On Improvements of the Human Mental Search Algorithm for Global Optimisation

Seyed Jalaleddin Mousavirad
Nature-inspired Computational Intelligence Lab
Ontario Tech University
Oshawa, Canada

Gerald Schaefer
Department of Computer Science
Loughborough University
Loughborough, U.K.

Leila Esmaeili
Department of Computer Engineering
Qom Branch, Islamic Azad University
Qom, Iran

Iakov Korovin
Southern Federal University
Taganrog, Russia

***Abstract*—Population-based metaheuristic algorithms are problem-independent approaches to solve global optimisation problems. The human mental search (HMS) algorithm is a powerful population-based metaheuristic algorithm that has been shown to yield competitive performance for a variety of optimisation problems. HMS comprises three main operators, mental search, grouping, and movement. Mental search explores the neighbourhood of candidate solutions based on a Levy flight distribution to allow for simultaneous exploration and exploitation. Grouping is used to cluster the current population in order to find a promising area in search space, while during movement, candidate solutions move towards the identified promising area.**

In this paper, we propose an improved HMS algorithm – HMS-IS-OSK – that introduces an adaptive selection of the number of mental processes to improve the exploitation ability of HMS, and a one-step k-means algorithm for grouping to decrease the computational complexity. To evaluate the proposed algorithm, we perform a set of experiments on the CEC 2017 benchmark functions with dimensionalities of 30, 50, and 100. The obtained results show that HMS-IS-OSK outperforms standard HMS as well as other population-based metaheuristic algorithms including covariance matrix adaptation evolution strategy (CMA-ES), particle swarm optimisation (PSO), artificial bee colony algorithm (ABC), whale optimisation algorithm (WOA), grey wolf optimiser (GWO), and moth-flame optimisation (MFO).

***Index Terms*—Global optimisation; metaheuristic algorithms; human mental search.**

I. Introduction

Many real-world problems can be formulated as optimisation problems and consequently there is significant interest in effective global optimisation algorithms in both industry and academia [1], [2].

Conventional optimisation algorithms such as gradient-based approaches are often ineffective and sometimes not applicable [3]–[5] and suffer from local optima stagnation and the need to calculate derivative information [6]. To address these problems, population-based metaheuristic algorithms such as genetic algorithm (GA) [7], particle swarm optimisation (PSO) [8] and others, have been proposed.

Population-based metaheuristic algorithms generally start with a population of randomly generated candidate solutions. Then, new candidate solutions are generated based on defined stochastic operators. In this manner, information is shared among different candidate solutions, allowing candidate solutions to move towards a promising region in search space. In recent years, much work has been devoted to improving existing population-based algorithms [3], [5], [9] or proposing new algorithms [10]–[12].

Human mental search (HMS) [13] is a recent population-based metaheuristic algorithm inspired by the manner of exploring the bid space of online auctions. HMS comprises three operators: (1) mental search, which explores the neighbourhood of each candidate solution based on a Levy flight distribution; grouping which uses a clustering algorithm, k-means, to identify a promising region in search space, and movement during which candidate solutions approach the promising region.

HMS has shown very competitive performance in comparison to other state-of-the-art algorithms for a variety of optimisation problems with different characteristics such as unimodal, multi-modal, high-dimensional, complex, shifted, and rotated functions [13]. Furthermore, HMS outperforms other algorithms in classic engineering problems such as pressure vessel design, welded beam design, and three-bar truss design [13], while also having been used for computer vision applications such as image segmentation [14] and thresholding [15].

The mental search operator in HMS explores the vicinity of candidate solutions based on a Levy flight distribution, a type of random walk with varying step size, which can improve exploration and exploitation simultaneously. One parameter here is the number of searches that is performed for each candidate solution which is determined randomly between a pre-set minimum and maximum of mental processes. In this paper, we propose a novel adaptive approach to determine the number of mental searches to enhance exploitation ability of HMS.

In the grouping stage of HMS, standard k-means is em-

This paper is published due to the financial support of the Federal Target Programme of the Ministry of Science and Higher Education of Russian Federation, project unique identifier RFMEFI60819X0281.

ployed to cluster the current population, which is computationally expensive. To reduce the computation cost, in this paper, we propose the use of a one-step k-means algorithm for grouping. An extensive set of experiments conducted on the CEC 2017 benchmark functions [16] demonstrate that our proposed HMS-IS-OSK algorithm outperforms standard HMS as well as other population-based metaheuristic algorithms including covariance matrix adaptation evolution strategy (CMA-ES) [17], particle swarm optimisation (PSO) [8], artificial bee colony algorithm (ABC) [18], whale optimisation algorithm (WOA) [11], grey wolf optimiser (GWO) [10], and moth-flame optimisation (MFO) [12].

The remainder of the paper is organised as follows. Section II explains the original HMS algorithm. Our proposed improved HMS-IS-OSK algorithm is then described in Section III. Experimental results are presented in Section IV while Section V concludes the paper.

II. Human Mental Search

HMS [13] is a recent population-based metaheuristic algorithm for solving optimisation problems which is inspired by the manner of a human mental search in the bid space of online auctions. In HMS, each candidate solution is called a bid. Algorithm 1 defines HMS in terms of pseudo-code. Like other population-based metaheuristic algorithms, HMS starts with a population of random candidate solutions. HMS then iteratively improves these using the algorithm's main operators, mental search, grouping, and movement, which are described in the following.

A. Mental Search

During mental search, several new bids are generated around each current bid based on a random walk where the step size follows a Levy distribution. Since there are typically a number small steps followed by long jump, the use of the Levy distribution is designed to improve simultaneously exploration and exploitation in HMS.

A new position is calculated as

$$NS = bid + S, \tag{1}$$

with

$$S = (2 - NFE(2/NFE_{\max}))0.01\frac{u}{v^{1/\beta}}(x^i - x^*), \tag{2}$$

where NFE is the number of function evaluations so far, $NFE_{\max}$ is the maximum number of function evaluations, x^i is the current bid, and x^* the best bid found so far. u and v are random numbers calculated as

$$u \sim N(0, \sigma_u^2), \quad v \sim N(0, \sigma_v^2), \tag{3}$$

and

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\frac{\pi\beta}{2})}{\Gamma[(\frac{1+\beta}{2})]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \quad \sigma_v = 1, \tag{4}$$

where Γ is a standard gamma function.

