Pamukkale Univ Muh Bilim Derg, 29(3), 264-273, 2023

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

Particle Swarm Optimization with a new intensification strategy based on K-Means

K-Ortalamalara dayalı yeni yoğunlaştırma stratejisi ile Parçacık Sürüsü Optimizasyonu

Tahir SAG[1], *Aysegul IHSAN*[2*]

[1]Computer Engineering, Faculty of Technology, Selcuk University, Konya, Turkey.
tahirsag@selcuk.edu.tr
[2]Information Technologies Engineering, Faculty of Technology, Selcuk University, Konya, Turkey.
aysegulihsann@gmail.com

Received/Geliş Tarihi: 26.01.2022 Revision/Düzeltme Tarihi: 12.08.2022 doi: 10.5505/pajes.2022.37898
Accepted/Kabul Tarihi: 15.08.2022 Research Article/Araştırma Makalesi

Abstract

Particle Swarm Optimization (PSO) is a swarm intelligence-based metaheuristic algorithm inspired by the foraging behaviors of fish or birds. Despite the advantages of having a simple and effective working structure, PSO also has some disadvantages, such as early convergence, getting trapped in local minima, and weak global search capabilities. In this study, a novel intensification strategy based on K-Means clustering has been proposed to enhance the performance of PSO. The proposed method is called Particle Swarm Optimization with a New Intensification Strategy based on K-Means (PSO-ISK). In the first step of PSO-ISK, particles in PSO are grouped into different clusters. Then, a center and the farthest particle from the center are identified for each cluster. PSO-ISK proposes a new intensification strategy by improving the results of the farthest particle from the center. The performance of PSO-ISK is analyzed using 16 different benchmark test functions. The obtained results are compared with Standard PSO (SPSO) and 7 different PSO variants. According to the comparison results, PSO-ISK provides a notable performance improvement by outperforming SPSO and all seven PSO variants. The comparisons conducted have proven that PSO-ISK produces more effective outcomes than other studies, which results in a significant contribution to improving performance.

Keywords: Intensification strategy, K-Means, PSO.

Öz

Parçacık Sürü Optimizasyonu (PSO), sürü zekâsı temelli metaheuristik algoritmadır. PSO, balıkların veya kuşların yiyecek arama davranışlarından esinlenilerek modellenmiştir. PSO, sade ve etkili bir çalışma yapısına sahip olmasının avantajlarına rağmen, erken yakınsama, yerel minimuma takılma ve zayıf küresel arama kapasitesi gibi bazı dezavantajları da bulunmaktadır. Bu çalışmada, PSO'nun performansını artırmak için K-Ortalamalar kümelemeye dayalı yeni bir yoğunlaştırma stratejisi önerilmiştir. Önerilen yönteme, K-Ortalamalara Dayalı Yeni Yoğunlaştırma Stratejisi ile Parçacık Sürüsü Optimizasyonu (PSO-ISK) adı verilmiştir. PSO-ISK'nın ilk adımında, PSO'daki parçacıklar farklı kümelere ayrılmaktadır. Sonraki adımda ise, her küme için bir merkez ve merkeze en uzak parçacık belirlenmektedir. Bu çalışmanın sonucunda, PSO-ISK, merkeze en uzak parçacığın sonuçlarını iyileştirerek yeni bir yoğunlaştırma stratejisi önermektedir. PSO-ISK'nın performansı, 16 farklı benchmark test fonksiyonu kullanılarak sonuçlar analiz edilmiştir. Elde edilen sonuçlar, Standart PSO (SPSO) ve 7 farklı PSO varyantı ile karşılaştırılmıştır. Yapılan karşılaştırmalar sonucunda, PSO-ISK'nın diğer çalışmalara göre daha etkili sonuçlar elde ettiği ve PSO-ISK'nın performans iyileştirmesindeki önemi kanıtlamıştır.

Anahtar kelimeler: K-Ortalamalar, PSO, Yoğunlaştırma stratejisi.

