Neural Computing and Applications (2020) 32:10359–10386
https://doi.org/10.1007/s00521-019-04575-1

ORIGINAL ARTICLE

A new algorithm for normal and large-scale optimization problems: Nomadic People Optimizer

Sinan Q. Salih[1,2] · AbdulRahman A. Alsewari[3]

Received: 12 December 2018 / Accepted: 16 October 2019 / Published online: 28 October 2019

Abstract

Metaheuristic algorithms have received much attention recently for solving different optimization and engineering problems. Most of these methods were inspired by nature or the behavior of certain swarms, such as birds, ants, bees, or even bats, while others were inspired by a specific social behavior such as colonies, or political ideologies. These algorithms faced an important issue, which is the balancing between the global search (exploration) and local search (exploitation) capabilities. In this research, a novel swarm-based metaheuristic algorithm which depends on the behavior of nomadic people was developed, it is called "Nomadic People Optimizer (NPO)". The proposed algorithm simulates the nature of these people in their movement and searches for sources of life (such as water or grass for grazing), and how they have lived hundreds of years, continuously migrating to the most comfortable and suitable places to live. The algorithm was primarily designed based on the multi-swarm approach, consisting of several clans and each clan looking for the best place, in other words, for the best solution depending on the position of their leader. The algorithm is validated based on 36 unconstrained benchmark functions. For the comparison purpose, six well-established nature-inspired algorithms are performed for evaluating the robustness of NPO algorithm. The proposed and the benchmark algorithms are tested for large-scale optimization problems which are associated with high-dimensional variability. The attained results demonstrated a remarkable solution for the NPO algorithm. In addition, the achieved results evidenced the potential high convergence, lower iterations, and less time-consuming required for finding the current best solution.

Keywords Nature-inspired algorithm · Metaheuristics · Nomadic People Optimizer · Benchmark test functions

1 Introduction

The process of optimization involves a holistic search for the optimal response to a given problem. Many fields of study, including economy, engineering, and medical fields, have inherent problems that require optimization problems. The development of the optimization algorithms has been the focus of many researchers globally. The primary aim of optimization algorithms (also known as search methods) is the establishment of an optimal solution to an optimization problem in such a way that the given quantity is optimized subjected to a possible set of constraints [1, 2].

Though this is a simple definition of optimization, it conceals several complex issues [3]. Some of the issues concealed in this definition include (a) there may be a combination of different types of data in the solution; (b) the search area may be restricted by nonlinear constraints; (c) the convolution of the search space with many individual solutions; (d) the tendency of the features of the problem changing with time; and (e) the presence of conflicting objectives in the optimized quantity. These are some of the problems that portray the complexities that an optimization algorithm may encounter.

In the process of solving optimization problems with a high-dimensional search space, it is impossible to achieve

✉ Sinan Q. Salih
sinanq.salih@duytan.edu.vn

1 Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam

2 College of Computer Science and Information Technology, University of Anbar, Ramadi, Al-Anbar, Iraq

3 Faculty of Computing, College of Computing and Applied Sciences, University Malaysia Pahang, 26300 Gambang, Pahang, Malaysia

a suitable solution with the classical optimization algorithms due to the exponential increase in the search space with the size of the problem. Therefore, it is not feasible to solve high-dimensional search space optimization problems using exact techniques such as exhaustive search [4]. Another problem of the classic optimization algorithms is their inability to find sufficient global optima (local optima stagnation). Furthermore, some of the classical optimization algorithms need search space derivation as well. It is, therefore, pertinent that these classical algorithms cannot adequately solve real-world optimization problems [5, 6].

Metaheuristic algorithms are currently being used as the primary approach to achieving optimal solutions to real optimization issues [7, 8]. These approaches mainly benefit from the stochastic operators that distinguish them from the deterministic algorithms [9] which reliably establishes the solution to a given problem using similar starting points. There are several engineering applications have evidenced the potential of the metaheuristic algorithms for optimization process [10–17]. However, the main problem associated with the optimization algorithm is the local optima entrapment which is regarded as a problem for the deterministic approach. Local optima stagnation is the ability of an optimization algorithm to find just the local solutions to a problem and consequently failing to find the true global solution (optimum). Since there are many local solutions in real problems, it may be difficult to reliably find the global optimum using deterministic algorithms [18].

Metaheuristics are used these days for several purposes, such as enhancing system performance and reducing product cost to meet market demands. Optimization, from the engineering perspective, involves fine-tuning one or more system parameters to achieve optimal system performance. This is an important task, especially when faced with a complicated and highly dimensional problem space [19]. In recent literature, many different metaheuristics have been proposed and successfully used for solving different engineering problems [20–28].

The metaheuristic algorithms may be classified using many criteria and this may be illustrated by their classification based on their features with respect to their search path, memory usage, the type of neighborhood exploration used, and the current number of solutions carried from one iteration to the next. In the literature, the metaheuristic algorithms are fundamentally classified into the single-solution based metaheuristics (SSBM) and the population-based metaheuristics (PBM). Generally, the SSBMs are exploitative-oriented, while the PBMs are more explorative-oriented. Figure 1 illustrates the classification of metaheuristics based on their number of solutions.

Furthermore, the metaheuristic algorithms are categorized into the swarm-based algorithms (SBA), physics-based algorithms (PBAs), and evolutionary algorithms (EAs). The EA is inspired from natural evolutionary behaviors. Some of the evolutionarily inspired metaheuristic algorithms are evolution strategy (ES), differential evolution (DE), genetic programming (GP), genetic algorithm (GA), evolutionary programming (EP), probability-based incremental learning (PBIL), and biogeography-based optimization (BBO) [29–32].

The SBAs are the next category; they are inspired by social behaviors of living groups [33–35]. Some of the common swarm-based algorithms include the particle swarm optimization (PSO) which was inspired by the individual and social behavior of birds, the cuckoo search (CS) which mimics the unusual egg-laying behavior, the firefly algorithm (FA) which was inspired by the characteristics light flashes from fireflies, the artificial bee colony (ABC) which was inspired by the behavior of bee swarms when searching for food, the Grey Wolf Optimizer (GWO) which mimics the behavior of Grey wolves when hunting preys, and the Whale Optimization Algorithm (WOA) which mimics the social behavior of the Humpback whales [20, 36–44].

Some authors have introduced a new swarm intelligence category known as social-inspired metaheuristic algorithms. The algorithms belong to this category are inspired from the social and cultural interactions seen in human behaviors. The common algorithms in this category include Teaching–Learning-based Optimization (TLBO), Socio Evolution & Learning Optimization Algorithm (SELO), Cultural Evolution Algorithm (CEA), Artificial Memory Optimization (AMO), and Human mental search (HMS) [5, 45–47]. The physics-based algorithms mimic the physical rules of nature. The common algorithms in this category include the water cycle (WC), gravitational search algorithm (GSA), Lightning Attachment Procedure Optimization (LAPO), simulated annealing (SA) and mine blast (MB) algorithms [4, 48–52].

A metaheuristic approach will successfully optimize a given problem only if the right balance between exploration (diversification) and exploitation (intensification) can be established. Exploitation is necessary for identifying the search parts that have quality solutions and also important for the intensification of the search in some of the potential accumulated search areas. The existing metaheuristic algorithms differ in the way they try to strike the balance between exploration and exploitation [53, 54].

The existed literature suggested the effectiveness of the metaheuristics in solving several design problems and points toward their ability to solve highly complex NP-hard problems searching [53, 55–62]. However, there is still lack of studies focusing on large-scale multidimensional problems.

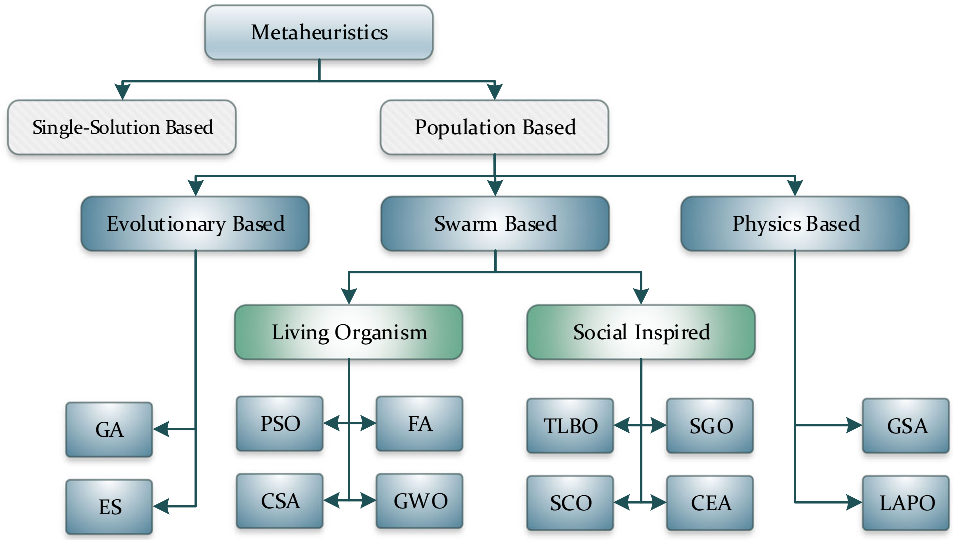


Fig. 1 Classification of metaheuristics based on number of solutions

Furthermore, tuning of control parameters can also a relevant issue as far as the existing metaheuristic is concerned. To be specific, the tuning process can be painstakingly difficult even for a small dimension problem, let alone dealing with large-scale multidimensional problems. Specifically, poor tuning of the control parameters leads to inefficient exploration and exploitation, hence, affecting the performance of the metaheuristic algorithm at hand. Therefore, a parameter-free metaheuristic is well desired in terms of reducing the complexity of parameter tuning and can be used in different domains without any additional adaptive methods.

To overcome the above-mentioned drawbacks in the existing metaheuristics, a novel parameter-free multi-swarm metaheuristic is proposed in this paper. It is a social-based algorithm inspired by the movement of nomads when searching for the sources of food in the desert. The proposed algorithm is known as the Nomadic People Optimizer (NPO) algorithm. The proposed algorithm with its unique structure has the ability to handle large-scale problems.

The paper is organized as follows: Sect. 2 is divided into two parts, first part explains the inspiration of the proposed algorithm, while the second part explains NPO algorithm with the developed operators, and the mathematical model, together with the pseudocode of the proposed NPO algorithm. Section 3 contains the results and discusses the statistical results for the standard benchmarking functions. Finally, Sect. 4 provides the conclusions from the study.

2 Nomadic People Optimizer

This section explained the fundamentals of NPO, beginning with the definition of the nomadic people and their lifestyle which formed the basis of the proposed algorithm. Moreover, the mathematical model for NPO was provided.

2.1 Nomad people inhabit pattern: source of inspiration

Nomads refer to those who live their entire life traveling from one place to another with their herds of camels, cattle, and sheep in search of natural sources of water and food. These herds graze on pastures close to water sources and provide their owners food, as well as other major necessities, such as skin and wool for clothing and tent-making. The milk from the herd serves as a source of calcium and protein for the nomads. It is well-known that nomads do not familiarize themselves with an environment or cultivate the lands within their settlement as they do not settle in one place for a long time. In fact, the nomads can be categorized into several types, such as the Berbers, the Gypsy, and the Bedouins.

The Bedouins' classification and lifestyle inspired the new algorithm, NPO. The Bedouins' families are made up of the Sheikh family and normal families. The role of the Sheikh is usually hereditary (from father to son) or in situations of conflict where a normal family may take over power from the Sheikh's family if the normal family becomes more influential. The sheikh as the leader of the clan determines the locations essential for survival and pattern of distribution for the families of the rest of the clan. The sheikh would send the families in search of a new suitable location. The selected families would move randomly in different directions and distances. When a family finds a better place, the Sheikh moves toward the new position and re-establishes the clan (i.e., the normal families) in a semicircular shape around his tent, Fig. 2 illustrates the distributions of the families. The distributions of the families' tents are in a semicircular pattern, with the Sheikh's tent at the center. The sheikh is the central figure with the authority not only over the families and their fates, but also when and where the clan should move. In times of conflict, the sheikh is responsible in deciding whether the variances between clans are either resolved peacefully or in fights.

The Bedouins continuously travel in search of location rich with resources necessary to sustain their livelihood. Migration occurs annually during summer and winter as slight climatic and territorial changes are exploited using seasonal or periodic movements between the summer and winter pasture areas (SPA and WPA). The movement of Bedouin clans is scattered over the deserts of the Arabian Peninsula, Western Iraq, parts of Syria, Jordan, Palestine, Egypt, and North Africa as illustrated in Fig. 3. SPA is usually determined by the availability of water and pasture sources, as well as suitable climatic conditions. WPA aims for places with wells and dams, which could be either small or closed areas. The occupation of SPA is between the period of May and October (late harvest period) and the clan will later move to WPA for the rest of the year.

2.2 Mathematical model of NPO

2.2.1 NPO terminology

The terminologies used to describe NPO are discussed below:

1. Leader (σ): An individual represents the current local best solution in the swarm.
2. Best Leader (σ^E): An individual represents the global best solution in all swarms, which is used in the meeting room approach.
3. Normal Leader (σ^N): An individual represents the other leaders except the Best Leader (σ^E).
4. Family (x): An individual represents a member in the swarm or clan which has a lower fitness value than the leader.
5. Clan (c): a group of families (x), including the Leader (σ), which represents an individual swarm. NPO consists of several clans, each clan consists of several families and single Leader.
6. Fitness or Objective Function ($f(x)$): a term refers to the function or method to evaluate the goodness of a position in the search space. It takes the coordinates in the solution space and returns a numerical value (goodness). The fitness function provides an interface between the physical problem and the optimization algorithm.
7. Direction (Ψ): It is a variable used for guiding the Normal Leaders toward the Best Leaders.

2.2.2 NPO algorithm

The NPO algorithm is comprised of five main operators, which are (1) initial meeting, (2) semicircular distribution, (3) families searching, (4) leadership transition, and lastly, 5) periodical meeting.

1. Initial meeting (initialization):

A set of Leaders (σ), where $\sigma_i = \{\sigma_1, \sigma_2, \ldots, \#Clans\}$ is initialized randomly by using the following equation:

$$\vec{\sigma_c} = (\mathrm{UB} - \mathrm{LB}) \times Rand + \mathrm{LB} \tag{1}$$

where UB and LB represent the upper bound and lower bound, respectively, while *Rand* denotes a random value between 0 and 1, and $\vec{\sigma_c}$ represents the position of the leader of the clan c.