The number of new bids for each bid generated around a bid is randomly chosen between M_L, the minimum number of mental processes, and M_H, the maxmum number of mental processes.

Algorithm 1 Pseudo-code of HMS algorithm.

procedure HMS ALGORITHM
    // Variables: L: lower bound; U: upper bound, M_l: minimum number of mental processes, M_h: maximum number of mental processes, N_{pop}: number of bids, N_{var}: number of variables, K: number of clusters, $iter$: current iteration, $NFE_{\max}$: maximum number of function evaluations

    X = initialise population of N_{pop} bids
    Calculate the objective function values of bids
    x^* = find the best bid in the initial population
    for i from 1 to N_{pop} **do**
        β_i = generate integer random number between L and U
    end for
    $NFE = N_{pop}$
    $iter = 0$
    while $NFE <= NFE_{\max}$ **do**
        $iter = iter + 1$
        // Mental Search
        for i from 1 to N_{pop} **do**
            q_i = generate integer random number between M_l and M_h
        end for
        for i from 1 to N_{pop} **do**
            for j from 1 to q_i **do**
                $s = (2 - \frac{2NFE}{NFE_{\max}})0.01\frac{u}{v^{1/\beta_i}}(x^i - x^*)$
                $NS_j = X^i + s$
            end for
            t = find NS with lowest objective function value
            if $cost(t) < cost(X^i)$ **then**
                $X^i = t$
            end if
        end for
        // Clustering
        Cluster N_{pop} bids into K clusters
        Calculate mean objective function value of each cluster
        Select cluster with lowest mean objective function value as winner cluster
        $winner$ = select best bid in winner cluster
        // Move bids towards best strategy
        for i from 1 to N_{pop} **do**
            for n from 1 to N_{var} **do**
                $X_n^i = X_n^i + C(r \times winner_n - X_n^i)$
            end for
        end for
        for i from 1 to N_{pop} **do**
            β_i = generate random number between L and U
        end for
        x^+ = find best bid in current bids
        if $cost(x^+) < cost(x^*)$ **then**
            $x^* = x^+$
        end if
    end while
end procedure

B. Grouping

The goal of the grouping operator is to cluster the population of bids, with standard HMS algorithm using k-means as the employed clustering algorithm. After population clustering, the mean objective function value of each cluster is calculated, and the cluster with the lowest mean objective function value (for a minimisation problem) is selected as the winner cluster representing a promising area in search space.

C. Movement

Having identified the promising area, other bids should move towards this area. To this end, other bids perform random movement towards the best bid in the promising area. In particular, bids are updated as

$$^{t+1}bid_n = {}^{t}bid_n + C(r \times {}^{t}winner_n - {}^{t}bid_n), \tag{5}$$

where $^{t+1}bid_n$ is the n-th bid element at iteration $t+1$, $^{t}winner_n$ is the n-th element of the best bid in the winner group, t is the current iteration, C is a constant, and r is a random number between 0 and 1 taken from the normal distribution.

III. Proposed algorithm

In this paper, we propose a new improved HMS algorithm named HMS-IS-OSK which is based on two new contributions. First, we present an adaptive approach to select the number of mental search processes, while second, during grouping we use a one-step k-means algorithm to reduce the computational burden.

A. Selection of the number of mental search processes

The number of mental searches in standard HMS is a random number between M_L (minimum number of mental processes) and M_H (maximum number of mental processes). In HMS-IS-OSK, instead of randomly determining the number of mental searches for each bid, it is chosen as a function of the objective function value. In particular, the number of mental searches is selected as being proportional to the quality of the bid as determined by its objective function value. As a result, the vicinity of a better candidate solution is searched more in order to increase the chances of finding the global optimum leading to an improved exploitation ability of HMS.

We calculate NML_i, the number of mental searches for candidate solution x_i, as

$$NML_i = M_L + \text{round}\left(\frac{N_{pop} - \text{rank}(x_i) + 1}{N_{pop}}(M_H - M_L)\right), \tag{6}$$

where $\text{rank}(x_i)$ is the rank of x_i when sorting all candidate solutions by objective function value with the best solution (the one with the lowest objective function value in a minimisation problem) being assigned rank 1 and the worst being assigned rank N_{pop} (the size of the population). Consequently, for the best solution M_H mental searches are performed, while M_L searches are conducted for the worst candidate solution.

B. One-step k-means clustering for grouping

In the grouping operator of HMS, clustering of bids is performed using the k-means algorithm [19]. Unfortunately, this represents a computational burden since k-means iteratively calculates the centroids of clusters and the mappings of samples to clusters until a convergence criterion is met.

Consequently, to improve the computational complexity, we propose to use a one-step k-means algorithm instead of standard k-means. Here, only one iteration of k-means is performed, and the grouping operator thus proceeds as follows:

Step 1: Initialise the cluster centres, $\{c_1, c_2, ..., c_K\}$, randomly.

Step 2: Assign each bid, $x_i, i = 1, ..., NP$, to its closest cluster centre, i.e. x_i is assigned to c_j if and only if $\|x_i - c_j\| \leq \|x_i - c_m\|$ for all $m = \{1, 2, ..., K\}$ with $j \neq m$, where $\|x_i - c_j\|$ is the (Euclidean) distance between x_i and c_m.

Step 3: Update the cluster centres as

$$c_i^{'} = \frac{1}{n_i} \sum_{x_i \in c_i} x_i, \tag{7}$$

where n_i is the number of bids in the i-th cluster.