1 Introduction

Swarm intelligence (SI) is a research topic that has not lost its popularity for decades, simulating the collective behavior of primitive organisms living together to solve complex problems. Therefore, new algorithms in this specialized literature are still proposed by researchers and successfully applied to several engineering optimization problems, such as horse herd optimization algorithm [1], jellyfish search optimizer [2], gradient-based optimizer [3], political optimizer [4], sailfish optimizer [5], butterfly optimization algorithm [6], tree-seed algorithm [7], social spider algorithm [8]. On the other hand, one of the well-known algorithms among the state-of-the-art metaheuristics is particle swarm optimization (PSO) [9], which is inspired by the collective movement of flocks of birds and fish. Although to find global optima, PSO has an efficient strategy using the cognitive and social experience of the candidate solutions called particles, it can be getting stuck in local optimal points due to reduced intensification during iterations. To cope with this problem, various modification studies on PSO have been presented up to now [10]-[16]. Clustering is the well-studied data mining process of grouping similar data objects in an extensive dataset into a number of classes. The main advantage of this technique is that it is adaptable to changes. The best-known clustering algorithm is K-Means which has a simple and effective structure. It aims to partition the data objects into predefined k sets to minimize the within-cluster sum of squares. This study focuses on improving the performance of PSO using K-Means. A brief literature review on PSO with clustering techniques is given as follows.

PSO and K-Means have been used together in various studies in the literature. One of them was presented by Solaiman et al. [17]. They proposed a hybrid method based on K-Means and PSO to ensure efficient energy management of wireless sensor networks. They reported that their hybrid method was superior to conventional protocols. In another study, Gao et al. presented a hybrid version of PSO and K-Means algorithms [18]. They also used Gaussian distribution and Lévy flight strategy to escape from the local optimum. In a recent study,

*Corresponding author/Yazışılan Yazar

Mahajan et al. presented another hybrid method of K-Means with PSO to optimize cluster formation. Then they applied it to improve the efficiency of the prediction of environmental pollution [19]. Sun et al. evaluated the fabric's tactile comfort with a new clustering algorithm of self-adaptive PSO based on K-Means [20]. Younus et al. presented a hybrid method of PSO and K-Means to accurately retrieve images from large image databases [21]. Liu et al. developed a new hybrid algorithm for data analysis focusing on PSO and K-Means [22]. In another study, Jamali et al. studied the disadvantages of the K-Means algorithm, which are its sensitivity to noisy features and its dependence on the selection of the initial cluster centers. Then they improved the K-Means with PSO and investigated the performance of their method on lateralizing the epileptogenic hemisphere for temporal lobe epilepsy cases [23]. Tarkhaneh et al. proposed a hybrid approach combining four techniques: Cuckoo Search, PSO, K-Means algorithms, and levy distribution [24]. They used PSO and K-Means for generating more efficiency, while levy flight was used to obtain faster convergence. This study proposes a novel variant of PSO with the K-Means algorithm, which is called Particle Swarm Optimization with K-Means. To evolve the exploration capability of PSO, a novel hybrid method is created by adapting K-Means within PSO. The proposed method is named Particle Swarm Optimization with a New Intensification Strategy Based on K-Means (PSO-ISK). Here, unlike many previous studies, K-Means and PSO algorithms are not used in the field of data analysis but directly to increase the exploration capability of PSO in the field of optimization. The main contributions of this study can be summarized as follows:

- The K-Means algorithm is adapted to use it more effectively in the optimization process.
- A novel intensification strategy for PSO is proposed to improve the exploration capability so that PSO can overcome stagnation during iterations.
- To prove the effectiveness of PSO-ISK, experimental results are calculated by running 16 test problems that are frequently used in the literature.
- A comprehensive analysis is provided by comparing PSO-ISK with the standard PSO algorithm as well as the seven PSO variants stated in the literature.

The content provided in the rest of this paper is organized as follows: In Section II, standard PSO and K-Means algorithms are briefly summarized. In Section III, PSO-ISK is described in detail. In Section IV, the experiments and comparisons are presented, and the results are discussed. Finally, Section V contains conclusions and future study.

2 Background

The PSO algorithm and the K-Means approach are examined in this section. Following that, the PSO-ISK algorithm is examined.

2.1 Particle swarm optimization algorithm

PSO is one of the most popular SI algorithms, inspired by the social behavior of primitive agents such as birds and fish in a flock [9]. In the analogy of PSO, a number of agents fly over the environment to search for the best food source. For this purpose, they use each other's cognitive and social experiences. In the PSO algorithm, these agents are called particles, which means the candidate solutions and the best food source means the global optimum in the search space. In addition, the quality of food sources is measured by a fitness function based on the decision variables. The particles are randomly generated by using Equation-(1) in the initial swarm.