2. Semicircular distribution (Local Search—Exploitation):

A set of families (x), where $X_i = \{X_1, X_2, \ldots, \#$ Families$\}$ is distributed around the corresponding leader σ. Mathematically, it is possible to distribute points randomly within a given circle with a known radius using the equations of the 2D circle. These points are circled around the origin (center of the circle) by the value of the angle, as given in the following equations:

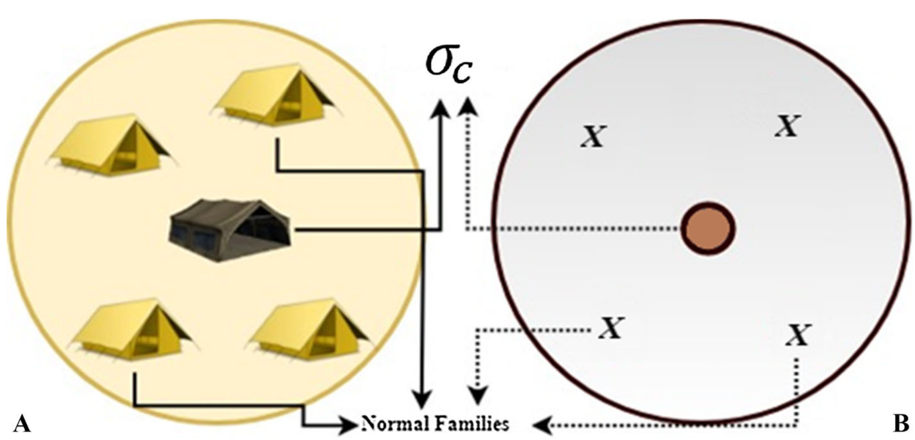


Fig. 2 Semicircular distribution of the families

Fig. 3 The distribution of the Bedouins over Arabic countries

$$X = \left(Rd \times \sqrt{R_1}\right) \times \cos(\theta) + X_0 \quad (2)$$

$$Y = \left(Rd \times \sqrt{R_2}\right) \times \sin(\theta) + Y_0 \quad (3)$$

where X_0 and Y_0 represent the coordinates of the origin point (center of the circle), while R_1 and R_2 denote the random coordinates of a point within the perimeter of that circle. Meanwhile, θ refers to the angle value of that point, which is a random value lies between [0, 2π].

Equations 2 and 3 are used when the generated points are within a circle in 2D shape, (i.e., X, Y, and θ). Nevertheless, if the solutions are represented within the search space, the problems do not require any X and Y coordinates. Hence, the representation of the solutions is unary (i.e., single dimension), instead of 2D. As such, the distribution of tents randomly around the leader's tent requires an X coordinate, while excluding the non-required Y coordinate. With that, the equation was developed to fit this scenario, as given in Eq. 4 in the following:

$$\overrightarrow{X_c} = \overrightarrow{\sigma_c} \times \sqrt{R} \times \cos(\theta) \quad (4)$$

where $\overrightarrow{X_c}$ represents the position of a family, $\overrightarrow{\sigma_c}$ represents the position of the leader for the same swarm—or clan (c), and R represents a random number in range [0, 1]. In some cases, where the LB is 0 or positive, the equation is multiplied by $|cos(\theta)|$. It can be noted from Eq. 4 that the position of the families is based entirely on the position of their leader, and this is within the powers of the leader since he is in charge of distributing the families around his tent.

3. Families searching (global search—exploration):

In NPO, the exploration part is executed when there is no new local best solution in the swarm. In such situations, the families search for better positions far from the current local best. All families move in different directions in the search space based on random steps and directions generated by Lévy Flight formula as follows:

$$\overrightarrow{X_i^{\text{new}}} = \overrightarrow{X_i^{\text{old}}} + \left(a_c * \overrightarrow{\left(\sigma_c - X_i^{\text{old}}\right)} \oplus \text{Levy}\right) \quad (5)$$

where $\overrightarrow{X_i^{\text{new}}}$ and $\overrightarrow{X_i^{\text{old}}}$ represent the new and old positions of the current family respectively, a_c represent the area of the clan which is the average distance between all the normal families and σ_c. a_c can be calculated using the following equation: -

$$a_c = \frac{\sum_{i=1}^{\Phi} \sqrt{\left(\overrightarrow{\sigma_c} - \overrightarrow{X_i^{\text{old}}}\right)^2}}{\Phi} \quad (6)$$

where Φ denotes the number of families in each clan, $\overrightarrow{\sigma_c}$ and x_c^i represent the positions of the leader and the normal families, respectively. The distance between x_c and σ_c gets closer when families are distributed around σ_c in a small circle (i.e., semicircular distribution), which leads to explore the search space by small step size. While the large distances between all x_c and σ_c enhances the ability of the families to explore the search space far from current σ_c. Thus, the value of a_c has a great effect on the searching process.

The families move in different directions, and in random step sizes; the step sizes are generated by the Lévy flight (λ_c) equation as follows:

$$\text{Levy} \sim u = t^{-\lambda} \quad (1 < \lambda \leq 3) \quad (7)$$

The Lévy flight equation is usually used to generate a random walk while drawing the random step length from a Lévy distribution with an infinite mean and variance [37]. The stochastic equation for random walk is typically represented in Eq. 5. A random walk is generally a Markov chain that depends on its current location (the first term in the above equation) and the transition probability (the second term) to determine its next status/location. The product $\oplus$ means entry-wise multiplications. Here, the random walk via Lévy flight is more efficient in exploring the search space as its step length is much longer in the long run.

4. Leadership transition (exploitation):

For each clan, check whether there is a new family that has a better fitness than the leader of the same clan, then, the family becomes a leader and vice versa.

5. The periodical meetings (exploitation–exploration):

The periodical meetings are dissimilar to the initial meeting, except for the redistribution of Leaders in the desert. During these periodical meetings, the Leaders strive to resolve any external problem and discuss the best locations for relocation purpose. The reason for this meeting is to enable each Leaders to have control over his place, but without arousing the ambitions of others, instead, bringing them closer to himself.

The periodic meetings occur in two stages and they involve only the Leaders. The families are disallowed from interfering, except those in power. The first phase of the meeting is to determine the most powerful Leader, or in precise, the Leader of the best location who will propose solutions to other Sheikhs for them to update their locations. This update is performed by adding the variance between the position of the strongest Leader and that of the normal Leader as depicted in the following equation:

$$\Delta\mathrm{Pos} = \Psi\left(\frac{\sqrt{\sum_i^D \left(\sigma^E - \sigma_c^N\right)^2}}{\#D}\right) \tag{8}$$

where σ^E represents the position of the best Leader, and σ_{Ni} denotes the position of the normal Leaders. Meanwhile, $\#D$ is the number of dimensions of the problem, Ψ refers to the direction, and $\Delta\mathrm{Pos}$ represents the normalized distance between the best Leader and the normal Leader. The direction variable Ψ guides the normal Leaders to better positions depending on the fitness value of the best sheikh, as follows:

$$\Psi = \begin{cases} 1 & \text{if } f(\sigma^E) \geq 0 \\ -1 & \text{otherwise} \end{cases} \tag{9}$$

The normal Leaders update their positions via Eq. 10. This equation represents a part of the exploration stage in NPO.

$$\overrightarrow{\sigma_c^{\mathrm{new}}} = \overrightarrow{\sigma_c^N} + \Delta\mathrm{Pos}\left(\sigma^E - \sigma_c^N\right) * \frac{IT}{\#T} \tag{10}$$

where σ_c^{new} and σ_c^N represent the new and old position of the normal Leader, respectively, while IT and $\#T$ represent the current iteration and the total number of iterations, respectively.

During the periodical meeting, the positions of all normal Leaders are updated. The Leader stays at the new position if it is better than before, apart from establishing his new clan based on the second step (semicircular distribution), otherwise he returns to the old position. It is important to mention that the periodical meeting is a unique method of sharing information between swarms, for it reflects a cooperative scheme for multi-swarms. As mentioned before, each clan represents an individual swarm, while the periodical meeting facilitates communication between them. This cooperative multi-swarm scheme is called Meeting Room Approach (MRA). MRA can be applied with other metaheuristics, it helps them to balance between exploration and exploitation, which promotes faster convergence, in comparison to other standard versions of the algorithms. The pseudocode of the periodical meeting or MRA is given below:

Algorithm: Periodical Meeting or Meeting Room Approach

Input: All Leaders σ
Output: Best Leader Ever σ^E, Updated Positions for all Normal Leaders
Procedure:
  Determine the best leader ever as σ^E
  Determine the value of the direction variable Ψ via $eq. 9$
  Calculate ΔPos via $eq. 8$
  For each normal leader σ_c^N
    Move towards the best leader ever σ^E, via $eq. 10$
    Calculate the fitness value for each σ_c^{new} using the objective function
    If: the σ_c^{new} is better than the σ_c^N, ***Then*** keep it
    Else: keep the σ_c^N
  End For
  Return σ^E and other σ

Figure 4 illustrates the main structure of the MRA, where several clans interact with each other by sharing their positions. Therefore, MRA represents a social learning approach between the leaders or the best solution in each clan. The black circles represent the best solutions or the leaders, while the white circles represent the families.

The main steps of NPO are summarized in the following pseudocode, while the flowchart is given in Fig. 5.

Algorithm: Nomadic People Optimizer (NPO)

Input: No. of Clans (#Clans), No. of Families (Φ), No. of Iterations ($\#T$)
Output: The Best Sheikh
Procedure:
Define the objective function $f(x)$,
Initialize the Leaders σ_c^O , $c = \{1, 2, 3, \ldots, \#Clans\}$
Calculate the fitness value for each leader via $f(x)$
Repeat (Itr):
  For c = 1 to $\#Clan$
    Distribute the solutions/families around the leader in a semi-circular shape via $eq. 4$
    Calculate the fitness value for each solution x_i^c via $f(x)$
    Set the best x_i^c in the clan c as σ_c^B
    If σ_c^B is better than the original σ_c^O Then, Swap them $\sigma_c^O = \sigma_c^B$
    Else: Explore the search space using the following steps:
        Calculate the average distance between all families via $eq. 6$
        Move the family towards the new position via $eq. 5$
        Calculate the fitness value for each solution x_i^cvia $f(x)$
        Set the best x_i^c in the clan c as σ_c^B
        If σ_c^B is better than the original σ_c^O Then, Swap them $\sigma_c^O = \sigma_c^B$
    End if
  End For
  Implement the Periodical Meeting
Loop Until $(Itr > \#T)$
Return σ^E

3 Algorithm results, discussion, and evaluation settings

The performance of NPO was evaluated by carrying out two sets of experiments. The first experimental set looked into the overall performance of the algorithms over a fixed number of iterations. Upon completing certain number of iterations, the performance of the algorithms was evaluated based on the mean and the best fitness values found for each benchmarked function. Next, the second experimental set investigated the convergence behavior of the algorithms. In this case, the algorithm was run on various numbers of iterations to evaluate the mean fitness value established for each case. Hence, the convergence behavior of the algorithms based on the number of iterations was obtained. The proposed NPO algorithm in this study was applied on new combination of benchmarked functions, while its performance was compared to that of six well-known algorithms metaheuristics (particle swarm optimization (PSO2011) [63], artificial bee colony (ABC) [64], flower pollination algorithm (FPA) [65], Grey Wolf Optimizer (GWO) [41], Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [29], and Firefly Algorithm (FFA) [38]).

3.1 Benchmark functions

It is mandatory that the performance of any newly developed algorithm should be benchmarked and validated against that of other existing algorithms using a good set of test functions. Most researchers prefer to test the performance of their algorithms on a large test set, especially when optimization functions are involved. However, the effectiveness of one algorithm over others cannot solely depend on its ability to solve problems that are either too specialized or without diverse features. The evaluation of an algorithm demands the identification of the kind of problems that it had a better performance compared to others. This will help in determining the type of problems that the algorithm can be used to solve. This can only be achieved by using a test suite that is large enough to embrace a range of problems such as unimodal, multimodal, separable, non-separable, and multidimensional problems [66, 67].

This study focused on the test function benchmarks and their diverse features such as modality and separability. A function is multimodal if it has more than one local optimum and is used to test the ability of an algorithm to

Fig. 4 Meeting room approach

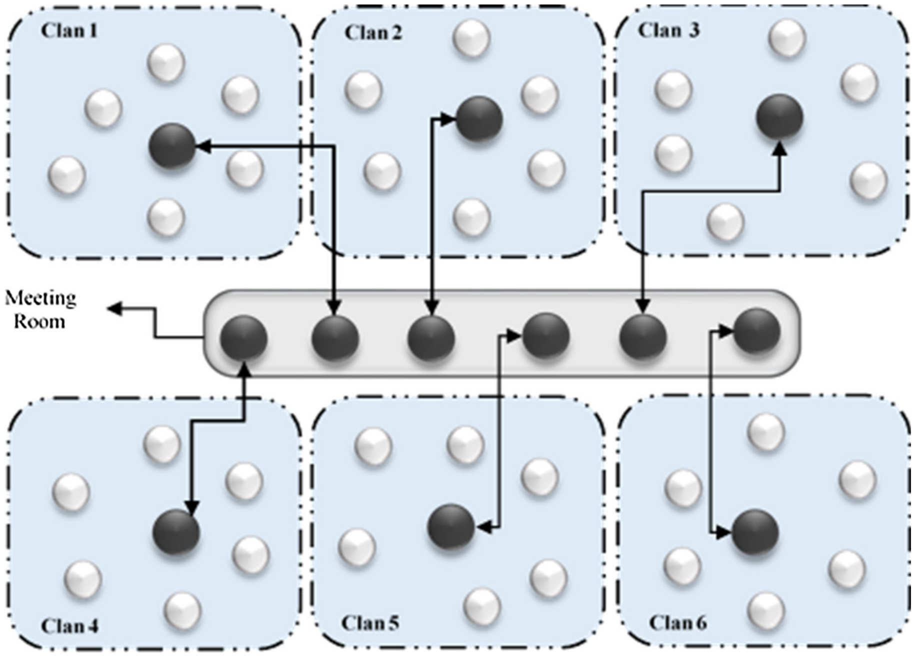
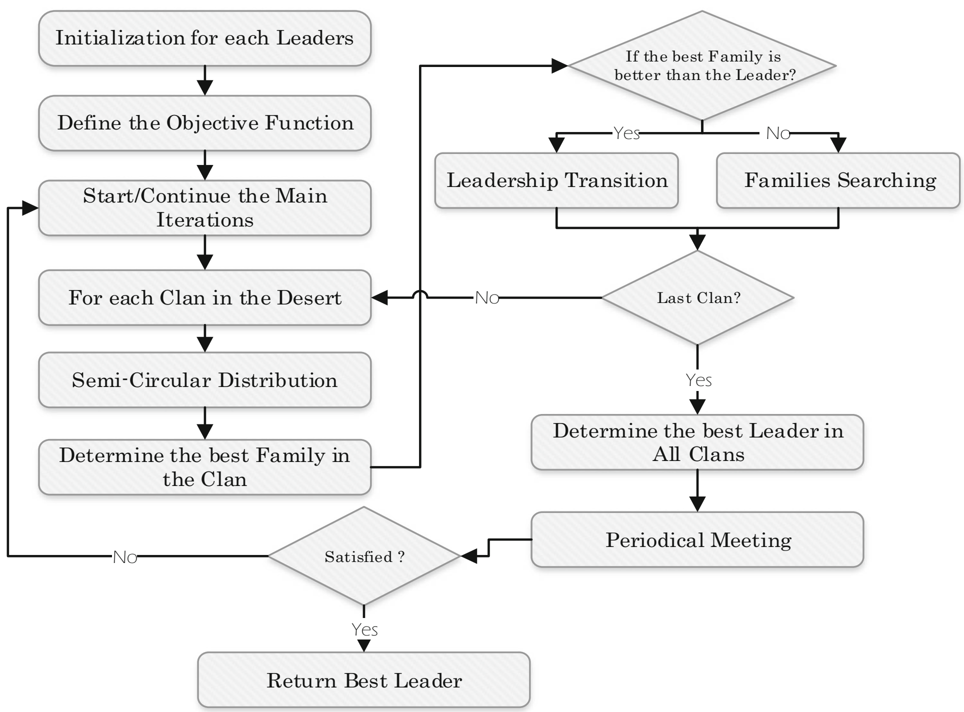


Fig. 5 Flowchart of NPO



escape being trapped in any local minima. If an algorithm is built with a poorly constructed exploration process, it cannot effectively search the function landscape, and this could result in having the algorithm stuck at local minima. The most difficult class of problems for most algorithms is the multimodal functions with many local minima. The difficulty of different benchmark functions is expressed in terms of their reparability. Because each variable of a function in separable functions is independent of the other variables, they are generally easily solved compared to their inseparable counterpart.