IV. Experimental results

To evaluate the performance of HMS-IS-OSK, we examined our algorithm on the CEC 2017 benchmark functions [16]. This test suite comprises 30 benchmark functions with different characteristics including unimodal functions (F1 to F3), multi-modal functions (F4 to F10), hybrid multi-modal functions (F11 to F20) and composite functions (F21 to F30). In addition, we compare HMS-IS-OSK to the standard HMS algorithm as well as several other state-of-the-art optimisation algorithms, namely covariance matrix adaptation evolution strategy (CMA-ES) [17], particle swarm optimisation (PSO) [8], artificial bee colony algorithm (ABC) [18], whale optimisation algorithm (WOA) [11], grey wolf optimiser (GWO) [10], and moth-flame optimisation (MFO) [12].

In all experiments, the maximum number of function evaluations is set to $3000 \times D$ where D is the dimensionality of the search space with $D = \{30, 50, 100\}$ in the experiments. The parameters we use for the various algorithms are listed in Table I. Due to the stochastic nature of the algorithms, we run each algorithm 25 times on each problem and report the mean and standard deviation over these 25 runs, and as performance measure we use the difference between the (known) optimal function value and the result achieved by the algorithm.

TABLE I: Parameter settings.

algorithm	parameter	value
CMA-ES	λ	50
PSO	cognitive constant (C_1)	2
	social constant (C_2)	2
	inertia constant (w)	1 to 0
ABC	limit	# food sources $\times$ # dimensions
GWO	no parameters	
WOA	b	1
HMS	number of clusters	5
	C	1
	minimum mental process	2
	maximum mental process	5
HMS-IS-OSK	number of clusters	5
	C	1
	minimum mental process	2
	maximum mental process	5

TABLE II: Results for the two components of the proposed algorithm. The last row indicates the wins/ties/losses compared to standard HMS.

Functions	D=30			D=50			D=100		
	HMS-IS	HMS-OSK	HMS	HMS-IS	HMS-OSK	HMS	HMS-IS	HMS-OSK	HMS
F1	1.1333E+04	1.6803E+05	4.5474E+06	5.3850E+06	6.4243E+07	2.9857E+08	3.9148E+09	3.9744E+09	1.5123E+10
F2	2.0883E+20	1.3598E+23	4.1884E+24	3.2015E+48	8.5331E+48	1.4181E+53	3.5644E+127	7.1939E+119	6.5322E+134
F3	7.0237E+03	5.1203E+05	1.0924E+04	2.8626E+04	4.1968E+04	3.4819E+04	1.0341E+05	1.7306E+05	1.1390E+05
F4	1.0103E+02	5.4845E+02	1.2098E+02	2.1385E+02	2.9770E+02	3.0418E+02	9.1954E+02	1.0411E+03	2.0087E+03
F5	1.0250E+02	1.1356E+02	1.1852E+02	2.2535E+02	2.4017E+02	2.3754E+02	6.6112E+02	6.9696E+02	6.7842E+02
F6	2.9863E+00	7.6092E+00	1.0126E+01	1.3860E+01	1.8017E+01	1.8691E+01	2.8578E+01	3.3479E+01	3.3241E+01
F7	1.3699E+02	1.4789E+02	1.4604E+02	2.8241E+02	3.7119E+02	3.0006E+02	9.7624E+02	1.2030E+03	9.6739E+02
F8	9.5549E+01	1.1109E+02	1.0710E+02	2.5413E+02	2.6955E+02	2.5397E+02	6.8871E+02	7.1811E+02	7.1447E+02
F9	1.0772E+03	1.4792E+03	1.3897E+03	6.1325E+03	7.2766E+03	6.4648E+03	2.0369E+04	2.2115E+04	2.2191E+04
F10	3.5328E+03	3.9401E+03	3.7070E+03	7.1579E+03	7.1881E+03	7.5305E+03	1.6950E+04	1.7161E+04	1.7861E+04
F11	1.6724E+02	3.3811E+02	2.2836E+02	7.5596E+02	8.1194E+02	8.2531E+02	1.3425E+04	1.6146E+04	1.3478E+04
F12	3.7044E+06	6.6262E+06	7.4963E+06	3.2376E+07	7.4018E+07	1.0680E+08	4.8877E+08	7.3800E+08	1.5186E+09
F13	2.5964E+04	3.2470E+04	4.5431E+04	4.4274E+04	4.9945E+04	1.1729E+05	1.2891E+06	2.5622E+06	3.7796E+07
F14	4.0042E+04	3.4173E+04	4.2677E+04	1.9069E+05	2.3383E+05	2.8548E+05	1.3315E+06	2.0182E+06	2.0662E+06
F15	6.8041E+03	1.3026E+04	7.8994E+03	1.7076E+04	2.0128E+04	3.2367E+04	3.4688E+04	3.7170E+04	5.0022E+05
F16	9.4703E+02	9.4706E+02	9.1324E+02	1.9496E+03	2.1455E+03	2.1049E+03	5.5686E+03	5.2824E+03	6.0599E+03
F17	3.8137E+02	4.3350E+02	4.2775E+02	1.4912E+03	1.3949E+03	1.5141E+03	4.8976E+03	5.0145E+03	5.2535E+03
F18	4.5836E+05	3.3730E+05	5.3347E+05	1.2241E+06	1.8158E+06	1.7825E+06	3.1085E+06	5.0211E+06	5.6661E+06
F19	1.1615E+04	1.6062E+04	1.3463E+04	2.3796E+04	1.8379E+04	3.1230E+04	1.5384E+05	1.6172E+05	1.5409E+06
F20	3.7820E+02	3.8798E+02	3.2808E+02	1.0221E+03	1.1944E+03	1.0999E+03	3.5195E+03	3.3721E+03	3.6500E+03
F21	3.0852E+02	3.2537E+02	3.1963E+02	4.6371E+02	4.8690E+02	4.9060E+02	1.1027E+03	1.0823E+03	1.1029E+03
F22	3.3034E+03	3.9434E+03	3.7307E+03	7.3499E+03	7.9584E+03	7.8483E+03	1.8874E+04	1.8282E+04	1.9942E+04
F23	4.6829E+02	5.0025E+02	5.0421E+02	7.6946E+02	9.7867E+02	8.1351E+02	1.1941E+03	1.2985E+03	1.4144E+03
F24	5.7116E+02	6.0038E+02	5.8483E+02	8.1736E+02	8.6206E+02	8.7778E+02	1.7404E+03	1.8407E+03	1.9496E+03
F25	3.9016E+02	3.9578E+02	3.9819E+02	5.8227E+02	6.2760E+02	6.3361E+02	1.2039E+03	1.3452E+03	1.5947E+03
F26	2.2743E+03	2.2521E+03	2.4883E+03	4.6725E+03	5.0829E+03	5.4707E+03	1.3572E+04	1.3977E+04	1.5759E+04
F27	5.1569E+02	5.1965E+02	5.1885E+02	6.6463E+02	8.8420E+02	7.2196E+02	8.0290E+02	8.5092E+02	8.7511E+02
F28	4.8551E+02	4.8635E+02	5.1503E+02	5.3020E+02	1.0471E+03	7.0932E+02	1.0267E+03	2.3307E+03	2.3667E+03
F29	8.3567E+02	9.5045E+02	9.0647E+02	1.3729E+03	1.6076E+03	1.6741E+03	4.8350E+03	4.8265E+03	5.0434E+03
F30	1.0455E+04	1.9303E+04	3.3808E+04	1.2888E+06	1.6807E+06	2.3078E+06	1.2178E+06	1.1061E+06	1.0232E+07
w/t/l	28/0/2	13/0/17		28/0/2	18/0/12		29/0/1	24/0/6	