$$x_{i,j}^{0} = lower_j^{min} + rand_i^{j} \times \left(upper_j^{max} - lower_j^{min}\right) \quad (1)$$

where $i = 1,2, \dots, N$ and $j = 1,2, \dots, D$ and $x_{i,j}^{0}$ indicates the j-th parameter of an i-th particle in the initial iteration, $lower_j^{min}$ and $upper_j^{max}$ are the lower limit and upper limit for j-th dimension, respectively. $rand_i^{j}$ is a random number generated in the range of $[0, 1]$ for each dimension, N is the swarm size, and D is the dimension of the optimization problem. The locations of the particles are kept in memory then the particles act according to the two best values: p_{best} and g_{best}. The p_{best} is the personal best value of a particle found up to the current iteration. The g_{best} is the best value of all particles in the swarm found up to the current iteration. Apart from the particles in the swarm, the p_{best} values for each particle and g_{best} particle are stored externally in memory. The amount of position update for a particle is defined as the velocity, which is calculated using Equation-(2). Then the locations of all particles are updated with Equation-(3) in each iteration.

$$v_{i,j}^{t+1} = w \times v_{i,j}^{t} + c_1 \times rand_1^{t} \times \left({p_{best}}_{i,j}^{t} - x_{i,j}^{t}\right) + c_2 \times rand_2^{t} \times \left({g_{best}}_{i,j}^{t} - x_{i,j}^{t}\right) \quad (2)$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t+1} \quad (3)$$

where, $v_{i,j}^{t+1}$ is the velocity of the j-th parameter of an i-th particle at iteration $t + 1$. The $rand_1$ and $rand_2$ parameters are random numbers generated in the range of $[0, 1]$. The coefficients c_1 and c_2 are used to reflect the cognitive and social information on location updates, respectively. w is the inertia weight. In this study, inertia weight is calculated by Equation-(4).

$$w = \frac{(Max_{iter} - iter)}{Max_{iter}} \quad (4)$$

After updating the location for a particle, a greedy selection procedure is implemented between the current p_{best} and the new calculated value. The better one according to the fitness function is selected for the new p_{best}. At the end of each iteration, a similar procedure is applied to update p_{best}. PSO iteratively repeats this location update process until a predetermined termination condition is satisfied. Finally, the g_{best} value obtained in the last iteration is given as the result of the algorithm.

2.2 The K-Means algorithm

The K-Means algorithm is a straightforward and efficient clustering method that is the most popular unsupervised learning algorithm in data mining literature. The term "K-Means" as a whole was first used by MacQueen in 1967 [25]. On the other hand, its idea was previously presented by Steinhaus in 1956. K-Means must take the number of clusters as a parameter to divide the data into clusters. This is actually a disadvantage. On the other hand, the algorithm has a simple running characteristic. After the k-value is determined, the algorithm randomly selects the center points. It calculates the distance between each data and randomly determines center points and assigns the data to a cluster according to the nearest center point. Then, a center point is selected again for each

cluster, and clustering is done according to the new center points. This continues until the system becomes stable. For this purpose, the average coordinate values are calculated for each cluster in each iteration, and the calculated value becomes the new center of that cluster. The random assignment of the initial center points in the K-Means algorithm can cause some issues. As a result, many approaches for determining initial centers have been presented.

Let $x = \{x_1, x_2, x_3, \dots, x_m\}$ be a dataset with N dimensions and k represents the vector of center points, where m is the number of samples in the dataset. The K-Means algorithm consists of 4 stages:

1. Determination of cluster centers: In the first iteration, the center points are randomly generated. In the next iterations, new center points are determined by calculating the centers of the previously classified samples,
2. Classification according to the distances to the given center points: The Euclidean distances of the samples to each cluster center are calculated using Equation-(5). Then the sample is assigned to the cluster center closest to it,

$$D_i = \sqrt{\sum_{i=1}^{m} \sum_{dim=1}^{n} \left({x_i}^{dim} - {k_i}^{dim} \right)^2} \quad (5)$$

3. Determination of new centers according to the classification made: The coordinates of the center points are updated by averaging the assigned samples. This can be formulated as Equation-(6),

$${k_i}^{dim} = \left(\sum_{dim=1}^{n} {z_i}^{dim} \right) / |k_i| \quad z = \{p : p \in k_i\} \quad (6)$$

where, $|k_i|$ represents the cardinality of k_i.

4. Repeating steps 2 and 3 until stable: The healthiest method for the termination condition of the algorithm is to create an optimization criterion. The goal is to keep the value of this objective function as low as possible. In other words, it is to minimize the sum of the intra-cluster distance. For each cluster, the distances from the center point of each point in it are summed. The algorithm continues to run until these values do not change or until little changes occur.

2.3 Particle swarm optimization with a k-means based on intensification strategy

With a novel intensity-enhancing method, the K-Means clustering strategy is adapted to PSO to improve PSO performance. The proposed algorithm is named Particle Swarm Optimization with a New Intensification Strategy Based on K-Means (PSO-ISK). The explanation of the PSO-ISK algorithm can be given in two stages: intensifying the particle farthest from the center and updating the location of the farthest particles.