To evaluate the performance of the NPO, 36 test functions were carefully selected in this study from several references [66, 68]. These test functions were divided into four groups (Unimodal Non-Separable (U-N) with 9 tests, Unimodal Separable (U-S) with 6 tests, Multimodal Non-Separable (M-N) with 13 tests, and Multimodal Separable M-S) with 8 tests. Table 1 shows these functions. The table presents the name of the test function, the type (U-S, U-N, M-N, or M-S), the number of decision variables or dimensions (Dim), the lower and lower boundaries of the test function (LB and UB), and the optimal solution (Opt.).

3.2 Algorithms comparison and simulation settings

All the experiments, including NPO and the previously mentioned 36 benchmark test functions were executed on a personal computer (Core i7, 3.60 GHz, 16 GB of RAM, 64-bit Windows 10 operating system) using MATLAB 2014a. The performance of NPO was compared to PSO2011, ABC, FPA, GWO, CMA-ES, and FFA. The experiments were executed in 30 different runs, and the best, worst, median, mean, and standard deviation were recorded. Table 2 presents the specific/default parameters for the metaheuristics mentioned above.

NPO was compared to the other metaheuristics based on the mentioned statistical parameters based on 30 run times. Also, NPO is compared to the other metaheuristics based on statistical test which is Wilcoxon signed-rank test. To establish the speed of the NPO in converging to the optimal solution, a convergence analysis for all the algorithms was performed. The results of the 30 runtimes (means and standard deviation) are compared to those of the mentioned metaheuristics. To evaluate the ability of NPO algorithm when solving the large-scale problems, 13 test functions were selected with three different number of decision variables—or dimensions—, they are (100, 500, and 2000).

3.3 Benchmarking results

This section presents the results of proposed NPO, it is divided into two subsections. In the first subsection, the performance of NPO over the unconstrained test function is presented, while the second subsections present the performance of NPO over the large-scale problems.

3.3.1 NPO for unconstrained test functions

After executing and recording all the experiments over the 36 benchmark test functions, the outcomes showed that the NPO exerted superior performance and could reach the optimal solution for many test functions. Although some of these test functions had been exceptionally challenging to solve and their best results could not be efficiently arrived at with the NPO, the algorithm was able to reach values very close to their ideal best results. Table 3 presents the results of NPO and the other six metaheuristics over the 36 test functions.

Table 4 summarizes the results and the comparison with other metaheuristics. The table also depicts the number of test functions that had been solved via NPO. The symbol '+' represents the number of test functions where NPO exhibited better results while '−' denotes the worst results, '=' reflects both algorithms with similar good or bad results, finally, '*' refers to the number of test functions where NPO reached the optimal solution. From the table, it is evident that NPO had successfully outperformed the other metaheuristic by 45.8%.

Although the statistical results presented in Tables 3 and 4 provided a first insight into the performance of NPO, a pairwise statistical test is typically used for a better comparison. For this purpose, by using the results obtained from 30 runs of each algorithm, a Wilcoxon Signed-Rank Test is performed with a statistical significance value $\propto = 0.05$. The null hypothesis H_0 for this test is: "There is no difference between the median of the solutions produced by algorithm A and the median of the solutions produced by algorithm B for the same benchmark problem". i.e., median (A) = median (B). To determine whether algorithm A reached a statistically better solution than algorithm B, or if not, whether the alternative hypothesis is valid, the size of the ranks provided by the Wilcoxon Signed-Rank Test (i.e., T+, and T−) are examined.

In Table 5, the statistical pairwise results of the NPO algorithm compared to those of other algorithms are given. In this table, the asterisk (*) indicates that a p value of less than 0.05, which means there is a significant difference between the two algorithms in that test. The legends used in this test are:

(a) The sum of negative ranks equals the sum of positive ranks.
(b) Based on positive ranks.
(c) Based on negative ranks.

Table 5 displayed the Wilcoxon signed-rank test for comparison of the 30 runs of each metaheuristic. The table can be summarized as follows:

- NPO versus PSO: The test indicates that there are more significant negative ranks ($N = 31$) rather than significant positive ranks ($P = 5$). Meaning that the median of NPO is more than median of PSO, in other words, the H_0 is rejected and the NPO has superior performance and has outperformed PSO.
- NPO versus ABC: The test indicates that there are more significant negative ranks ($N = 18$) rather than significant positive ranks ($P = 13$). Meaning that the median of NPO is more than median of ABC, in other words,

the H_0 is rejected and NPO is statistically better than ABC algorithm.

- NPO versus FPA: The test indicates that there are more significant negative ranks ($N = 14$) rather than significant positive ranks ($P = 12$). Meaning that the median of NPO is more than median of FPA, in other words, the H_0 is rejected and NPO has better performance than FPA.
- NPO versus GWO: The test indicates that there more significant negative ranks ($N = 13$) rather than significant positive ranks ($P = 11$). This means that the median of NPO is more than median of GWO, in other words, the H_0 is rejected and the NPO is better than GWO, however, there are 12 tests where both have equal results.
- NPO versus CMA-ES: The test indicates that there more significant negative ranks ($N = 20$) rather than significant positive ranks ($P = 12$). This means that the median of NPO is more than median of CMA-ES, in other words, the H_0 is rejected and the NPO is better than GWO.
- NPO versus FFA: The test indicates that there more significant negative ranks ($N = 19$) rather than significant positive ranks ($P = 11$). This means that the median of NPO is more than median of FFA in other words, the H_0 is rejected, and the NPO statistically is better than FFA.

Table 2 The specific parameters used in the studied metaheuristic

Algorithm	Parameter	Settings
PSO2011	Swarm size S.S	50
	Inertia weight ω	Linearly decrease (0.9–0.1)
	Cognitive parameter $c1$	1.49
	Social parameter $c2$	1.49
ABC	Colony size C.S	50
	No. Food Source	C.S/2
	Limit	50
FPA	Swarm size S.S	50
	Switch probability P	0.8
	Levy flight λ	1.5
GWO	Swarm size S.S	50
	a	Linearly decrease (2–0.1)
CMA-ES	Initial point $Xmean$	$Rand$(1, D)
	Step size (Sigma σ)	0.5
	Population size (lambda λ)	$4 + Floor(3 \times Log(D))$
	Mutation (μ)	$\lambda/2$
FFA	Swarm size S.S	50
	a	0.5
	β_{min}	0.2
	γ	1.0
	δ	0.96
NPO	Swarm size ($\sigma \times \#F$)	50 (5 $\times$ 10)

Table 1 Benchmark test functions used for evaluation

f_n	Name	Type	Dim	LB, UB	Opt.	f_n	Name	Type	Dim	UB, LB	Opt.
f_1	Beale	U-N	2	− 4.5,4.5	0	f_{19}	Cross-in-tray	M-N	2	− 10, 10	− 2.0626
f_2	Easom	U-N	2	− 10,10	− 1	f_{20}	Griewank	M-N	30	− 600, 600	0
f_3	Matyas	U-N	2	− 10,10	0	f_{21}	GoldStein-Price	M-N	2	− 2, 2	3
f_4	Powell	U-N	24	− 4, 5	0	f_{22}	Hartman 3	M-N	3	0, 1	− 3.8627
f_5	Schaffer No.1	U-N	2	− 100, 100	0	f_{23}	Hartman 6	M-N	6	0, 1	− 3.3223
f_6	Schaffer No.3	U-N	2	− 100, 100	0.001567	f_{24}	Penalized	M-N	30	− 50, 50	0
f_7	Schaffer No.4	U-N	2	− 100, 100	0.29258	f_{25}	Penalized No.2	M-N	30	− 50, 50	0
f_8	Zakhrov	U-N	30	− 5, 10	0	f_{26}	Perm	M-N	4	− 4, 4	0
f_9	Quartic	U-N	30	− 1.28,1.28	0	f_{27}	Powersum	M-N	4	0, 4	0
f_{10}	Schwefel 2.21	U-S	30	− 100, 100	0	f_{28}	Shubert	M-N	2	− 10, 10	− 186.7309
f_{11}	Schwefel 2.22	U-S	30	− 10, 10	0	f_{29}	Alpine No.1	M-S	30	− 10, 10	0
f_{12}	Sphere	U-S	30	− 100, 100	0	f_{30}	BohachevskyNo.1	M-S	2	− 100, 100	0
f_{13}	Step2	U-S	30	− 100, 100	0	f_{31}	Booth	M-S	2	− 10, 10	0
f_{14}	Stepint	U-S	5	− 5.12, 5.12	0	f_{32}	Branin	M-S	2	− 5, 5	0.39789
f_{15}	Sumsquares	U-S	30	− 10, 10	0	f_{33}	Michalewics 2	M-S	2	0, π	− 1.8013
f_{16}	Ackley	M-N	30	− 32, 32	0	f_{34}	Michalewics 5	M-S	5	0, π	− 4.6876
f_{17}	BohachevskyNo.2	M-N	2	− 10, 10	0	f_{35}	Michalewics 10	M-S	10	0, π	− 9.6601
f_{18}	BohachevskyNo.3	M-N	2	− 100,100	0	f_{36}	Rastrigin	M-S	30	− 5.12, 5.12	0

Table 3 Results of the metaheuristics over benchmark test functions

f_n	Statistic	PSO2011	ABC	FPA	GWO	CMA-ES	FFA	NPO
1	Best	2.7457E−06	1.55E−12	0	5.56E−10	0	2.56E−12	7.7702e−12
	Worst	0.0014077	1.31E−08	0	0.76207	0.47422	1.83E−09	0.000948752
	Median	0.000090535	2.45E−10	0	2.48E−08	0.29947	2.35E−10	2.64811E−05
	Mean	0.00020231	1.15E−09	0	0.025402	0.25402	4.04E−10	2.1461e−04
	S.D	0.00028704	2.63E−09	0	0.13913	0.46538	4.78E−10	5.2267e−04
2	Best	− 0.99974	− 1	− 1	− 1	− 1	− 1	− 1
	Worst	− 0.95692	− 0.998047865	− 1	− 1	− 0.9348	0	− 1
	Median	− 0.99655	− 1	− 1	− 1	− 0.9784	− 1	− 1
	Mean	− 0.99273	− 0.999917119	− 1	− 1	− 0.9331	− 0.7.33	− 1
	S.D	0.009772	3.59E−04	0	1.51E−07	0.0471	0.4502	0
3	Best	3.5745E−08	4.31E−15	9.96E−57	4.94E−247	0	2.45E−12	0
	Worst	0.000055609	1.36E−11	5.91E−47	9.01E−210	0	4.68E−10	0
	Median	7.4128E−06	1.36E−12	1.36E−50	7.65E−222	0	7.97E−11	0
	Mean	0.000013438	2.57E−12	3.14E−48	3.02E−211	0	1.21E−10	0
	S.D	0.000014366	3.66E−12	1.11E−47	0	0	1.22E−10	0
4	Best	1.44E−01	3.77E−05	0.43335	4.43E−13	0.0034	4.17E−09	0
	Worst	1.76E+01	7.45E−03	14.5569	1.66E−08	0.0010	2.96E−06	3.94E−39
	Median	3.56E+00	1.53E−03	3.5848	1.64E−09	0.0033	1.90E−07	3.89E−153
	Mean	4.48E+00	2.13E−03	4.4113	3.01E−09	0.0030	4.49E−07	3.50E−20
	S.D	4.07E+00	1.98E−03	3.411	3.82E−09	0.0011	6.53E−07	1.31E−40
5	Best	3.75E−07	0	0	0	0	3.05E−12	0
	Worst	6.14E−04	0	0	0	0	2.50E−03	0
	Median	1.70E−05	0	0	0	0	5.30E−11	0
	Mean	4.80E−05	0	0	0	0	8.32E−05	0
	S.D	1.13E−04	0	0	0	0	4.56E−04	0
6	Best	0.0015737	0.001566855	0.0015669	0.001566855	0.0015669	0.0015669	0.0015669
	Worst	0.0094556	0.001587719	0.0015672	0.001567308	0.0015671	0.001567	0.0015669
	Median	0.0024329	0.001567421	0.0015669	0.001566888	0.0015669	0.0015669	0.0015669
	Mean	0.0035316	0.00156986	0.0015669	0.001566911	0.0015669	0.0015669	0.0015669
	S.D	0.002101	4.71E−06	8.60E−08	8.83E−08	4.02E−08	3.047E−08	1.36E−08
7	Best	0.29259	0.292578645	0.29258	0.29258	0.29258	0.29258	0.29258
	Worst	0.29659	0.292589884	0.29258	0.29258	0.29661	0.29271	0.29258
	Median	0.29391	0.292579232	0.29258	0.29258	0.29258	0.2926	0.29258
	Mean	0.294	0.292580317	0.29258	0.29258	0.29288	0.2927	0.29258
	S.D	0.0012469	2.50E−06	2.35E−07	1.72E−07	8.84E−04	1.9409E−04	1.39E−15