In our first experiment, we investigate – separately – the proposed modifications of the HMS algorithm. The results are given in Table II where HMS-IS indicates HMS with our proposed selection of mental searches, while HMS-OSK denotes HMS with one-step k-means.

We also conduct a two-sided Wilcoxon signed test with a confidence interval of 95% [20] to determine if the proposed improvements lead to statistical differences. Here, the null hypothesis H_0 indicates no difference between two algorithms, while the alternative hypothesis H_1 points to a statistical difference. Consequently, if the calculated p-value is below 0.05, there is a significant difference between two algorithms.

The results are given in Table III, from where we can observe that for $D = 30$ and $D = 50$, $p > 0.05$ indicating no significance difference between HMS-OSK and HMS, and thus that although computationally less complex, the performance of HMS-OSK is comparable to that of standard HMS. On the other hand, for $D = 100$, HMS-OK outperforms HMS and does so statistically. This is interesting, since while the motivation of the one step k-means algorithm was to lower the computational complexity, it can also lead to better optimisation performance, especially for higher-dimensional problems.

Looking at HMS-IS, from Table II we can see that it outperforms HMS for almost all cases, while the Wilcoxon signed rank test results in Table III confirm that this improved performance is statistically significant for all dimensionalities.

TABLE III: Results (p-values) of Wilcoxon signed rank test between the two proposed modifications and standard HMS.

	D=30	D=50	D=100
HMS-OSK vs. HMS	0.9918	0.1779	3.5888E-04
HMS-IS vs. HMS	1.9729E-05	1.9209E-06	3.1652E-06

Now turning our attention to the full HMS-IS-OSK algorithms, we give its results in Tables IV, V, and IV for $D = 30$, $D = 50$, and $D = 100$, respectively, while the tables also contain results for standard HMS and the other optimisation algorithms mentioned above.

From Table IV, we can see that for $D = 30$ HMS-IS-OSK gives the best or second best results for 27 of the 30 test functions and thus clearly yields the overall best-performing algorithm followed by standard HMS.

For $D = 50$ (Table V), HMS-IS-OSK outperforms all other algorithms for 27 of the 30 benchmark functions while it does so for 28 of the 30 functions for $D = 100$ (Table VI), which rather impressively demonstrates the superiority of our proposed algorithm compared to other state-of-the-art optimisation techniques. Standard HMS yields the second overall rank for both $D = 50$ and $D = 100$.

Table VII gives the results of the Wilcoxon signed rank test between HMS-IS-OSK and the other algorithms, showing our proposed algorithm to be statistically superior to all methods as indicated by p-values below 0.05 for all comparisons.

TABLE IV: Results for all algorithms and all functions for $D = 30$. For each function, we report (top row) the difference to the optimal function value and (bottom row) the ranking of the algorithm.