2.3.1 Determination of the particle farthest from the center

PSO-ISK is initialized with a randomly generated population, just like in the standard PSO algorithm. The particles are then clustered by the K-Means algorithm into a predefined number of sub-groups according to the dimensions of the optimization problem. For each cluster, the particle with the best fitness value is selected as the g_{best} particle (center) of that cluster, while the particle with the worst fitness value in the cluster is defined as the farthest particle from the center. For example, suppose the number of clusters for a colony with 30 particles is 3 ($k = 3$) in a 2-dimensional search space (x_1, x_2). First, the cluster centers are selected for each cluster, which is shown with stars in Figure 1(a). Next, the farthest particle to the center (marked with a circle) is determined for each cluster.

2.3.2 Updating the location of the farthest particles

A location update is performed for the farthest particles selected from each cluster. For PSO-ISK, Equation-(7), which is the revised version of Equation-(2), calculates the velocity of the particle.

$$v_{i,j}^{t+1} = w \times v_{i,j}^{t} + c_1 \times rand_1^t \times \left({p_{best}}_{i,j}^{t} - fartest(c)_{i,j}^{t} \right) + c_2 \times rand_2^t \times \left(center(c)_{i,j}^{t} - fartest(c)_{i,j}^{t} \right) \quad (7)$$

where, $center(c)_{i,j}^{t}$ represents the center of the cluster in the current iteration and $c = 1,2, \dots, k$ and c is the cluster-index of the farthest particle to be updated, and k is the total number of clusters. $fartest(c)_{i,j}^{t}$ is the cluster's furthest particle in the current iteration. $v_{i,j}^{t+1}$ denotes the velocity of an i-th particle's j-th parameter at iteration $t + 1$. Random numbers in the range [0,1] are created using the $rand_1$and $rand_2$ parameters. The c_1 and c_2 coefficients represent cognitive and social information on location updates, respectively. w is the inertia weight. After making the velocity updates, new fitness values of the particles are calculated, and if necessary, $p_{best}$and g_{best}. updates are performed, and the same procedure is continued until the stop criteria are provided. In Equation-(7), the parameter values c_1 and c_2 are equal to 2. Because comparison findings are obtained from other papers. These algorithms are LFPSO [10], CLPSO [12], HPSO-TVAC [13], FIPSO [14], SPSO-40 [15], LPSO [15], and DMS-PSO [16]. Because the parameters c_1 and c_2 are equal to 2 in the aforementioned algorithms, the parameters c_1 and c_2 are equal to 2 in this study for comparison under the same conditions [10]-[16].

Consequently, the intensity of the swarm is improved by updating the quality of the farthest particle through convergence to the best solution in the region. In Figure 1(b), the updated locations of the farthest particles are simulated for three clusters.

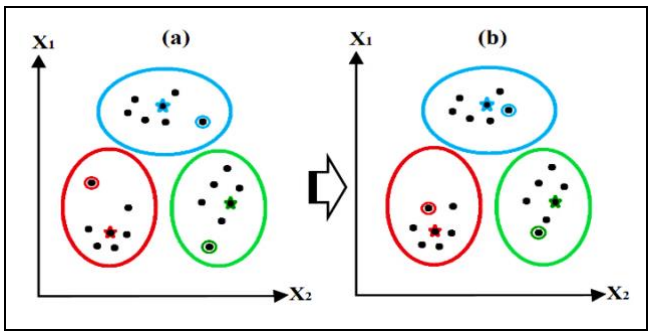

Figure 1(a): The centers and the farthest particles for each cluster. (b): The updated locations for the farthest particles.

The implementation of this step in each iteration ensures that the intensity is distributed across different regions in the search space. In addition, it does not cause premature convergence due to the small number of determined clusters.

To ensure the total number of function evaluations is equal to that in standard PSO, $(N-k)$ particles are generated in the PSO-ISK algorithm since k extra functions are evaluated in each iteration in the algorithm. Figure 2 illustrates the flowchart for the PSO-ISK algorithm. Figure 3 shows the procedure strategies that detail the PSO-ISK algorithm's step-by-step execution. The execution of the PSO-ISK algorithm is shown in pseudo-code in Figure 4.