Table 3 (continued)

f_n	Statistic	PSO2011	ABC	FPA	GWO	CMA-ES	FFA	NPO
8	Best	246.5546	1.89E+02	17.2851	4.40E−29	0	4.6417	0
	Worst	520.3061	3.16E+02	169.1789	6.74E−26	0	31.1956	0
	Median	402.8341	2.51E+02	44.2972	1.65E−29	0	18.4768	0
	Mean	389.7976	2.48E+02	53.1959	6.38E−27	0	17.9621	0
	S.D	72.833	3.47E+01	31.1372	1.39E−26	0	6.8025	0
9	Best	1.3389	0.1153898	0.028022	1.85E−04	0.00142	0.00409	6.00E−08
	Worst	7.7498	0.294229888	0.56196	0.001033197	0.00771	0.096157	8.24E−05
	Median	5.4099	0.206504048	0.13346	4.47E−04	0.00622	0.015618	9.58E−06
	Mean	6.9606	0.195930447	0.17695	4.47E−04	0.00514	0.025424	1.74E−05
	S.D	0.6477	0.055478056	0.12362	2.11E−04	0.00201	0.023129	2.04E−05
10	Best	57.2433	30.76537	10.4292	2.71E−19	0.038303	0.0119	0
	Worst	80.7359	52.86893	21.4516	4.46E−17	0.094114	0.0370	0
	Median	68.9134	40.23089	15.5261	7.06E−18	0.061089	0.0214	0
	Mean	68.8236	40.79692	15.7964	1.18E−17	0.063286	0.0230	0
	S.D	4.6243	6.529544	3.1328	1.34E−17	0.15721	0.0064	0
11	Best	1.7985	1.30E−14	42.2929	8.283E−41	0.27895	0.1237	0
	Worst	4.3046	1.02E−13	57.9641	4.068E−39	50.1202	0.4179	0
	Median	3.1041	4.37E−14	44.5161	4.53E−40	0.36643	0.2351	0
	Mean	2.5986	4.52E−14	49.8548	9.00E−40	3.2888	0.2377	0
	S.D	1.0657	2.25E−14	6.819	1.02E−39	9.9447	0.0654	0
12	Best	1.2945	0	2.66E−52	0	1.021E−42	0.0012864	0
	Worst	5.1309	0	2.34E−39	0	2.541E−39	0.0053216	0
	Median	3.8818	0	6.97E−46	0	1.108E−41	0.0027475	0
	Mean	2.7707	0	8.28E−41	0	8.048E−40	0.0030036	0
	S.D	1.0831	0	4.26E−40	0	5.904E−40	0.001052	0
13	Best	672	0	26	0	0	0	0
	Worst	881	0	70	0	0	0	0
	Median	799.5	0	44.5	0	0	0	0
	Mean	783.9333	0	47.5333	0	0	0	0
	S.D	57.3597	0	12.1335	0	0	0	0
14	Best	0	0	0	0	0	0	0
	Worst	1	0	0	0	3	0	0
	Median	0	0	0	0	0	0	0
	Mean	0.066667	0	0	0	0.26667	0	0
	S.D	0.25371	0	0	0	0.94443	0	0

Table 3 (continued)

f_n	Statistic	PSO2011	ABC	FPA	GWO	CMA-ES	FFA	NPO
15	Best	3195.407	2.79E−16	11.2992	5.14E−30	2.2719e−06	0.0077476	0
	Worst	5905.7553	2.67E−16	111.8415	8.86E−28	1.3746e−05	1.7702	0
	Median	4865.1481	2.69E−16	47.7022	4.52E−29	5.2988e−06	0.072834	0
	Mean	4811.7969	2.72E−16	53.6457	1.21E−28	6.7746e−06	0.21006	0
	S.D	588.3249	8.51E−12	27.5951	1.76E−28	3.4405e−06	0.34752	0
16	Best	1.2293	0.020580523	0	0	0.0012	2.86E−05	0
	Worst	4.5856	0.154424436	0	0	0.0023	0.00058994	0
	Median	3.2952	0.063475712	0	0	0.0017	0.00024973	0
	Mean	3.1853	0.069228159	0	0	0.0017	0.00027632	0
	S.D	0.92982	0.036675834	0	0	2.528E−04	0.00014538	0
17	Best	0.0058429	0	0	0	0	1.08E−07	0
	Worst	0.22859	0	0	0	0.1462661	4.41E−06	0
	Median	0.058131	0	0	0	0.0717123	1.41E−06	0
	Mean	0.075826	0	0	0	0.0074451	1.60E−06	0
	S.D	0.059017	0	0	0	0.0396471	1.16E−06	0
18	Best	0.0019593	0	0	0	0	4.41E−08	0
	Worst	0.29095	0	0	0	0.0000147	3.66E−06	0
	Median	0.019745	0	0	0	0.0003361	5.74E−07	0
	Mean	0.037888	0	0	0	0.0001048	8.64E−07	0
	S.D	0.056996	0	0	0	0.0005741	9.09E−07	0
19	Best	− 2.0626	− 2.0626	− 2.0626	− 2.0626	− 2.0626	− 2.0626	− 2.0626
	Worst	− 2.0623	− 2.0626	− 2.0623	− 2.0623	− 2.0623	− 2.0623	− 2.0623
	Median	− 2.0626	− 2.0626	− 2.0626	− 2.0626	− 2.0626	− 2.0626	− 2.0626
	Mean	− 2.0626	− 2.0626	− 2.0626	− 2.0626	− 2.0626	− 2.0626	− 2.0626
	S.D	0.000057938	9.03E−16	1.35E−10	2.70E−09	5.03E−07	1.02E−03	5.014E−16
20	Best	0.36776	4.44E−16	0	0	0.0002661	1.55E−08	0
	Worst	4.2538	0.0173688	0	0	0.0243122	1.85E−06	0
	Median	1.3513	8.16E−15	0	0	0.0043354	4.32E−07	0
	Mean	1.6518	5.79E−04	0	0	0.0060214	5.55E−07	0
	S.D	1.0598	0.003171094	0	0	0.677778	4.58E−07	0
21	Best	3	3	3	3	3	3	3
	Worst	3.0031	3	3	3	3	3	3.6273
	Median	3.0002	3	3	3	3	3	3.0105
	Mean	3.0004	3	3	3	3	3	3.0185
	S.D	6.05E−04	1.79E−15	1.02E−15	1.31E−05	1.4871E−03	9.72E−09	0.0023689

Table 3 (continued)

f_n	Statistic	PSO2011	ABC	FPA	GWO	CMA-ES	FFA	NPO
22	Best	− 3.8627	− 3.2618	− 3.8628	− 3.8628	− 3.8628	− 3.8628	− 3.8628
	Worst	− 3.859	− 3.0021	− 3.8628	− 3.854900634	− 3.8628	− 3.8628	− 0.9528
	Median	− 3.8622	− 3.0428	− 3.8628	− 3.8628	− 3.8628	− 3.8628	− 3.7741
	Mean	− 3.8619	− 3.8628	− 3.8628	− 3.8624	− 3.8628	− 3.8628	− 3.7241
	S.D	0.00095848	4.723E−10	3.16E−15	0.002870273	2.373E−16	4.72E−10	0.72518
23	Best	− 3.0425	− 3.0425	− 3.0425	− 3.042457538	− 3.0425	− 3.0425	− 3.0425
	Worst	− 2.9144	− 2.98	− 2.981	− 2.917750648	− 2.9810	− 2.98	− 2.93984
	Median	− 2.9855	− 3.0425	− 3.0425	− 3.042454284	− 3.0425	− 3.0425	− 3.03345
	Mean	− 2.9851	− 3.028	− 3.0404	− 3.01605053	− 3.0179	− 3.028	− 3.01605
	S.D	0.035303	0.026583	0.011219	0.037190414	0.0312	0.026583	0.045706
24	Best	8.8242	3.82E−16	1.3124	0.0065555	3.278E−07	1.20E−05	3.7177e−03
	Worst	14.8547	1.18E−15	30.5618	0.063406	1.115E−05	5.41E−05	0.002411
	Median	11.8014	7.86E−16	5.7516	0.020171	5.296E−07	2.20E−05	0.001012
	Mean	11.8403	7.86E−16	7.6587	0.024756	6.237E−07	2.55E−05	0.001024
	S.D	4.0177	1.61E−16	6.5852	0.013532	2.718E−07	1.10E−05	4.2575e−04
25	Best	2.1049	5.0164E−16	29.6125	1.41E−05	4.183E−06	0.0001365	1.95734
	Worst	5.7544	1.4295E−15	51.5251	0.80499	2.073E−05	0.00053382	2.1618
	Median	3.3085	8.7177E−16	42.7366	0.29864	8.639E−06	0.0002889	2.9694
	Mean	3.5478	8.60768E−16	44.4933	0.29976	1.054E−05	0.00028766	2.7593
	S.D	7.0145	1.79894E−16	21.8372	0.19359	5.430E−06	9.07E−05	0.2139
26	Best	7.67E−02	1.31E−08	7.91E−06	9.38E−06	0	4.63E−07	0.0043
	Worst	3.2113	0.011847851	0.00096267	1.5743	0.14269	0.01093	1.9613
	Median	1.1133	0.000954721	0.00019502	0.00027899	0.076673	0.0005889	0.2505
	Mean	1.2307	0.003526435	0.00023581	0.05431	0.094781	0.0038414	0.5342
	S.D	0.84764	0.001604834	0.00023964	0.2871	0.142695	0.0050794	0.6202
27	Best	0.02734	5.82E−05	0.00019968	0.00012909	2.672E−05	1.98E−08	3.138E−10
	Worst	2.3947	0.005481231	0.010521	0.88183	0.006112	0.0016547	0.0086957
	Median	0.30682	0.000318945	0.0015656	0.0011625	4.906E−04	0.0002464	0.081461
	Mean	0.51315	0.00013892	0.0021292	0.10797	0.0015471	0.00029232	0.0067372
	S.D	0.51074	0.00013892	0.0021506	0.25769	0.001881	0.000372	1.8389
28	Best	− 186.7309	− 186.7309	− 186.7309	− 186.7309	− 186.7309	− 186.7309	− 186.7309
	Worst	− 176.1507	− 186.7309	− 186.7301	− 186.5771	− 179.2478	− 186.7309	− 161.7144
	Median	− 185.4696	− 186.7309	− 186.7308	− 186.7308	− 180.4184	− 186.7309	− 186.544
	Mean	− 184.4844	− 186.7309	− 186.7307	− 186.7215	− 185.3798	− 186.7309	− 184.9683
	S.D	2.7483	8.88E−06	0.00018617	0.031254	3.94785	1.55E−06	0.0015698

Table 3 (continued)

f_n	Statistic	PSO2011	ABC	FPA	GWO	CMA-ES	FFA	NPO
29	Best	3.1901	5.04E−08	3.0672	4.60E−41	0.0049316	0.0028	0
	Worst	5.6217	1.83E−05	9.2612	0.00088014	0.021716	0.0109	0
	Median	4.1841	8.59E−07	6.2124	4.04E−38	0.0092847	0.0075	0
	Mean	4.9583	1.76E−06	6.2209	4.66E−05	0.010317	0.0068	0
	S.D	1.4454	3.35E−06	1.7611	0.00017085	0.0039505	0.0028	0
30	Best	0.0014994	0	0	0	0	6.90E−08	0
	Worst	0.35754	0	0	0	0.08475	7.60E−06	0
	Median	0.050016	0	0	0	0.02134	1.15E−06	0
	Mean	0.10545	0	0	0	0.68741	2.06E−06	0
	S.D	0.11116	0	0	0	0.12458	1.89E−06	0
31	Best	7.1698E−07	4.70E−20	0	1.63E−07	0	2.81E−12	3.98E−10
	Worst	0.00089499	3.53E−17	0	2.66E−06	0	5.20E−09	0.039862
	Median	0.000077554	5.50E−18	0	4.79E−06	0	1.65E−09	0.000117
	Mean	0.00017393	8.57E−18	0	6.43E−06	0	2.20E−09	0.0016352
	S.D	0.00023355	9.23E−18	0	6.59E−06	0	1.56E−09	0.0072483
32	Best	0.39789	0.39788	0.39789	0.39789	0.3979	0.39789	0.3979
	Worst	0.39935	0.39788	0.39789	0.39797	0.3979	0.39789	0.3979
	Median	0.39802	0.39788	0.39789	0.39789	0.3979	0.39789	0.3979
	Mean	0.39816	0.39788	0.39789	0.39789	0.3979	0.39789	0.3979
	S.D	0.00038302	0	1.59E−11	1.58E−05	2.457E−10	1.17E−09	0
33	Best	− 1.8013	− 1.80130341	− 1.8013	− 1.8013	− 1.8013	− 1.8013	− 1.8013
	Worst	− 1.801	− 1.80130341	− 1.8013	− 1.8013	− 1.8013	− 1.8013	− 0.99999
	Median	− 1.8013	− 1.80130341	− 1.8013	− 1.8013	− 1.8013	− 1.8013	− 1.7879
	Mean	− 1.8013	− 1.80130341	− 1.8013	− 1.8013	− 1.8013	− 1.8013	− 1.6392
	S.D	6.79E−05	9.03E−16	9.03E−16	8.82E−07	4.412E−16	6.23E−10	0.29461
34	Best	− 3.9775	− 4.687658179	− 4.6861	− 4.6876	− 4.69346	− 4.6877	− 3.9969
	Worst	− 3.0457	− 4.687658179	− 4.4941	− 3.5992	− 3.99781	− 4.1684	− 1.701
	Median	− 3.4219	− 4.687658179	− 4.6438	− 4.4959	− 4.52137	− 4.5377	− 3.3491
	Mean	− 3.4417	− 4.687658179	− 4.6447	− 4.4262	− 4.0122.	− 4.5629	− 3.2611
	S.D	2.60E−01	2.61E−15	0.039355	0.32476	0.49512	0.11493	0.49051
35	Best	− 5.4708	− 9.660151716	− 8.5604	− 9.3067	− 7.8126	− 9.5515	− 9.2284
	Worst	− 3.8777	− 9.660151716	− 6.9282	− 5.5648	− 6.1748	− 7.138	− 5.7357
	Median	− 4.5549	− 9.660151716	− 7.5294	− 8.0418	− 6.9177	− 9.0418	− 5.8739
	Mean	− 4.5434	− 9.660151716	− 7.5887	− 7.9528	− 6.9336	− 8.8045	− 6.6797
	S.D	0.41908	0	0.41986	0.93142	0.4898	0.6493	0.86726

Table 3 (continued)

f_n	Statistic	PSO2011	ABC	FPA	GWO	CMA-ES	FFA	NPO
36	Best	0.024484	5.68E−14	0	0	0.00104	4.94E−10	0
	Worst	2.151	1.40E−11	0	0	0.03991	2.06E−07	0
	Median	1.076	1.71E−13	0	0	0.00933	5.18E−08	0
	Mean	1.1077	8.83E−13	0	0	0.01094	5.99E−08	0
	S.D	0.55972	2.76E−12	0	0	0.01038	5.08E−08	0

3.3.2 NPO for large-scale problems

To test the robustness of NPO over large-scale problems with different dimensions sizes, several continuous benchmark functions were chosen and used with medium (100), large (500), and very large (2000) number of decision variables (dimensions). The results of the 30 runtimes (best, worst, median, mean and standard deviation) are compared to those of the mentioned metaheuristics. The test functions used in this section are 13 problems. These problems are

$$(f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13}, f_{15}, f_{16}, f_{20}, f_{24}, f_{25}, f_{29}, \text{ and} f_{36}).$$

The main difference between these test functions and the rest is that these functions have dynamic number of variables, thus, these functions can be used with the mentioned above dimensions. The results are presented in Tables 6, 7, and 8 respectively.