	CMA-ES	PSO	ABC	WOA	GWO	MFO	HMS	HMS-IS-OSK
F1	2.1057E+10	7.1568E+07	2.9068E+07	8.4266E+07	1.7484E+09	8.8022E+09	4.5474E+06	1.2433E+04
	8	4	3	5	6	7	2	1
F2	3.3285E+42	8.1280E+11	3.9300E+41	1.6974E+32	4.0913E+28	1.1593E+39	4.1884E+24	8.4311E+18
	8	1	7	5	4	6	3	2
F3	2.1087E+05	3.2432E+02	3.6790E+05	1.9935E+05	3.9428E+04	1.0779E+05	1.0924E+04	1.0997E+04
	7	1	8	6	4	5	2	3
F4	3.6176E+03	7.5395E+01	1.1866E+02	2.0201E+02	1.8606E+02	6.6353E+02	1.2098E+02	1.0075E+02
	8	1	3	6	5	7	4	2
F5	3.3072E+02	2.1189E+02	2.3934E+02	2.7780E+02	9.5905E+01	1.9786E+02	1.1852E+02	1.0165E+02
	8	5	6	7	1	4	3	2
F6	6.3041E+01	5.2053E+01	5.1961E+00	6.8248E+01	7.5763E+00	3.0375E+01	1.0126E+01	3.3328E+00
	7	6	2	8	3	5	4	1
F7	1.8099E+02	2.4703E+02	2.7763E+02	5.0560E+02	1.5277E+02	3.8272E+02	1.4604E+02	1.2812E+02
	4	5	6	8	3	7	2	1
F8	2.6860E+02	1.5672E+02	2.4705E+02	2.3121E+02	8.1409E+01	1.9039E+02	1.0710E+02	9.4055E+01
	8	4	7	6	1	5	3	2
F9	1.6636E+03	4.6448E+03	3.0760E+03	7.9868E+03	9.4833E+02	5.7595E+03	1.3897E+03	8.9200E+02
	4	6	5	8	2	7	3	1
F10	7.0296E+03	4.8263E+03	8.1736E+03	5.4595E+03	3.2182E+03	4.6322E+03	3.7070E+03	3.3349E+03
	7	5	8	6	1	4	3	2
F11	1.6670E+04	1.5259E+02	9.4044E+03	1.9779E+03	8.3078E+02	3.6395E+03	2.2836E+02	1.6631E+02
	8	1	7	5	4	6	3	2
F12	4.3228E+09	1.4466E+07	4.9689E+08	8.5179E+07	6.4799E+07	2.2258E+08	7.4963E+06	2.2742E+06
	8	3	7	5	4	6	2	1
F13	3.8909E+09	1.8713E+06	2.5698E+06	2.0706E+05	1.2327E+07	4.8566E+07	4.5431E+04	2.5190E+04
	8	4	5	3	6	7	2	1
F14	5.4543E+06	1.4107E+04	2.9837E+05	2.3871E+06	2.5657E+05	1.2983E+05	4.2677E+04	3.6053E+04
	8	1	6	7	5	4	3	2
F15	4.8952E+08	1.5636E+05	1.3958E+06	1.2634E+05	1.5845E+06	4.5340E+04	7.8994E+03	7.3424E+03
	8	5	6	4	7	3	2	1
F16	3.2451E+03	1.2571E+03	2.3367E+03	2.0587E+03	9.5852E+02	1.5687E+03	9.1324E+02	9.3363E+02
	8	4	7	6	3	5	1	2
F17	1.9053E+03	5.4329E+02	1.1393E+03	9.2401E+02	2.6596E+02	7.1002E+02	4.2775E+02	3.0558E+02
	8	4	7	6	1	5	3	2
F18	3.3457E+07	2.1741E+05	1.4622E+07	4.1731E+06	1.1282E+06	3.0607E+06	5.3347E+05	3.6901E+05
	8	1	7	6	4	5	3	2
F19	4.4237E+08	5.9588E+05	8.3738E+04	4.3773E+06	8.9645E+05	9.4809E+06	1.3463E+04	1.9372E+04
	8	4	3	6	5	7	1	2
F20	8.2500E+02	5.9648E+02	9.9359E+02	8.4174E+02	3.8497E+02	6.5689E+02	3.2808E+02	3.5631E+02
	6	4	8	7	3	5	1	2
F21	5.5102E+02	4.0542E+02	4.4639E+02	4.7918E+02	2.8400E+02	3.9721E+02	3.1963E+02	3.0464E+02
	8	5	6	7	1	4	3	2
F22	7.4914E+03	3.1253E+03	8.2462E+03	5.1366E+03	2.0673E+03	4.1025E+03	3.7307E+03	3.3341E+03
	7	2	8	6	1	5	4	3
F23	7.4573E+02	8.4375E+02	6.0663E+02	7.3688E+02	4.5410E+02	5.1771E+02	5.0421E+02	4.7455E+02
	7	8	5	6	1	4	3	2
F24	7.7132E+02	8.1723E+02	6.8304E+02	8.1269E+02	5.1170E+02	5.7800E+02	5.8483E+02	5.6434E+02
	6	8	5	7	1	3	4	2
F25	1.4006E+03	3.9652E+02	4.2031E+02	4.9556E+02	4.7356E+02	8.2039E+02	3.9819E+02	3.9060E+02
	8	2	4	6	5	7	3	1
F26	5.6196E+03	2.1556E+03	3.1500E+03	5.2144E+03	2.0699E+03	3.0636E+03	2.4883E+03	2.2927E+03
	8	2	6	7	1	5	4	3
F27	6.9922E+02	5.1884E+02	5.0001E+02	7.2429E+02	5.4557E+02	5.4642E+02	5.1885E+02	5.1347E+02
	7	3	1	8	5	6	4	2
F28	3.8029E+03	4.4412E+02	4.9998E+02	5.7303E+02	5.8176E+02	1.3554E+03	5.1503E+02	4.7622E+02
	8	1	3	5	6	7	4	2
F29	2.8292E+03	1.3993E+03	2.0172E+03	2.1742E+03	9.4969E+02	1.1897E+03	9.0647E+02	8.1921E+02
	8	5	6	7	3	4	2	1
F30	4.8792E+08	1.8814E+06	4.9885E+05	2.0393E+07	6.7067E+06	1.0783E+06	3.3808E+04	9.6989E+03
	8	5	3	7	6	4	2	1
average rank	7.4	3.7	5.5	6.2	3.4	5.3	2.77	1.77

TABLE V: Results For all algorithms and all functions for $D = 50$, laid out in the same fashion as Table IV.