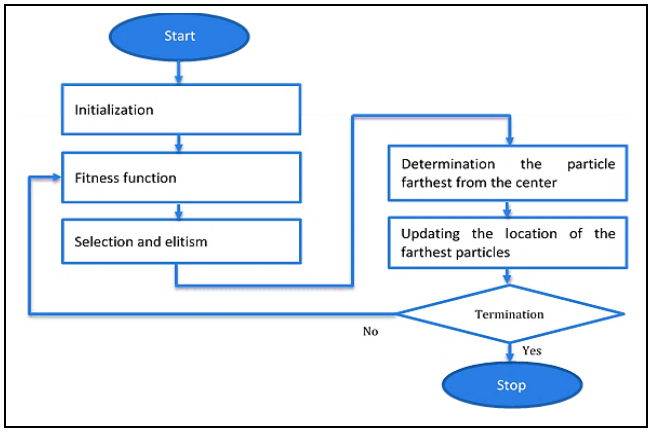

Figure 2. Flowchart of the PSO-ISK algorithm.

Step 1. Initialization

In the first step, an initial population is generated randomly. Each individual is considered a potential solution for the problem.

Step 2. Fitness function

In this iteration, the particles are moved to the new locations inside the search space using the positions and velocities that are calculated in the first iteration. The velocities of all particles are then calculated to move the particles in the next iteration. Moreover, the fitness values of all particles are calculated.

Step 3. Selection and elitism

In this iteration, the particles continue moving toward the optimal solution. At the beginning, the particles are moved to their new positions, and their fitness values are calculated. As shown, the first particle became much closer to the optimal solution than the other particles. As shown in this iteration, the velocities of all particles are not zero; in other words, the particles will be moved in the next iteration.

Step 4. Determination of the particle farthest from the center

Separate the particles into clusters with K-Means. Then identify the particle with the best fitness as the center and the one with the worst fitness as the farthest point in each cluster.

Step 5. Updating the location of the farthest particles

Update the location of the selected farthest particles in each cluster by using Equation-(5).

Step 6. Termination

Terminate the running if the condition is satisfied and output the g_{best} obtained in the last iteration; otherwise, go to Step 2.

Figure 3. The overview of the PSO-ISK algorithm.

1- Assign the parameter values for the PSO-ISK algorithm
2- Input the following parameters: swarm size (SS), max_{iter}, x_{max}, x_{min}, and $k_{(dim)} = \{k_1, k_2, \dots, k_N\}$
3- Initialize the following parameters: w_{max}, w, v_{max}, v_{min}, $c_{(i)} = \{c_1, c_2, \dots, c_N\}$
4- **for₁** each particle p_i in SS, do the following:
5- Initialize the particle's position (x_i) randomly
6- Initialize the particle's velocity (v_i) randomly
7- Calculate the first position $(f(x_i))$ using Equation-(1)
8- Set the particle's best position $(p_{i,best})$ to the current position
9- **if₁** $f(p_{i,best}) < f(g_{best})$, then set $g_{best} = p_{i,best}$
10- **endif₁**
11- **for₂** $iter = (1 \text{ to } max_{iter})$, do the following:
12- **for₃** each particle p_i in SS, do the following:
13- Calculate v_i using Equation-(2)
14- Calculate the new position $(f(x_i))$ by Equation-(3)
15- **if₂** $f(x_i) < f(p_{i,best})$, then set $p_{i,best} = x_i$
16- **elseif₂** $f(p_{i,best}) < f(g_{best})$, then set $g_{best} = p_{i,best}$
17- **endif₂**
18- Choose a particle for the cluster centers
19- **for₄** each particle dim, do the following:
20- Using Equation-(5), calculate the distance between each particle and the cluster centers (x_i)
21- Based on the Euclidean similarity measure, assign the particle to the closest central group $c_{(i)} = \{c_1, c_2, \dots, c_N\}$
22- Apply the K-Means algorithm to each group
23- Choose the new cluster center with Equation-(6).
24- **endfor₄**
25- Update the position of the farthest particle by Equation-(7).
26- **endfor₃**
27- **endfor₂**
28- Update w using Equation-(4).
29- **endfor₁**

Figure 4. The pseudo-code of the PSO-ISK algorithm.

3 The proposed approach

In this section, the benchmark test functions used in this study are first explained. Next, it is detailed how the value chosen to be used in the PSO-ISK algorithm is determined. Then the numerical results and discussions are given in detail. In this study, the PSO-ISK algorithm is tested by running on 16 benchmark test functions. To prove the accuracy of the results obtained with PSO-ISK and compare its performance, the comparisons and analyses were given in two experiments. In the first stage, the comparison of PSO-ISK with standard PSO is presented. In the second stage, a comparison with seven variants of PSO is given. PSO-ISK is developed in MATLAB R2017a. All implementations are run on a PC with an Intel Core (TM) i5-5200U processor and 8 GB of DDR3-1600 MHz (dual channel). The standard PSO algorithm proposed by Omran [26] is used for all experiments in this study since the basic PSO has some known weaknesses in the literature.