The results have clearly showed that NPO has the ability to handle the large-scale problems. NPO has attained the best results in 11 out of 13 test functions, while the other algorithms have failed with most of these tests, especially with the very large problems (i.e., number of dimensions = 2000). The multi-swarm structure of NPO provides stable performance in terms of the scalability, and outperforms the other algorithms from the literature, except for GWO, which attained the second place for most of the problems in the experiment. The exploration part of NPO helps the algorithm to explore a wide area in the search space, and avoid trapping in the local optima.

3.4 Convergence analysis

The convergence curves for several test functions of NPO and the other algorithms are provided in Fig. 6a–f for the first 100 iterations. It is clear that NPO has fast convergence as compared with the other algorithms, because of two reasons. Firstly, the semicircular distribution and the leadership transition change the position of the families (solutions) faster than the other algorithms, in other words, these two operators enhance the local search mechanism and get a new local best solution each iteration. At the same time, when the families are distributed based on semicircular distribution operator, their new positions depend on their leaders (local best solutions), meaning that the families are converging fast enough toward the optimal solution. Secondly, the periodical meeting operator (meeting room approach) increases the exploration ability of NPO by sharing the information between the leaders (local best solutions), which enhances the searching ability of the families when they are looking for new positions.

As a summary of the convergence, each member (family) in NPO has its own responsibility to improve its

Table 4 Summarized comparison results of NPO versus other algorithms

Algorithm	Results								
	U-N/U-S			M-N/M-S			All Tests		
	+	−	=	+	−	=	+	−	=
NPO versus PSO	14	0	1	14	2	5	28	2	6
NPO versus ABC	7	1	7	6	6	9	13	7	16
NPO versus FPA	9	1	5	5	3	13	14	4	18
NPO versus CMA-ES	6	1	8	7	4	10	13	5	18
NPO versus GWO	8	1	6	4	3	14	12	4	20
NPO versus FFA	10	1	4	9	5	7	19	6	11
Total	54	5	31	46	22	58	99	28	89
	60%	5.5%	34.5%	36.5%	17.5%	46%	45.8%	13%	41.2%

position within the clan and helps the clan to find better position within the search space. This is evident in the convergence curves, NPO showed a superior performance as compared to the rest algorithms.

3.5 Discussion

This section discusses the outcomes of NPO, and attempts to answer the following question: 'why is NPO efficient?' In fact, two primary reasons can be outlined in this case, which are: (1) NPO has good exploration and exploitation capabilities, and (2) NPO has a powerful mechanism that balances exploration and exploitation capabilities. The exploration is applied twice, the first time occurs when the leaders are initialized at the first meeting and meet at the periodical meeting, while the second time takes place when the families are searching in the search space, or in precise, the third operator. The exploration of NPO differs from that of other metaheuristics, in which it explores the search space by employing several members of swarms, while other swarm-based algorithms commonly use a specific mechanism between the global best solution and the whole swarm. Moreover, the meeting room approach (MRA), which is proposed in this paper, forces the normal leaders to follow the best leader by using the direction variable Ψ. This variable guides them toward better places, in precise, they may find better positions for their clans. In this paper, two values were employed for the direction variable, $+1$ or -1, because the values of fitness appeared either positive or negative, while in future studies, the researcher may use varied values based on their case studies, if these values do not suit them.

The exploration ability of NPO was optimum when NPO was applied on multimodal test functions, which comprised of 21 test functions. NPO successfully discovered 19 optimal solutions. In addition, NPO proved that it possesses the ability to avoid all local optima and could approach the global optima on most of these tests. The convergent curves showed that NPO had the fastest convergence on multimodal test functions as well.

On the other hand, the exploitation stage consists of two operators: semicircular distribution, and leadership transition. The first operator represents the local search mechanism of NPO, while the other operator exploits the solutions generated by the other two operators. Besides, it is worth to mention that each clan with the second, third, and fourth operators reflects an independent search algorithm, which indicates that search algorithms are embedded in NPO $\#Clans$ (no. of clans). For each iteration (generation) in NPO, the families in each clan search for better places to move to, thus discovering leaders, and the clans can be enhanced internally. Even if those leaders fail to emerge as better than the global best leader, they still represent an enhancement in NPO, thus leading to an enhancement in the searching process, when MRA is applied. Figure 9 illustrates an idea of the processes that take place in both stages and the general block diagram of NPO.

It is obvious that NPO does not contain any controlling parameters, except for structural parameters, i.e., number of clans and number of iterations. Although these parameters do not influence the search behavior of NPO, they do have an impact on the probability of finding the best solutions, or in precise, more families or more clans find the solutions faster in the expense of execution time. This paper had examined five clans and ten families in each clan (50 swarm size in total), in which the performance was found efficient in terms of execution time (Fig. 7).

It is important to note that NPO exhibited exceptional performance with noise test function, especially quartic test function (f_9). To the best of our knowledge, no algorithm in the literature has recorded the performance level achieved in this study. On the other hand, some functions, such as Matyas (f_3) and Stepint (f_{14}), proved to be difficult functions since the flatness of the function did not provide the algorithm any information to channel the search space

Table 5 Wilcoxon signed-rank test

f_n	Versus PSO		Versus ABC		Versus FPA		Versus GWO		Versus CMA-ES		Versus FFA	
	Z	R	Z	R	Z	R	Z	R	Z	R	Z	R
f_1	− **2.705**	**d***	− 4.782	c*	− 4.782	c*	− 2.993	c*	− 4.782	c*	− 4.782	c*
f_2	− **4.782**	**d***	− **2.023**	**d***	0.000	b	0.000	b	− **2.023**	**d***	− **2.828**	**d***
f_3	− **4.782**	**d***	− **4.286**	**d***	− **4.782**	**d***	− **4.782**	**d***	0.000	b	− **4.782**	**d***
f_4	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***
f_5	− **4.782**	**d***	0.000	b	0.000	b	0.000	b	0.000	b	− **4.782**	**d***
f_6	− **4.782**	**d***	− **1.414**	**d**	0.000	b	− **4.264**	**d***	− **1.414**	**d**	0.000	b
f_7	− **4.782**	**d***	− **2.460**	**d***	− 1.414	c	− **4.782**	**d***	− **1.113**	**d**	− **3.197**	**d***
f_8	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	0.000	b	− **4.782**	**d***
f_9	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***
f_{10}	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***
f_{11}	− **4.782**	**d***	− **4.783**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***
f_{12}	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	0.000	b	− **4.782**	**d***	− **4.782**	**d***
f_{13}	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	0.000	b	0.000	b	0.000	b
f_{14}	− **1.414**	**d**	0.000	b	0.000	b	0.000	b	− 1.414	d	0.000	b
f_{15}	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***
f_{16}	− **4.782**	**d***	− **4.782**	**d***	0.000	b	0.000	b	− **4.782**	**d***	− **4.782**	**d***
f_{17}	− **4.782**	**d***	0.000	b	0.000	b	0.000	b	− **4.762**	**d***	− **4.782**	**d***
f_{18}	− **4.782**	**d***	0.000	b	0.000	b	0.000	b	− **4.762**	**d***	− 4.782	c*
f_{19}	− **1.820**	**d**	− 1.841	c*	− 5.201	c*	− 1.841	c	− **1.003**	**d**	− **1.201**	**d**
f_{20}	− **4.782**	**d***	− **4.783**	**d***	0.000	b	0.000	b	− **4.762**	**d***	− **4.782**	**d***
f_{21}	− 4.741	c*	− 4.782	c*	− **4.782**	**d***	− 4.165	c*	− 2.023	c*	− 4.782	c*
f_{22}	− 4.762	c*	− 4.762	c*	− 4.762	c*	− 4.782	c*	− 1.414	c*	− 4.782	c*
f_{23}	− **3.445**	**d***	− **0.086**	**d**	− 3.060	c*	− **2.786**	**d***	− **0.451**	**d**	0.000	b
f_{24}	− **4.782**	**d***	− 4.782	c*	− **4.782**	**d***	0.000	b	− 4.762	c*	− 4.782	c*
f_{25}	− **4.782**	**d***	− 4.782	c*	− **4.782**	**d***	− 3.898	c*	− 4.782	c*	− 4.782	c*
f_{26}	− **1.306**	**d**	− 3.445	c*	− 4.782	c*	− 3.980	c*	− 3.980	c*	− 4.782	c*
f_{27}	− **1.244**	**d**	− 2.651	c*	− 4.762	c*	− **3.939**	**d***	− 4.268	c*	0.000	b
f_{28}	− **1.903**	**d**	− 4.268	c*	− 4.762	c*	− 4.268	c*	− **1.947**	**d***	− 4.703	c*
f_{29}	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***
f_{30}	− **4.782**	**d***	0.000	b	0.000	b	0.000	b	− 4.782	c*	− **4.782**	**d***
f_{31}	− 4.741	c*	− 4.782	c*	− 4.782	c*	− 4.782	c*	− 4.782	c*	− 4.782	c*
f_{32}	− 4.659	c*	− 5.477	c*	− 5.477	c*	− 5.477	c*	− **4.268**	**d***	0.000	b
f_{33}	− **4.782**	**d***	− **4.782**	**d***	− **4.782**	**d***	0.000	b	− 2.023	c*	− **4.782**	**d***
f_{34}	− 1.841	c	− 4.782	c*	− 4.782	c*	− 4.618	c*	− 4.782	c*	− 4.782	c*
f_{35}	− **4.083**	**d***	− 4.782	c*	− 4.350	c*	− 4.762	c*	− 0.652	c	− 4.741	c*
f_{36}	− **4.782**	**d***	− **4.762**	**d***	0.000	b	0.000	b	− **4.782**	**d***	− **4.782**	**d***

Bold values indicate the best results

toward the best solutions. NPO, nonetheless, attained the best solution for these two functions, hence proving its efficacy in solving problems with limited information.

With proven efficiency of NPO, it also has some shortcomings that should be investigated in future studies. One drawback of NPO is its failure in solving several test functions, especially those in the form of multimodal, such as (f_{22},f_{25},$f_{33} - f_{35}$). NPO failed in seeking the best solution for f_{35}, which refers to a multimodal test function in the used number of iterations, where it started rapidly at the beginning of the search, but then the convergence became very slow. Nevertheless, NPO attained a good position near to the optimal solution for this test when number of iterations have been increased. Figure 8 portrays the convergence of NPO for f_{35}.

As mentioned previously, NPO and the other algorithms are swarm-based metaheuristics. Hence, they have been evaluated within the same environment. A time-based comparison, however, showed that NPO reached the optimal solutions for most of the tests within shorter period of time, in comparison to other algorithms. Figure 9 displays the time-based comparison of single run for each test.

Table 6 Results of the metaheuristics over Large-Scale Problems (D = 100)

f_n	Statistic	PSO2011	ABC	FPA	GWO	CMA-ES	FFA	NPO
f_8	Best	1334.055	1254.458	541.7490	7.1317E−08	1779.142	0.9161	0
	Mean	1429.126	1365.845	987.7509	1.9303E−05	2658.035	23.0758	2.148E−12
	S.D	54.767	84.4225	243.5670	3.4657E−05	586.7334	18.5865	0.04781
f_9	Best	664.939	2.0912	2.1086	1.6465E−04	0.0070	0.0113	7.32E−07
	Mean	773.376	2.5277	5.1964	9.3399E−04	0.0142	0.0239	1.39E−05
	S.D	108.421	0.3258	2.7691	4.3166E−04	0.0091	0.0062	1.28E−05
f_{10}	Best	81.7736	86.4876	23.9367	6.0793E−12	3.2603	0.8033	0
	Mean	85.0544	89.1781	27.6855	1.1477E−08	4.0145	3.9197	0
	S.D	2.7092	1.6063	2.4035	3.6298E−08	0.1613	4.8113	0
f_{11}	Best	4.335E+129	0.0049	1.4386E+49	1.3413E−37	82.0537	66.0143	0
	Mean	6.860E+129	0.0110	3.1938E+65	9.9048E−37	98.4765	148.4295	0
	S.D	9.423E+130	0.0097	8.5272E+65	5.4157E−37	7.3200	116.4210	0
f_{12}	Best	152314.109	0.0064	5835.941	1.6252E−64	3.1450	0.040309	0
	Mean	159666.025	0.0117	7814.524	6.4274E63	5.1987	0.044081	0
	S.D	7207.4705	0.0048	1593.364	1.1686E−62	1.3472	0.0032	0
f_{13}	Best	3233	0	513	0	0	0	0
	Mean	3270.6	0	617.33	0	1.5333	0	0
	S.D	48.726	0	56.049	0	1.5976	0	0
f_{15}	Best	72373.335	5.128E−09	2793.84	3.0859E−65	8.8075	1.3207	0
	Mean	7825.9311	6.148E−08	4353.14	3.0494E−63	16.7818	3.2474	0
	S.D	5514.437	1.147E−08	836.445	4.8647E−63	4.9145	1.2411	0
f_{16}	Best	0.120141	0.4537	0	0	0.5028	3.4214E−09	0
	Mean	0.591568	1.5798	0	0	0.7173	1.3468E−04	0
	S.D	0.442324	1.0513	0	0	0.1032	7.5581E−05	0
f_{20}	Best	0.14781	4.151E−06	0	0	0.0021	3.14E−08	0
	Mean	0.23108	0.0024	0	0	0.0478	1.45E−07	0
	S.D	0.08286	0.0050	0	0	0.447	1.86E−07	0
f_{24}	Best	5.533E+08	0.47845	15.3689	0.13323	0.0140	7.434E−05	0.8788
	Mean	7.902E+08	0.44885	32366.04	0.23647	0.0295	9.956E−05	0.9922
	S.D	1.436E+08	0.00474	59623.47	0.06456	0.0121	2.184E−05	0.0578
f_{25}	Best	1.718E+09	1.8745	2.499E+05	5.0365	0.4198	0.00285	8.3533
	Mean	1.864E+09	2.8471	2.057E+06	6.0282	0.5233	0.00344	9.8193
	S.D	1.347E+08	0.0576	2.215E+06	0.4857	0.0247	5.634E−04	0.28
f_{29}	Best	185.5875	0.0822	30.0075	6.0308E−38	0.1713	0.0552	0
	Mean	204.521	0.1556	39.6843	5.5211E−36	0.2485	0.3184	0
	S.D	9.0134	0.0601	5.0509	3.0247E−35	0.0436	0.2329	0
f_{36}	Best	0.03714	13.1517	0	0	0.0022	4.9966E−10	0
	Mean	1.02581	16.6879	0	0	0.0161	1.1010E−08	0
	S.D	0.57207	2.4709	0	0	0.0109	1.1926E−08	0

Figure 9 shows that the NPO has a superior performance in terms of execution time in many test functions, such as $(f_2 - f_5, f_7 - f_{20}, f_{28} - f_{30}, f_{36})$. Meanwhile, the performance of NPO appeared moderate for the other test functions.