	CMA-ES	PSO	ABC	WOA	GWO	MFO	HMS	HMS-IS-OSK
F1	4.6441E+10	2.5720E+08	2.8113E+09	4.9622E+08	5.5904E+09	3.7493E+10	2.9857E+08	4.5474E+06
	8	2	5	4	6	7	3	1
F2	5.0690E+79	2.6595E+25	5.4847E+80	1.2544E+70	7.9606E+54	6.9415E+76	1.4181E+53	4.1884E+24
	7	2	8	5	4	6	3	1
F3	3.7512E+05	9.4784E+03	7.7088E+05	1.8480E+05	9.4210E+04	1.6749E+05	3.4819E+04	1.0924E+04
	7	1	8	6	4	5	3	2
F4	7.6086E+03	1.6199E+02	3.2371E+03	5.3017E+02	7.1393E+02	4.0648E+03	3.0418E+02	1.2098E+02
	8	2	6	4	5	7	3	1
F5	3.0757E+02	3.9414E+02	5.4884E+02	4.7031E+02	2.0719E+02	4.6227E+02	2.3754E+02	1.1852E+02
	4	5	8	7	2	6	3	1
F6	6.7951E+01	6.9480E+01	4.9262E+01	8.2980E+01	1.7605E+01	4.7800E+01	1.8691E+01	1.0126E+01
	6	7	5	8	2	4	3	1
F7	2.2033E+02	5.0114E+02	6.5013E+02	1.0571E+03	3.5860E+02	1.1313E+03	3.0006E+02	1.4604E+02
	2	5	6	7	4	8	3	1
F8	5.1706E+02	4.1505E+02	5.5100E+02	4.4884E+02	2.1926E+02	4.7887E+02	2.5397E+02	1.0710E+02
	7	4	8	5	2	6	3	1
F9	1.3531E+04	2.4689E+04	4.2688E+04	2.7577E+04	8.1628E+03	1.7132E+04	6.4648E+03	1.3897E+03
	4	6	8	7	3	5	2	1
F10	1.3213E+04	8.8185E+03	1.4961E+04	1.0079E+04	6.4989E+03	7.4675E+03	7.5305E+03	3.7070E+03
	7	5	8	6	2	3	4	1
F11	6.4508E+04	3.5899E+02	5.5885E+04	1.1658E+03	2.9657E+03	1.2804E+04	8.2531E+02	2.2836E+02
	8	2	7	4	5	6	3	1
F12	2.2426E+10	1.0743E+08	1.0541E+10	5.7899E+08	5.4397E+08	4.6250E+09	1.0680E+08	7.4963E+06
	8	3	7	5	4	6	2	1
F13	1.2889E+10	1.4658E+07	6.1667E+07	5.5536E+06	1.0500E+08	8.7988E+08	1.1729E+05	4.5431E+04
	8	4	5	3	6	7	2	1
F14	2.1737E+07	1.1725E+05	4.2881E+06	1.9715E+06	9.4496E+05	1.0794E+06	2.8548E+05	4.2677E+04
	8	2	7	6	4	5	3	1
F15	2.5080E+09	3.0465E+06	1.2845E+07	1.0810E+06	1.5627E+07	8.6522E+06	3.2367E+04	7.8994E+03
	8	4	6	3	7	5	2	1
F16	5.3659E+03	2.1542E+03	5.1113E+03	3.8099E+03	1.4609E+03	2.7268E+03	2.1049E+03	9.1324E+02
	8	4	7	6	2	5	3	1
F17	1.0470E+03	1.6773E+03	3.1192E+03	2.5499E+03	1.0603E+03	2.3006E+03	1.5141E+03	4.2775E+02
	2	5	8	7	3	6	4	1
F18	1.2736E+08	1.4980E+06	6.8308E+07	1.4140E+07	5.0951E+06	2.9571E+06	1.7825E+06	5.3347E+05
	8	2	7	6	5	4	3	1
F19	1.1004E+09	2.7802E+06	2.5486E+04	5.1254E+06	3.1637E+06	5.9295E+07	3.1230E+04	1.3463E+04
	8	4	2	6	5	7	3	1
F20	1.6233E+03	1.3045E+03	2.5033E+03	1.6992E+03	8.9671E+02	1.4003E+03	1.0999E+03	3.2808E+02
	6	4	8	7	2	5	3	1
F21	7.9945E+02	6.6140E+02	7.5220E+02	8.4253E+02	3.9388E+02	6.4065E+02	4.9060E+02	3.1963E+02
	7	5	6	8	2	4	3	1
F22	1.4255E+04	9.0947E+03	1.5100E+04	1.0073E+04	6.7774E+03	8.2823E+03	7.8483E+03	3.7307E+03
	7	5	8	6	2	4	3	1
F23	1.1399E+03	1.5510E+03	9.7798E+02	1.3481E+03	6.6419E+02	8.4139E+02	8.1351E+02	5.0421E+02
	6	8	5	7	2	4	3	1
F24	1.1408E+03	1.2074E+03	1.0874E+03	1.3049E+03	7.2354E+02	8.1562E+02	8.7778E+02	5.8483E+02
	6	7	5	8	2	3	4	1
F25	2.7216E+03	5.1174E+02	2.2217E+03	9.1079E+02	1.0626E+03	3.0814E+03	6.3361E+02	3.9819E+02
	7	2	6	4	5	8	3	1
F26	8.7578E+03	6.0903E+03	6.3256E+03	1.0536E+04	3.9505E+03	5.8167E+03	5.4707E+03	2.4883E+03
	7	5	6	8	2	4	3	1
F27	1.1203E+03	1.0704E+03	5.0001E+02	1.6911E+03	8.9893E+02	8.7495E+02	7.2196E+02	5.1885E+02
	7	6	1	8	5	4	3	2
F28	6.7681E+03	4.9726E+02	5.0001E+02	1.1993E+03	1.4770E+03	5.1097E+03	7.0932E+02	5.1503E+02
	8	1	2	5	6	7	4	3
F29	1.0413E+04	2.5531E+03	6.2498E+03	4.8495E+03	1.6206E+03	2.3113E+03	1.6741E+03	9.0647E+02
	8	5	7	6	2	4	3	1
F30	2.4322E+09	5.5781E+07	3.3766E+08	1.3506E+08	1.0880E+08	1.2885E+08	2.3078E+06	3.3808E+04
	8	3	7	6	4	5	2	1
average rank	6.77	4.00	6.23	5.93	3.63	5.33	2.97	1.13

TABLE VI: Results For all algorithms and all functions for $D = 100$, laid out in the same fashion as Table IV.