3.1 Benchmark test functions

PSO-ISK is applied to 16 benchmark test functions with different characteristics which consist of Unimodal Separable (US), Multimodal Separable (MS), Unimodal Non-Separable (UN), and Multimodal Non-Separable (MN). The mathematical definitions and details of the test problems are given in Table 1 [27]. The unimodal functions are used to test the capability of the exploitation process of an algorithm since these do not include local optima. On the other hand, the multimodal functions are used to evaluate the capability of the exploration process of an algorithm since these include multiple local optimal points at which it causes getting stuck.

3.2 Determining the optimal cluster number

The number of clusters is a user-defined control parameter of K-Means, and it is directly related to the performance of the algorithm. To see the effect of the different numbers of clusters, Firstly, PSO-ISK is applied on the benchmark test functions for six pre-runs to determine the optimal value of k-value (k) with values of 2, 3, 4, 5, 7, and 10 under the same conditions. In this study, PSO-ISK is run for 200000 FEs, the swarm size (SS). are 30 and 60, c_1 and c_2 are equal to 2, $rand_1$ and $rand_2$ are random numbers between 0 and 1 [0,1]. Table 2 presents the statistical results statement for all test functions, including the mean and standard deviation (Std.Dv) obtained from 30 independent runs. The best results have been highlighted in bold type within the tables. Additionally, to the phrase, Table-2 shows the results based on the k and SS. Table 2 analyzes high-low cluster densities and small-large SS.

The rank-based comparisons for the k-values are provided in Table 3. The order numbers in Table-3 indicate how several times the result is achieved, and the findings are displayed in Table 3 by SS, k-value, and (k). For instance, Table-3 shows 13 results for $k = 10$ and order number = 1. As a result, when $k = 10$, the optimal value is discovered 13 times. In other example, Table-3 displays 6 outcomes for $k = 3$ and order number = 5. As a result, when $k = 3$, the 5th rank result is discovered 6 times. It can be seen from Table-2 and Table-3 that the results of running with $k = 10$ obtained the best ranking for thirteen test functions (*F1, F2, F3, F4, F5, F6, F7, F8, F11, F12, F14, F15,* and *F16*). It also ranks as the second-best in the F13 test function. However, the run with $k = 7$ can achieve the best rank for nine test functions (*F3*, *F10*, *F11*, and *F12*). The other runs with $k = 3$, $k = 4$, and $k = 5$ each have only seven first ranks. In a population with a SS of 30, k is not considered to be bigger than 10. Each cluster includes about two particles when the swarm size is 30 and the number of clusters is 11 ($k = 11$). In other words, the strategy is incompatible when fewer than three particles remain in the cluster. Consequently, the swarm size is set to 60 in order to increase the number of clusters. Table-2 and Table-3 show that the results of $k = 20$, $k = 12$, and $k = 15$ yielded the worst rankings for 13 test functions. These results show that PSO-ISK can achieve more successful results in a shorter time, mostly when the number of clusters is 10 ($k = 10$). Therefore, the k value is chosen as ten in all experiments given in this study.

3.3 The comparison of PSO-ISK and SPSO2007

The standard PSO 2007 (SPSO) [26] and proposed PSO-ISK algorithms run 30 times with FEs under the same conditions for 16 benchmark test functions. All functions are operated for 2, 3, 4, and 5 dimensions in the independent runs. SPSO is run for this study, and its results are compared with those of PSO-ISK. The mean values of the obtained results, standard deviations (Std.Dv), and the best-obtained results in multiple runs are given in Table 4, and the best values for each test function are shown in bold. Considering the outputs in Table 4, the PSO-ISK algorithm generally achieves better values compared to the SPSO algorithm for all dimensions. This inference can also be seen in Table 5, which shows the numerical results of the superiority of PSO-ISK over SPSO. Table 5 lists the total number of test functions for each dimension where PSO-ISK is better, equal, or worse than SPSO according to the mean values. While PSO-ISK lags behind SPSO in only 1 or 2 test functions, for the rest of the 16 test functions, it appears to be able to achieve equivalent or better results. Therefore, it can be noted that PSO-ISK has remarkable robustness.