3.6 Exploitation and exploration analysis

To analyze the two highly influential factors (exploration and exploitation) of the metaheuristics, five commonly used numerical optimization problems with different modality were employed with 30 dimensions. These test functions are

Table 7 Results of the metaheuristics over large-scale problems (D = 500)

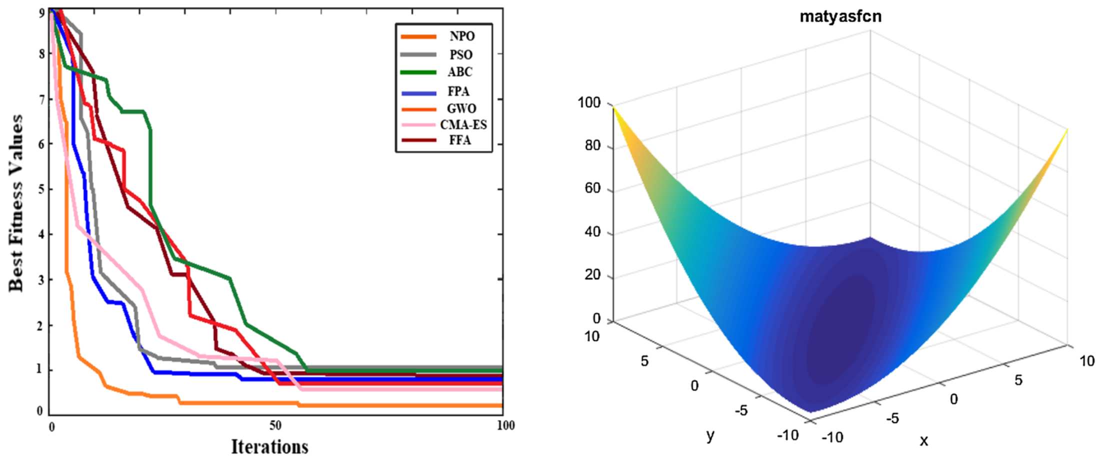
f_n	Statistic	PSO2011	ABC	FPA	GWO	CMA-ES	FFA	NPO
f_8	Best	3.55412	7726.247	13357.241	1337.803	11606.348	115.5513	3.262E−105
	Mean	4.77746	8094.705	4.323E+16	2035.742	19530.154	277.7533	189.1589
	S.D	0.85447	246.1136	1.514E+16	3506.202	4430.996	170.7327	124.3142
f_9	Best	21895.22	9.5551	3945.361	0.0015	14.1004	1.3354	7.32E−07
	Mean	23222.46	13.2630	4576.369	0.0034	19.9608	1.6052	1.39E−05
	S.D	1034.118	4.3713	358.529	0.0012	4.1955	0.2452	1.28E−05
f_{10}	Best	90.0878	98.0309	32.087	32.3180	44.3891	78.0813	0
	Mean	91.6079	98.2855	36.716	46.7162	46.0148	79.1870	0
	S.D	1.07263	0.1744	3.2361	6.1147	0.2847	1.3015	0
f_{11}	Best	–	2.5478	2646.036	2.3985E−18	INF	INF	0
	Mean	–	2.2331	INF	2.2662E+102	INF	INF	0
	S.D	–	0.4784	NaN	1.2387E+103	NaN	NaN	0
f_{12}	Best	15160.206	17730.447	495913.721	3.4875E−31	5424.264	0.2541	0
	Mean	19520.951	34161.687	709919.845	1.4785E−30	6255.894	0.6185	0
	S.D	3517.773	10941.154	10100.475	2.6478E−30	432.9677	0.0561	0
f_{13}	Best	2.076E+07	8	4076	0	5841	16	0
	Mean	4.753E+07	22.842	4471.00	0	6555.0214	17.00	0
	S.D	3.544E+07	24.484	298.707	0	558.6245	1.00	0
f_{15}	Best	2.267E+06	0.00148	126351.566	9.5724E−31	32888.648	1880.0944	0
	Mean	2.318E+06	0.02854	173852.407	2.7055E−30	38574.394	2248.810	0
	S.D	5.125E+04	0.9453	22891.88	1.6336E−30	3363.985	581.7029	0
f_{16}	Best	0.03449	2.7839	0	0	5.7905	1.594E−04	0
	Mean	0.27365	3.4535	0	0	6.0433	0.0011	0
	S.D	0.14307	0.5038	0	0	0.1570	5.403E−04	0
f_{20}	Best	0.05649	199.0954	0	0	0.0060	1.8316e−08	0
	Mean	0.17311	278.6552	0	0	0.1226	4.8136e−06	0
	S.D	0.09765	61.7961	0	0	0.0918	5.6278e−06	0
f_{24}	Best	5.483E+09	4.6874	199170.663	0.70820	12.0199	8.4167	0.8788
	Mean	5.734E+09	8.9965	12613.77	0.75102	13.5082	13.3083	0.9922
	S.D	1.877E+08	2.9984	14376.24	0.02623	0.8086	3.6139	0.0578
f_{25}	Best	9.850E+09	39.9985	1.097E+07	44.4242	1765.485	160.9658	8.3533
	Mean	1.104E+10	42.8541	3.525E+07	45.7044	12082.965	187.347	9.8193
	S.D	7.341E+08	1.3147	2.387E+07	0.58574	8844.015	17.9665	0.28
f_{29}	Best	1118.486	104.5457	281.22	1.3988E−18	168.4668	20.0697	0
	Mean	1136.983	112.5482	307.91	5.6327E−05	191.7240	32.3101	0
	S.D	22.4322	5.4539	16.73	2.8410E−04	15.9773	5.4495	0
f_{36}	Best	0.006272	1434.661	0	0	4.477E−04	5.2875e−08	0
	Mean	0.024489	1487.747	0	0	0.0221	0.1159	0
	S.D	0.021632	54.4633	0	0	0.0183	0.2828	0

(f_{11},f_{12},f_{16},f_{24},f_{36}). This section focuses mainly on calculating diversity in swarm during iterations instead of running the algorithm over certain number of independent runs and averaging the results. Accordingly, we executed algorithm once, as our preliminary experiments also evidenced insignificant difference in results over multiple runs.

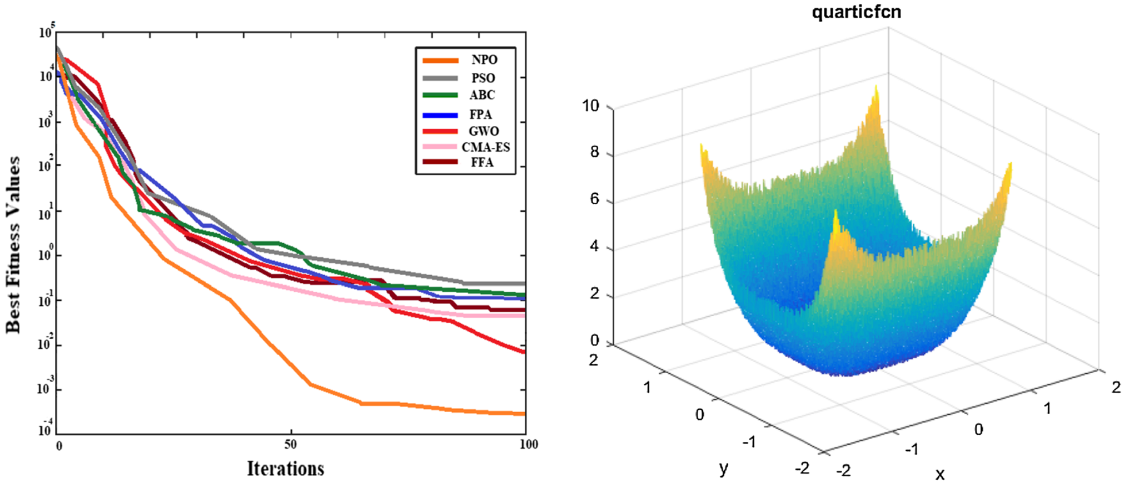
The exploitation has been calculated by using a counter in three different parts, first part when all families are initialized around their leader's tent. While the second and third parts when each time the Leadership operator is executed. On the other hand, the exploration has been calculated by using the same method, in three different parts as well. First part when the leaders are initialized in

Table 8 Results of the metaheuristics over large-scale problems (D = 2000)

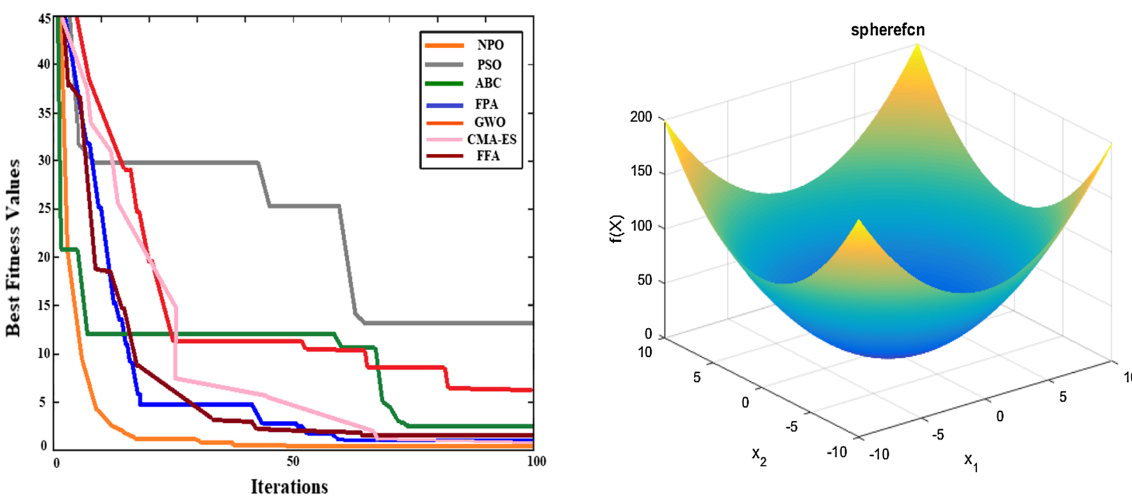
f_n	Statistic	PSO2011	ABC	FPA	GWO	CMA-ES	FFA	NPO
f_8	Best	1.338E+24	35572.845	5.8022E+23	11272.4784	28625.441	272221.21	2.245E−50
	Mean	1.562E+24	35821.841	1.5932E+24	13308.7124	43932.854	415776.54	2489.941
	S.D	1.963E+23	227.2855	6.4922E+23	2008.487	994.458	68451.24	1389.503
f_9	Best	3.956E+05	12.6052	2383	0.0037	713.7286	5800.845	7.32E−07
	Mean	4.086E+05	16.1874	4665	0.0083	873.6089	6941.251	1.39E−05
	S.D	1.167E+05	2.5176	1393	0.0024	120.2117	506.93	1.28E−05
f_{10}	Best	96.0583	99.6208	37.7266	75.747423	99.1100	97.8796	0
	Mean	96.8142	99.6793	43.4886	81.18971	99.8135	98.3709	0
	S.D	0.7548	0.0413	3.0758	3.262956	0.2618	0.2807	0
f_{11}	Best	–	22.1478	INF	7.1622E+03	INF	INF	0
	Mean	–	25.9658	INF	7.4488E+03	INF	INF	0
	S.D	–	1.4414	NaN	1.7400E+02	NaN	NaN	0
f_{12}	Best	3.7959E+06	3.0568E+06	23426	2.7822E−19	55388.645	4.3745 + 07	0
	Mean	3.8229E+06	3.1489E+06	33365	5.6527E−19	57807.158	4.3899 + 07	0
	S.D	4.4358E+04	9.1274E+04	48586	2.2700E−19	1940.947	1.0478 + 04	0
f_{13}	Best	3798199	142.2355	240181	0	53269	424709	0
	Mean	3.815E+06	162.2411	327451	0.0333	59625.458	448213	0
	S.D	18168.498	56.5468	482315	0.1825	3862.648	15294	0
f_{15}	Best	3.7712E+07	1.1104	265134	7.0539E−19	4.213E+05	3888146	0
	Mean	3.8317E+07	1.3347	322812	4.5726E−18	5.812E+05	4221967	0
	S.D	5.1128E+05	0.0114	384351	9.7734E−18	20511.648	1388215	0
f_{16}	Best	0.03245	2.3939	0	0	9.6756	1.1085E−04	0
	Mean	0.26227	3.0277	0	0	9.9159	9.8684E−04	0
	S.D	0.20556	0.4295	0	0	0.1928	5.2224E−04	0
f_{20}	Best	0.17971	2.741E+04	0	0	0.0287	3.8677E−07	0
	Mean	0.26673	2.818E+04	0	0	0.1017	0.0018	0
	S.D	0.09525	483.4293	0	0	0.0601	0.0100	0
f_{24}	Best	2.7797E+10	2.8421	4794010.6	1.1701	2.398E+05	1.634E+07	1.1679
	Mean	2.4371E+10	2.9982	1568531.7	1.1704	7.275E+05	2.364E+07	1.1767
	S.D	7.3089E+10	0.0854	8206735.6	0.0071	3.461E+05	5.615E+06	0.0030
f_{25}	Best	5.983E+10	201.948	4.3962E+07	194.3941	1.129E+07	2.181E+07	197.705
	Mean	5.049E+10	206.665	1.7392E+08	195.402	1.301E+07	2.796E+08	197.712
	S.D	8.779E+08	0.68451	1.0067E+08	0.514705	2.325E+06	2.810E+07	0.0281
f_{29}	Best	4665.602	2695.774	1357	1.3989E−11	697.0373	1558	0
	Mean	4718.114	2774.461	1422	2.3068E−04	726.0756	1604	0
	S.D	50.057	67.6205	76.7146	4.7534E−04	23.8696	25.48	0
f_{36}	Best	3.7212E−04	20592.204	0	0	0.0038	7.9327E−07	0
	Mean	0.3368912	21365.648	0	0	0.0154	0.0029	0
	S.D	0.03807503	489.8769	0	0	0.0107	0.0156	0