	CMA-ES	PSO	ABC	WOA	GWO	MFO	HMS	HMS-IS-OSK
F1	4.2008E+10	1.0664E+09	3.6114E+11	1.9755E+07	4.2486E+10	1.2403E+11	1.5123E+10	4.5474E+06
	5	3	8	2	6	7	4	1
F2	1.0383E+166	8.4451E+71	2.9588E+182	1.1464E+148	4.3009E+133	7.9917E+159	6.5322E+134	4.1884E+24
	7	2	8	5	3	6	4	1
F3	8.5028E+05	2.3560E+05	2.2239E+06	7.5474E+05	2.4503E+05	6.5534E+05	1.1390E+05	1.0924E+04
	7	3	8	6	4	5	2	1
F4	1.7146E+04	3.8656E+02	1.5305E+05	6.6924E+02	3.3637E+03	2.3752E+04	2.0087E+03	1.2098E+02
	6	2	8	3	5	7	4	1
F5	1.0710E+03	1.0851E+03	2.0306E+03	9.2522E+02	6.4067E+02	1.1967E+03	6.7842E+02	1.1852E+02
	5	6	8	4	2	7	3	1
F6	1.4423E+01	8.5974E+01	1.3922E+02	7.9330E+01	3.6507E+01	7.3175E+01	3.3241E+01	1.0126E+01
	2	7	8	6	4	5	3	1
F7	8.6477E+02	1.2510E+03	9.1900E+03	2.5687E+03	1.2469E+03	4.0405E+03	9.6739E+02	1.4604E+02
	2	5	8	6	4	7	3	1
F8	1.2745E+03	1.1991E+03	2.0784E+03	1.1186E+03	6.3276E+02	1.2651E+03	7.1447E+02	1.0710E+02
	7	5	8	4	2	6	3	1
F9	3.2532E+04	6.3247E+04	1.9289E+05	3.5369E+04	3.2701E+04	4.2072E+04	2.2191E+04	1.3897E+03
	3	7	8	5	4	6	2	1
F10	3.0353E+04	2.1981E+04	3.2594E+04	1.9488E+04	1.5669E+04	1.6792E+04	1.7861E+04	3.7070E+03
	7	6	8	5	2	3	4	1
F11	4.5125E+05	2.4397E+03	8.8802E+05	1.1656E+04	5.3270E+04	1.3235E+05	1.3478E+04	2.2836E+02
	7	2	8	3	5	6	4	1
F12	5.2721E+10	8.2741E+08	1.1872E+11	7.2320E+08	7.4034E+09	3.4815E+10	1.5186E+09	7.4963E+06
	7	3	8	2	5	6	4	1
F13	1.1649E+10	5.0818E+07	3.1252E+09	6.1953E+04	4.1458E+08	4.4317E+09	3.7796E+07	4.5431E+04
	8	4	6	2	5	7	3	1
F14	1.1494E+08	1.5734E+06	8.6075E+07	1.4506E+06	4.9534E+06	8.1393E+06	2.0662E+06	4.2677E+04
	8	3	7	2	5	6	4	1
F15	5.7293E+09	1.5174E+07	5.6989E+08	6.2129E+04	1.0217E+08	1.8139E+09	5.0022E+05	7.8994E+03
	8	4	6	2	5	7	3	1
F16	1.2045E+04	5.8526E+03	1.5177E+04	8.4316E+03	4.4163E+03	6.3580E+03	6.0599E+03	9.1324E+02
	7	3	8	6	2	5	4	1
F17	4.2235E+04	4.3163E+03	2.9662E+04	5.4693E+03	3.5358E+03	6.4352E+03	5.2535E+03	4.2775E+02
	8	3	7	5	2	6	4	1
F18	1.3858E+08	3.1777E+06	2.8617E+08	1.8815E+06	4.8340E+06	1.6349E+07	5.6661E+06	5.3347E+05
	7	3	8	2	4	6	5	1
F19	4.4419E+09	2.6744E+07	7.2872E+06	1.3301E+07	1.1502E+08	7.6923E+08	1.5409E+06	1.3463E+04
	8	5	3	4	6	7	2	1
F20	5.1410E+03	3.7676E+03	6.5695E+03	4.2334E+03	3.2788E+03	3.7754E+03	3.6500E+03	3.2808E+02
	7	4	8	6	2	5	3	1
F21	1.4369E+03	1.6924E+03	2.5583E+03	1.7856E+03	8.6885E+02	1.5533E+03	1.1029E+03	3.1963E+02
	4	6	8	7	2	5	3	1
F22	3.1095E+04	2.4380E+04	3.3449E+04	2.1741E+04	1.7406E+04	1.8352E+04	1.9942E+04	3.7307E+03
	7	6	8	5	2	3	4	1
F23	1.8416E+03	2.9073E+03	3.1588E+03	2.4400E+03	1.2305E+03	1.4872E+03	1.4144E+03	5.0421E+02
	5	7	8	6	2	4	3	1
F24	2.4409E+03	3.0356E+03	5.1587E+03	3.5545E+03	1.7524E+03	1.9440E+03	1.9496E+03	5.8483E+02
	5	6	8	7	2	3	4	1
F25	8.2654E+03	9.4492E+02	8.5519E+04	1.1255E+03	3.4180E+03	1.1216E+04	1.5947E+03	3.9819E+02
	6	2	8	3	5	7	4	1
F26	1.9266E+04	1.7110E+04	4.9569E+04	2.8247E+04	1.2222E+04	1.5335E+04	1.5759E+04	2.4883E+03
	6	5	8	7	2	3	4	1
F27	1.9963E+03	6.4331E+02	5.0002E+02	2.2667E+03	1.3314E+03	1.2842E+03	8.7511E+02	5.1885E+02
	7	3	1	8	6	5	4	2
F28	1.9124E+04	6.3827E+02	5.0002E+02	9.1321E+02	4.7270E+03	1.6200E+04	2.3667E+03	5.1503E+02
	8	3	1	4	6	7	5	2
F29	1.3385E+04	6.9356E+03	6.0959E+04	1.1110E+04	5.2786E+03	8.2201E+03	5.0434E+03	9.0647E+02
	7	4	8	6	3	5	2	1
F30	1.0157E+10	1.0689E+08	3.2747E+09	1.8263E+08	1.0336E+09	2.3325E+09	1.0232E+07	3.3808E+04
	8	3	7	4	5	6	2	1
average rank	6.30	4.17	7.13	4.57	3.73	5.60	3.43	1.06