3.4 The comparison of PSO-ISK and PSO variants

In this experiment, the success of the PSO-ISK approach is evaluated with the results taken directly [10]-[28]. Thus, the PSO-ISK algorithm could be compared with the seven PSO variants mentioned in different studies. The details of the PSO variants whose results are used in this study can be accessed from the related publications, and the PSO-ISK is compared with the results obtained from the literature. These algorithms are LFPSO [10], CLPSO [12], HPSO-TVAC [13], FIPSO [14], SPSO-40 [15], LPSO [15], and DMS-PSO [16]. To make a fair comparison, all PSO variants have the same values for the control parameters. The SS for all algorithms is determined to be 40, and the dimension of the problems are taken to be 30 as in previous studies. The values in Table 6 indicate the statistical results of 25 independent runs of all algorithms for eight test functions. The termination condition is 200000 FEs. Table 6 also includes the rank value of the related algorithm with mean and standard deviation (Std.Dv). The rank value indicates the degree of the obtained result by an algorithm, from best (1) to worst (8) among the eight PSO-variants. Table 7 also ranks the algorithms collectively according to the average of their rank values in all test problems. According to Table 7, PSO-ISK is the most successful algorithm for all functions except Rastrigin and Griewank. In other words, PSO-ISK can achieve the best values for Ackley, Penaltlized1, Penaltlized2, Schwefel2.22, Sphere, and Rosenbrock. The results show that the PSO-ISK modification for the PSO algorithm provides a highly effective improvement. Based on Table-7, PSO-ISK takes first place with a mean rank of 2.125, while LFPSO takes second place among 8 variants with a mean rank of 2.875.

Table 1. Test functions.

No.	Function	Dimention	Characteristic	Formulation
F1	Ackley	[−32, 32]	MN	$f_1(\vec{x}) = -20exp\left\{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right\} - exp\left\{\frac{1}{D}\sum_{i=1}^{D} cos(2\pi x_i)\right\} + 20 + e$
F2	Penalized1	[−50, 50]	MN	$f_2(\vec{x}) = \frac{\pi}{D}\left\{10sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i-1)^2[1+10sin^2(\pi y_{i+1})] + (y_D-1)^2\right\} + \sum_{i=1}^{D} u(x_i,10,100,4)$ $y_i = 1+\frac{1}{4}(x_i+1)u_{x_i,a,k,m} = \begin{cases} k(x_i-a)^m & x_i > a \\ 0 & -a \leqslant x_i \leqslant a \\ k(x_i-a)^m & x_i < -a \end{cases}$
F3	Penalized2	[−50, 50]	MN	$f_3(\vec{x}) = \frac{1}{10}\left\{\sin^2(\pi x_1) + \sum_{i=1}^{D-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})] + (x_n-1)^2[1+\sin^2(2\pi x_{i+1})]\right\} + \sum_{i=1}^{D} u(x_i,5,100,4)$
F4	Schwefel2.22	[-10, 10]	UN	$f_4(\vec{x}) = \sum_{i=1}^{D}\lvert x_i\rvert + \prod_{i=1}^{D}\lvert x_i\rvert$
F5	Sphere	[−100, 100]	US	$f_5(\vec{x}) = \sum_{i=1}^{D} x_i^2$
F6	Step	[−100, 100]	US	$f_6(\vec{x}) = \sum_{i=1}^{D}(\lfloor x_i + 0.5\rfloor)^2$
F7	Sum Square	[−100, 100]	US	$f_7(\vec{x}) = \sum_{i=1}^{D} i{x_i}^2$
F8	Rastrigin	[-5.12, 5.12]	MS	$f_8(\vec{x}) = \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i) + 10]$
F9	Rosenbrock	[−30, 30]	UN	$f_9(\vec{x}) = \sum_{i=1}^{D-1}[100(x_{i+1}-x_i^2)^2 + (x_i-1)^2]$
F10	Griewank	[-600, 600]	MN	$f_{10}(\vec{x}) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
F11	Levy	[−10, 10]	MN	$f_{11}(\vec{x}) = \sum_{i=1}^{D-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})] + \sin^2(3\pi x_1) + \lvert x_D-1\rvert[1+\sin^2(3\pi x_D)]$
F12	Schwefel2.21	[−100, 100]	UN	$f_{12}(\vec{x}) = max_i\{\lvert x_i\rvert, 1 \leqslant i \leqslant D\}$
F13	Schwefel2.26	[−500, 500]	UN	$f_{13}(\vec{x}) = 418.98288727243369 \times D - \sum_{i=1}^{D} x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$
F14	Alpine	[-10, 10]	MS	$f_{14}(\vec{x}) = \sum_{i=1}^{n}\lvert x_i \cdot \sin(x_i) + 0.1 \cdot x_i\rvert$
F15	Weierstrass	[−0.5, 0.5]	MN	$f_{15}(\vec{x}) = \sum_{i=1}^{D}\left(\sum_{k=0}^{k_{max}}\left[a^k\cos\left(2\pi b^k(x_i+0.5)\right)\right]\right) - D\sum_{k=0}^{k_{max}}[a^k\cos(2\pi b^k 0.5)]$ $a = 0.5, b = 3, k_{max} = 20$
F16	Elliptic	[−100, 100]	UN	$f_{16}(\vec{x}) = \sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}}x_i^2$

Table 2. Performance analysis of cluster numbers and swarm sizes.