A Convergence curve for f_4

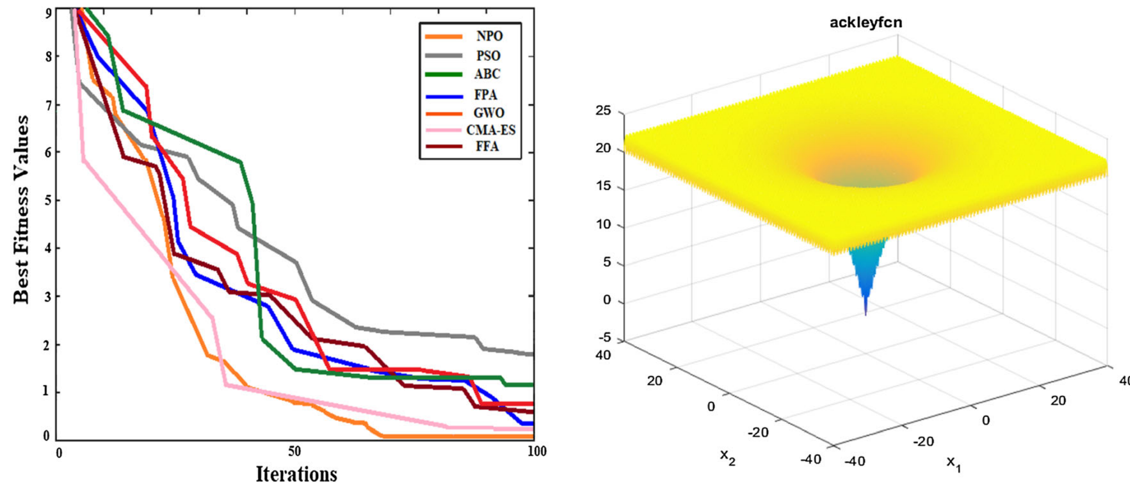


B Convergence curve for f_9

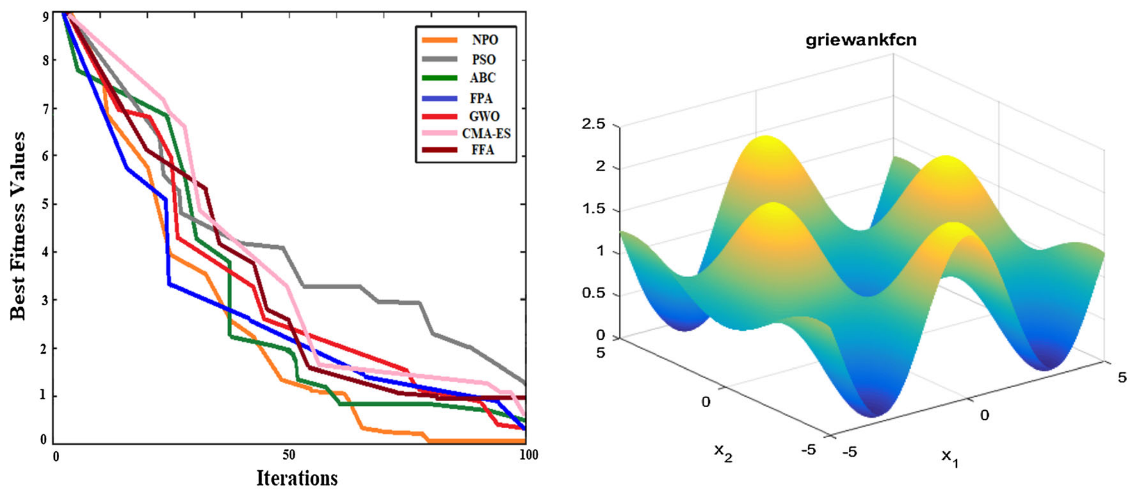


C Convergence curve for f_{12}

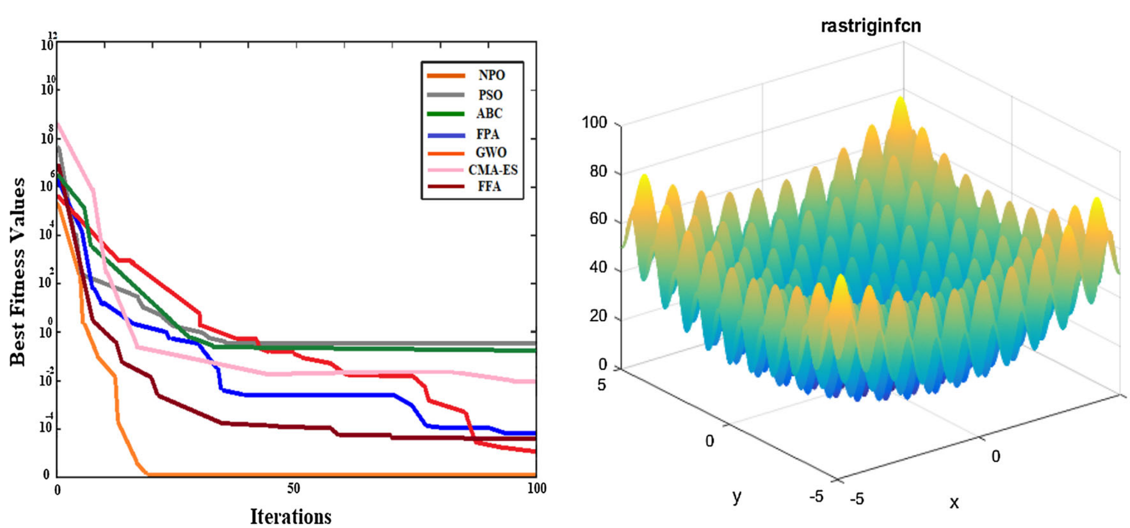
Fig. 6 **a–f** Convergence curves for functions (f_4,f_{12},f_9,f_{16},f_{20},f_{36})



D Convergence curve for f_{16}



E Convergence curve for f_{20}



F Convergence curve for f_{36}

Fig. 6 continued

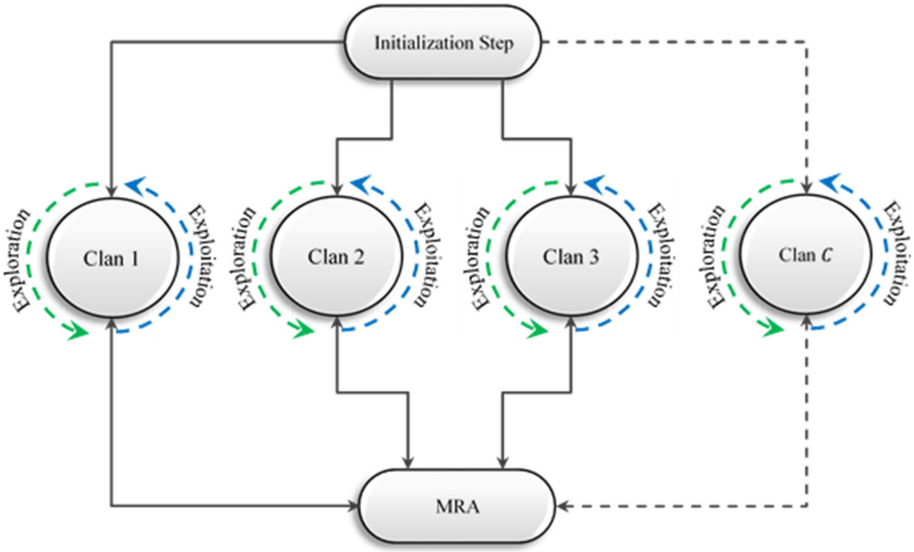


Fig. 7 Exploration and exploitation of NPO

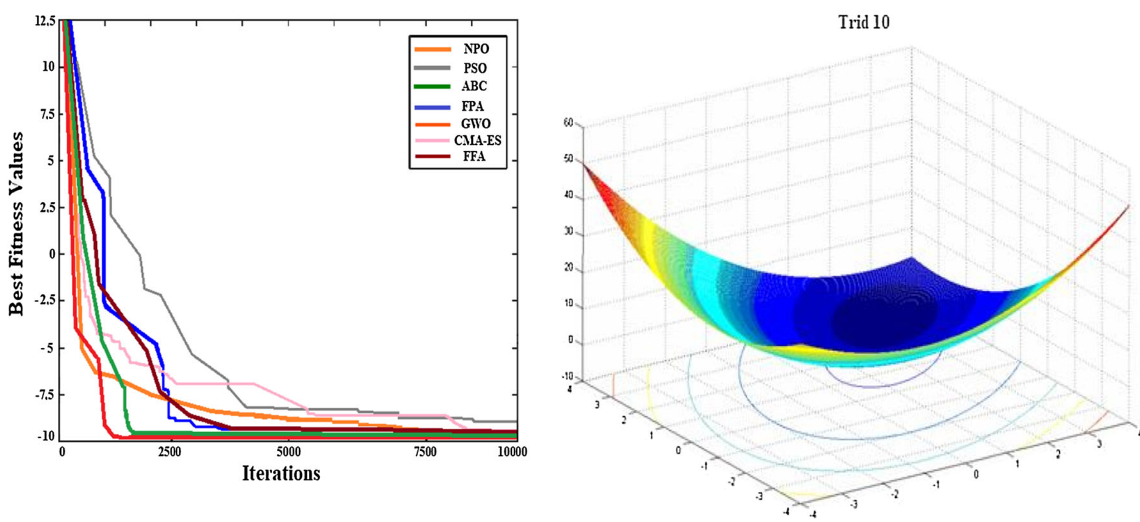


Fig. 8 Convergence curve for f_{35}

the initial meeting, while the second part during the families searching. Final part is when any leader updates his own position inside the MRA. Table presents the best objective function value found by NPO and the other metaheuristics, exploration, and exploitation.

From Table 9, there is a dynamic problem nature-related difference between the exploration and exploitation features of NPO. The flexibility of this difference could be attributed to two reasons: the first reason is the enhancing effect of MRA on the searching process through guiding the normal Sheikhs toward better positions once established; the second reason is related to the process of checking for any family with a better fitness than the Sheikh of the clan (Step 10 in the pseudocode). This condition controls the algorithm and decides whether a local search (leadership transition and semicircular distribution—or exploitation) or a global search (families searching—or exploration) is needed. Unlike the other algorithms, the exploitation and exploration functions are simultaneously executed in the NPO using different governing equations.

4 Conclusion

Hard optimization problems are roughly defined as problems that are difficult to find the optimum problem solution using any deterministic method within a ''reasonable'' time frame. These problems are satisfactorily solved using metaheuristics. Metaheuristic is algorithms which have the capability of solving a wide range of hard optimization problems without necessarily adapting to each problem. There are several issues faced by the metaheuristics, and

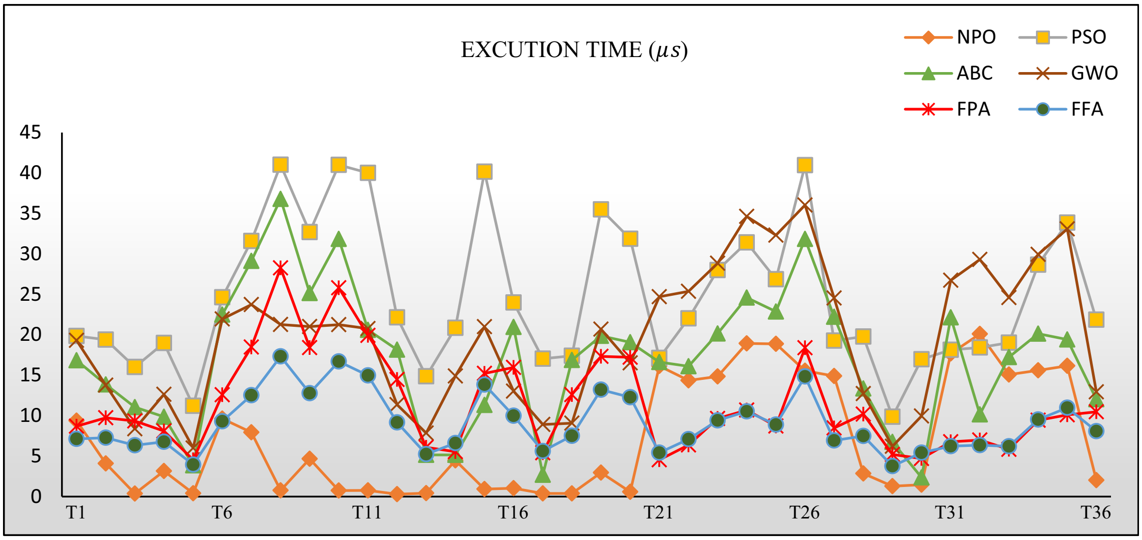


Fig. 9 Time-based comparison between all metaheuristics

Table 9 Results of exploration and exploitation

f_n	Measurements	PSO	ABC	FPA	GWO	FA	NPO
f_{11}	Solution	1.7985	1.30E−14	47493.2929	8.28E−41	0.27895	0
	Exploration	35%	58%	85%	32%	83%	22%
	Exploitation	65%	42%	15%	68%	17%	78%
f_{12}	Solution	1.2945	0	2.66E−52	0	0.0012864	0
	Exploration	33%	59%	63%	68%	88%	46%
	Exploitation	67%	41%	37%	32%	12%	54%
f_{16}	Solution	1.2293	0.020580523	0	0	2.86E−05	0
	Exploration	36%	61%	60%	45%	92%	78%
	Exploitation	67%	39%	40%	55%	08%	22%
f_{24}	Solution	8.8242	3.82E−16	1.3124	0.0065555	1.20E−05	0.008915
	Exploration	40%	58%	59%	71%	86%	87%
	Exploitation	60%	42%	41%	39%	14%	13%
f_{36}	Solution	0.024484	5.68E−14	0	0	4.94E−10	0
	Exploration	56%	72%	74%	76%	82%	71%
	Exploitation	44%	28%	26%	24%	18%	29%

one of such issues is balancing the global search ability—or exploration—with local search ability—or exploitation. Another problem of metaheuristics is their dependence and use of some control parameters whose values significantly controls algorithmic searching process. Hence, there is a need to tune these parameters for a better algorithmic performance.