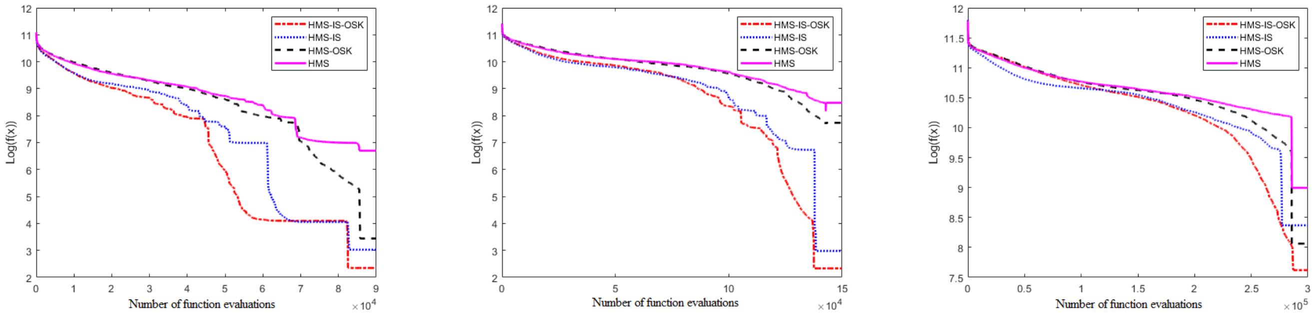

Fig. 1: Convergence curves for $F1$ function for $D = 30$ (left), $D = 50$ (middle), and $D = 100$ (right).

TABLE VII: Results of Wilcoxon signed rank test between HMS-IS-OSK and other algorithms.

	$D = 30$	$D = 50$	$D = 100$
HMS-IS-OK vs. CMA-SA	1.7344E-06	1.7344E-06	1.7344E-06
HMS-IS-OK vs. PSO	0.0387	9.3157E-06	1.7344E-06
HMS-IS-OK vs. ABC	2.1266E-06	2.3534E-06	2.3534E-06
HMS-IS-OK vs. WOA	1.7344E-06	1.7344E-06	1.7344E-06
HMS-IS-OK vs. GWO	0.0032	1.7344E-06	1.7344E-06
HMS-IS-OK vs. MFO	1.7344E-06	1.7344E-06	1.7344E-06
HMS-IS-OK vs. HMS	3.3173E-04	1.7344E-06	1.7344E-06

Last not least, we investigate the convergence behaviour of our proposed modifications. For this, we plot the convergence curves for (as a representative) the $F1$ benchmark function in Figure 1. As can be observed from there, HMS-IS-OSK consistently converges the fastest.

V. Conclusions

In this paper, we have proposed HMS-IS-OSK, an improved human mental search (HMS) optimisation algorithm. HMS-IS-OK introduces two improvements to standard HMS. First, to lessen the computational complexity, a one-step k-means clustering algorithm is employed instead of standard k-means during the grouping process of HMS. Second, an adaptive approach is proposed to select the number of mental search processes in order to yield better exploitation in vicinity of a good candidate solutions. Experimental results obtained on 30 CEC 2017 benchmark functions confirm superior performance of HMS-IS-OSK compared to standard HMS and to a number of state-of-the-art population-based optimisation algorithms. We are currently investigating the use of HMS-IS-OSK for optimising neural networks, software systems, and abnormal behaviour detection.

References

[1] E. Wari and W. Zhu, “A survey on metaheuristics for optimization in food manufacturing industry,” *Applied Soft Computing*, vol. 46, pp. 328–343, 2016.

[2] T. K. Sharma, M. Pant, and M. Singh, “Nature-inspired metaheuristic techniques as powerful optimizers in the paper industry,” *Materials and Manufacturing Processes*, vol. 28, no. 7, pp. 788–802, 2013.

[3] L. Cui, G. Li, Z. Zhu, Q. Lin, K.-C. Wong, J. Chen, N. Lu, and J. Lu, “Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism,” *Information Sciences*, vol. 422, pp. 122–143, 2018.

[4] Y. Hu, C.-K. Sim, and X. Yang, “A subgradient method based on gradient sampling for solving convex optimization problems,” *Numerical Functional Analysis and Optimization*, vol. 36, no. 12, pp. 1559–1584, 2015.

[5] S. J. Mousavirad and S. Rahnamayan, “Differential evolution algorithm based on a competition scheme,” in *14th International Conference on Computer Science and Education*, 2019.

[6] S. J. Mousavirad, A. A. Bidgoli, H. Ebrahimpour-Komleh, and G. Schaefer, “A memetic imperialist competitive algorithm with chaotic maps for multi-layer neural network training,” *International Journal of Bio-Inspired Computation*, vol. 14, no. 4, pp. 227–236, 2019.

[7] D. Whitley, “A genetic algorithm tutorial,” *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.

[8] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *IEEE International Conference on Evolutionary Computation*, 1998, pp. 69–73.

[9] S. J. Mousavirad, A. Asilian Bidgoli, and S. Rahnamayan, “Tackling deceptive optimization problems using opposition-based DE with center-based Latin hypercube initialization,” in *14th International Conference on Computer Science and Education*, 2019.

[10] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[11] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[12] S. Mirjalili, “Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm,” *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.

[13] S. J. Mousavirad and H. Ebrahimpour-Komleh, “Human mental search: a new population-based metaheuristic optimization algorithm,” *Applied Intelligence*, vol. 47, no. 3, pp. 850–887, 2017.

[14] S. J. Mousavirad, H. Ebrahimpour-Komleh, and G. Schaefer, “Effective image clustering based on human mental search,” *Applied Soft Computing*, vol. 78, pp. 209–220, 2019.

[15] S. J. Mousavirad and H. Ebrahimpour-Komleh, “Human mental search-based multilevel thresholding for image segmentation,” *Applied Soft Computing*, 2019.

[16] G. Wu, R. Mallipeddi, and P. Suganthan, “Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization,” Nanyang Technological University, Singapore, Tech. Rep., 2016.

[17] N. Hansen and A. Ostermeier, “Completely derandomized self-adaptation in evolution strategies,” *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[18] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,” *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[19] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *5th Berkeley Symposium on Mathematical Statistics and Probability*. Oakland, CA, USA, 1967, pp. 281–297.

[20] J. Derrac, S. García, D. Molina, and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,” *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.