hausdorff archives for bi-objective problems: theoretical and numerical results, Comput. Optim. Appl. 64 (2) (2016) 589–618, doi:10.1007/s10589-015-9815-8.
[54] O. Schütze, X. Esquivel, A. Lara, C.A.C. Coello, Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization, IEEE Trans. Evol. Comput. 16 (4) (2012) 504–522, doi:10.1109/TEVC.2011.2161872.
[55] O. Schütze, M. Laumanns, E. Tantar, C.A.C. Coello, E.G. Talbi, Computing gap free pareto front approximations with stochastic search algorithms, Evol. Comput. 18 (1) (2010) 65–96, doi:10.1162/evco.2010.18.1.18103.
[56] M. Li, J. Zheng, Spread Assessment for Evolutionary Multi-Objective Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 216–230. doi:10.1007/978-3-642-01020-0_20.
[57] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G. da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, Trans. Evol. Comp 7 (2) (2003) 117–132, doi:10.1109/TEVC.2003.810758.
[58] C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Methods Appl. Mech. Eng. 191 (11–12) (2002) 1245–1287. https://doi.org/10.1016/S0045-7825(01)00323-1.
[59] B. Kannan, S.N. Kramer, An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, J. Mech. Des. 116 (2) (1994) 405–411.
[60] A.H. Gandomi, X.-S. Yang, Benchmark problems in structural optimization, Springer Berlin Heidelberg (2011) 259–281, doi:10.1007/978-3-642-20859-1_12.
[61] G. Dhiman, A. Kaur, A hybrid algorithm based on particle swarm and spotted hyena optimizer for global optimization, Advances in Intelligent Systems and Computing, Springer, 2018. In press.
[62] G. Dhiman, A. Kaur, Spotted hyena optimizer for solving engineering design problems, in: International Conference on Machine Learning and Data Science, IEEE, 2018. In press.