The major contribution of this study was the design and implementation of a novel parameter-free multi-swarm nature-inspired metaheuristic for unconstrained (normal and large-scale) optimization problems. The proposed Nomadic People Optimizer (NPO) depends on a new multi-swarm approach and this is another contribution of this study. NPO is inspired by the movement of nomads when searching for the sources of food in the desert.

The Meeting Room Approach (MRA) is proposed for two reasons. First, it represents the communication way between the clans, while the second reason is to balance between the exploration and the exploitation. MRA reflects a multi-swarm cooperative scheme that is inspired from the communication behavior of groups of peoples. The MRA applies novel sharing of information between the leaders of the swarms, where the leaders represent the local best solutions. Additionally, NPO does not require any controlling parameter as it requires only two structural parameters which are the number of clans or leaders, and the number of families.

A set of test functions were combined in this paper. As such, the employed set consisted of 36 benchmark functions, which were divided into four groups (U-N, U-S, M-N, and M-S). These test functions can be utilized to evaluate the metaheuristics in terms of exploitation and exploration, apart from avoiding the trapping in the local optima. The results showed that the NPO exerted a superior performance, in comparison to other six well-known algorithms (PSO, ABC, GWO, FFA, CMA-ES, and FPA). In quantitative scale, the proposed algorithm demonstrated 45.8% of optimal solution from the benchmark tests. Additionally, the Wilcoxon signed-rank test, which is a statistical ranking method, showed that the NPO has a significant difference to other metaheuristics. The convergence analysis confirmed that NPO possesses the ability to seek the optimal solution in a rapid manner, mainly due to the proposed local and global searches, aside from MRA that controls the balance between them. On the other hand, NPO has been evaluated based on 13 large-scale problems, it had a superior performance against the other metaheuristics, meaning that NPO has the ability to handle the problems with very large number of decision variables.

For future studies, the proposed NPO could be enhanced to solve issues related to constraint optimization. In addition, it can be used to solve some optimization problems related to machine learning, such as training the artificial neural network (i.e., tuning the weights of the neural networks), and identifying the most relevant features in the classification/clustering problems (i.e., feature selection problem). Two more versions have been projected for development, which are binary and multiobjective NPOs.

Acknowledgements This research is funded by UMP PGRS170338: Analysis System based on Technological YouTube Channels Reviews, and UMP RDU180367 Grant: Enhance Kidney Algorithm for IOT Combinatorial Testing Problem.

Compliance with ethical standards

Conflict of interest There is no conflict of interest in publishing this paper.

References

1. Engelbrecht AP (2007) Computational intelligence. Wiley, New York
2. Lynn N, Suganthan PN (2017) Ensemble particle swarm optimizer. Appl Soft Comput 55:533–548. https://doi.org/10.1016/j.asoc.2017.02.007
3. Dorigo M, Stützle T (2019) Ant colony optimization: overview and recent advances. In: International series in operations research and management science
4. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
5. Mousavirad SJ, Ebrahimpour-Komleh H (2017) Human mental search: a new population-based metaheuristic optimization algorithm. Appl Intell 47:850–887. https://doi.org/10.1007/s10489-017-0903-6
6. Beyer HG, Finck S, Breuer T (2014) Evolution on trees: on the design of an evolution strategy for scenario-based multi-period portfolio optimization under transaction costs. Swarm Evol Comput 17:74–87. https://doi.org/10.1016/j.swevo.2014.03.002
7. Boussaïd I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. Inf Sci 237:82–117
8. Heidari AA, Pahlavani P (2017) An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. Appl Soft Comput J 60:115–134. https://doi.org/10.1016/j.asoc.2017.06.044
9. Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: from natural to artificial systems (No. 1). Oxford University Press
10. Yaseen Z, Mohtar WHMW, Ameen AMS et al (2019) Implementation of univariate paradigm for streamflow simulation using hybrid data-driven model: case study in tropical region. IEEE Access 7:7447–74481
11. Maroufpoor S, Maroufpoor E, Bozorg-Haddad O et al (2019) Soil moisture simulation using hybrid artificial intelligent model: Hybridization of adaptive neuro fuzzy inference system with grey wolf optimizer algorithm. J Hydrol 575:544–556. https://doi.org/10.1016/j.jhydrol.2019.05.045
12. Naganna S, Deka P, Ghorbani M et al (2019) Dew point temperature estimation: application of artificial intelligence model integrated with nature-inspired optimization algorithms. Water. https://doi.org/10.3390/w11040742
13. Yaseen Z, Ebtehaj I, Kim S et al (2019) Novel hybrid data-intelligence model for forecasting monthly rainfall with uncertainty analysis. Water 11:502. https://doi.org/10.3390/w11030502
14. Yaseen ZM, Ehteram M, Hossain MS et al (2019) A novel hybrid evolutionary data-intelligence algorithm for irrigation and power production management: application to multi-purpose reservoir systems. Sustain. https://doi.org/10.3390/su11071953
15. Yaseen ZM, Tran MT, Kim S et al (2018) Shear strength prediction of steel fiber reinforced concrete beam using hybrid intelligence models: a new approach. Eng Struct 177:244–255. https://doi.org/10.1016/j.engstruct.2018.09.074
16. Tao H, Diop L, Bodian A et al (2018) Reference evapotranspiration prediction using hybridized fuzzy model with firefly algorithm: regional case study in Burkina Faso. Agric Water Manag 208:140–151
17. Yaseen Z, Ehteram M, Sharafati A et al (2018) The integration of nature-inspired algorithms with least square support vector regression models: application to modeling river dissolved oxygen concentration. Water 10:1124
18. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010
19. Xiong N, Molina D, Ortiz ML, Herrera F (2015) A walk into metaheuristics for engineering optimization: principles, methods and recent trends. Int J Comput Intell Syst 8:606–636. https://doi.org/10.1080/18756891.2015.1046324
20. Ghorbani MA, Deo RC, Karimi V et al (2017) Implementation of a hybrid MLP-FFA model for water level prediction of Lake Egirdir, Turkey. Stoch Environ Res Risk Assess 32:1683–1697
21. Al Sudani ZA, Salih SQ, Yaseen ZM et al (2019) Development of multivariate adaptive regression spline integrated with differential evolution model for streamflow simulation. J Hydrol 573:1–15
22. Lalbakhsh A, Afzal MU, Esselle KP (2017) Multiobjective particle swarm optimization to design a time-delay equalizer metasurface for an electromagnetic band-gap resonator antenna. IEEE

Antennas Wirel Propag Lett 16:912–915. https://doi.org/10.1109/LAWP.2016.2614498
23. Roberge V, Tarbouchi M, Okou F (2014) Strategies to accelerate harmonic minimization in multilevel inverters using a parallel genetic algorithm on graphical processing unit. IEEE Trans Power Electron 29:5087–5090. https://doi.org/10.1109/TPEL.2014.2311737
24. Gao W-F, Huang L-L, Liu S-Y, Dai C (2015) Artificial bee colony algorithm based on information learning. IEEE Trans Cybern 45:2827–2839
25. Lalbakhsh A, Afzal MU, Esselle K (2016) Simulation-driven particle swarm optimization of spatial phase shifters. In: Proceedings of the 2016 18th international conference on electromagnetics in advanced applications, ICEAA 2016. pp 428–430
26. Lalbakhsh P, Zaeri B, Lalbakhsh A (2013) An improved model of ant colony optimization using a novel pheromone update strategy. IEICE Trans Inf Syst 96:2309–2318. https://doi.org/10.1587/transinf.E96.D.2309
27. Al-Musawi AA, Alwanas AAH, Salih SQ et al (2019) Shear strength of SFRCB without stirrups simulation: implementation of hybrid artificial intelligence model. Eng Comput. https://doi.org/10.1007/s00366-018-0681-8
28. Yaseen ZM, Afan HA, Tran MT (2018) Beam-column joint shear prediction using hybridized deep learning neural network with genetic algorithm. In: IOP conference series: earth and environmental science
29. Hansen N, Müller SD, Koumoutsakos P (2003) Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol Comput 11:1–18. https://doi.org/10.1162/106365603321828970
30. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359. https://doi.org/10.1023/A:1008202821328
31. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3:82–102. https://doi.org/10.1109/4235.771163
32. Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12:702–713. https://doi.org/10.1109/TEVC.2008.919004
33. Lim CP, Jain LC (2009) Advances in swarm intelligence. Stud Comput Intell 248:1–7
34. Jadon SS, Bansal JC, Tiwari R, Sharma H (2015) Accelerating artificial bee colony algorithm with adaptive local search. Memet Comput 7:215–230. https://doi.org/10.1007/s12293-015-0158-x
35. Espitia HE, Sofrony JI (2018) Statistical analysis for vortex particle swarm optimization. Appl Soft Comput J. https://doi.org/10.1016/j.asoc.2018.03.002
36. Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization. Swarm Intell 1:33–57. https://doi.org/10.1007/s11721-007-0002-0
37. Yang XS, Deb S (2009) Cuckoo search via lévy flights. In: 2009 world congress on nature and biologically inspired computing (NaBIC), pp 210–214
38. Yang XS (2009) Firefly algorithms for multimodal optimization. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 5792 LNCS:169–178. https://doi.org/10.1007/978-3-642-04944-6_14
39. Yang XS (2010) Firefly algorithm, Levy flights and global optimization. Res Dev Intell Syst. https://doi.org/10.1007/978-1-84882-983-1
40. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Tech Rep TR06, Erciyes Univ 10. https://doi.org/citeulike-article-id:6592152
41. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
42. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
43. Yaseen ZM, Ebtehaj I, Bonakdari H et al (2017) Novel approach for streamflow forecasting using a hybrid ANFIS-FFA model. J Hydrol 554:263–276. https://doi.org/10.1016/j.jhydrol.2017.09.007
44. Ghorbani MA, Deo RC, Yaseen ZM et al (2017) Pan evaporation prediction using a hybrid multilayer perceptron-firefly algorithm (MLP-FFA) model: case study in North Iran. Theor Appl Climatol 133:1119–1131
45. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Des 43:303–315. https://doi.org/10.1016/j.cad.2010.12.015
46. Kumar M, Kulkarni AJ, Satapathy SC (2017) Socio evolution & learning optimization algorithm: a socio-inspired optimization methodology. Future Gener Comput, Syst
47. Kuo HC, Lin CH (2013) Cultural evolution algorithm for global optimizations and its applications. J Appl Res Technol 11:510–522. https://doi.org/10.1016/S1665-6423(13)71558-X
48. Buckham BJ, Lambert C (1999) Simulated annealing applications. Mech Eng 1–16
49. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110–111:151–166. https://doi.org/10.1016/j.compstruc.2012.07.010
50. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput J 13:2592–2612. https://doi.org/10.1016/j.asoc.2012.11.026
51. Nematollahi AF, Rahiminejad A, Vahidi B (2017) A novel physical based meta-heuristic optimization method known as lightning attachment procedure optimization. Appl Soft Comput J 59:596–621. https://doi.org/10.1016/j.asoc.2017.06.033
52. Huang G (2017) Artificial memory optimization. Appl Soft Comput J 61:497–526. https://doi.org/10.1016/j.asoc.2017.08.021
53. Yang XS, Deb S, Zhao Y-X et al (2017) Swarm intelligence: past, present and future. Soft Comput 22:5923–5933
54. Birattari M, Paquete L, St T, Varrentrapp K (2001) Classification of metaheuristics and design of experiments for the analysis of components. Tech. Rep. AIDA-01-05
55. Črepinšek M, Liu S-H, Mernik M (2013) Exploration and exploitation in evolutionary algorithms: a survey. ACM Comput Surv 45(3):35. https://doi.org/10.1145/2480741.2480752
56. Silberholz J, Golden B (2010) Comparison of metaheuristics. In: Handbook of metaheuristics, Springer, Boston, MA, pp 625–640
57. Kashif H, Salleh N, Cheng S, Shi Y (2018) On the exploration and exploitation in popular swarm-based metaheuristic algorithms. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3592-0
58. Akay B, Karaboga D (2012) Artificial bee colony algorithm for large-scale problems and engineering design optimization. J Intell Manuf. https://doi.org/10.1007/s10845-010-0393-4
59. Taha AM, Chen S-D, Mustapha A (2015) Bat algorithm based hybrid filter-wrapper approach. Adv Oper Res. https://doi.org/10.1155/2015/961494
60. Taha AM, Chen S-D, Mustapha A (2015) Natural extensions: bat algorithm with memory. J Theor Appl Inf Technol 79:1–9
61. Ahmed HA, Zolkipli MF, Ahmad M (2018) A novel efficient substitution-box design based on firefly algorithm and discrete

chaotic map. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3557-3
62. Alzaidi AA, Ahmad M, Ahmed HS, Al Solami E (2018) Sine–cosine optimization-based bijective substitution-boxes construction using enhanced dynamics of chaotic map. Complexity. https://doi.org/10.1155/2018/9389065
63. Clerc M (2011) Standard PSO 2011 (SPSO). In: Part. Swarm Cent. https://www.particleswarm.info/Programs.html
64. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39:459–471. https://doi.org/10.1007/s10898-007-9149-x
65. Yang XS (2012) Flower pollination algorithm for global optimization. In: International conference on unconventional computing and natural computation. Springer, Berlin, pp 240–249
66. Jamil M, Yang XS, Zepernick HJD (2013) Test functions for global optimization: a comprehensive survey. In: Swarm intelligence and bio-inspired computation, Elsevier, pp 193–222
67. Qu BY, Liang JJ, Wang ZY et al (2016) Novel benchmark functions for continuous multimodal optimization with comparative results. Swarm Evol Comput 26:23–34. https://doi.org/10.1016/j.swevo.2015.07.003
68. Andrei N (2008) An unconstrained optimization test functions collection. Adv Model Optim 10:147–161