No.		$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 7$	$k = 10$	$k = 12$	$k = 15$	$k = 20$
		$SS = 30$	$SS = 30$	$SS = 30$	$SS = 30$	$SS = 30$	$SS = 30$	$SS = 60$	$SS = 60$	$SS = 60$
F1	Mean	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**
	Std.Dv	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Best	8.8818E-16	8.8818E-16	8.8818E-16	8.8818E-16	8.8818E-16	8.8818E-16	8.8818E-16	8.8818E-16	8.8818E-16
F2	Mean	**2.3558E-31**	**2.3558E-31**	**2.3558E-31**	**2.3558E-31**	**2.3558E-31**	**2.3558E-31**	**2.3558E-31**	**2.3558E-31**	**2.3558E-31**
	Std.Dv	4.4539E-47	4.4539E-47	4.4539E-47	4.4539E-47	4.4539E-47	4.4539E-47	4.4539E-47	4.4539E-47	4.4539E-47
	Best	2.3558E-31	2.3558E-31	2.3558E-31	2.3558E-31	2.3558E-31	2.3558E-31	2.3558E-31	2.3558E-31	2.3558E-31
F3	Mean	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**
	Std.Dv	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Best	1.4998E-33	1.4998E-33	1.4998E-33	1.4998E-33	1.4998E-33	1.4998E-33	1.4998E-33	1.4998E-33	1.4998E-33
F4	Mean	2.7281E-27	4.8439E-28	5.5544E-29	5.6389E-30	4.9807E-32	**1.9941E-45**	6.5348E-18	6.5758E-20	6.1567E-17
	Std.Dv	3.2769E-27	6.7514E-28	7.0531E-29	1.3038E-29	1.8419E-31	4.3484E-45	6.1422E-15	3.1234E-12	5.1640E-19
	Best	2.8495E-29	3.7504E-30	1.7019E-30	9.5295E-32	1.0091E-34	1.2954E-50	1.4852E-20	1.2561E-15	5.5122E-22
F5	Mean	9.3904E-52	2.5201E-53	7.9243E-55	6.8418E-57	8.1015E-62	**1.9582E-85**	6. 2452E-33	5.9028E-32	7.9243E-30
	Std.Dv	1.5492E-51	4.0619E-53	2.6339E-54	1.7647E-56	3.0271E-61	1.0726E-84	2.1426E-29	5.6289E-33	3.2515E-34
	Best	1.6759E-55	1.3062E-57	3.1839E-59	1.7495E-61	3.7070E-67	1.3331E-97	5.52254E-29	9.2241E-32	3.1041E-29
F6	Mean	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Std.Dv	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Best	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F7	Mean	1.6011E-53	8.3317E-55	9.0289E-57	3.2120E-57	2.6249E-63	**2.2827E-91**	7.0289E-50	8.1089E-51	9.1385E-52
	Std.Dv	3.1071E-53	2.6006E-54	2.5998E-56	1.6543E-56	8.7422E-63	4.9407E-91	2.5254E-54	2.5918E-54	2.1418E-55
	Best	9.3682E-57	5.9574E-59	1.2242E-60	1.0902E-62	4.9563E-69	6.0112E-101	1.2356E-50	1.2572E-50	1.9247E-50
F8	Mean	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Std.Dv	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Best	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F9	Mean	1.8181E-11	5.9853E-11	1.5745E-13	6.9601E-14	**1.1672E-14**	1.2733E-12	2.5498E-05	2.1045E-07	1.5556E-08
	Std.Dv	3.2351E-11	3.0607E-10	3.5254E-13	1.9890E-13	3.9630E-14	6.2326E-12	2.5554E-03	3.5254E-10	3.1235E-11
	Best	9.7255E-15	3.1785E-17	3.8755E-17	6.4705E-21	7.3066E-22	1.3449E-25	1.2451E-06	3.8755E-07	3.7415E-12
F10	Mean	**3.6476E-04**	1.3000E-03	1.7000E-03	5.5004E-04	1.8000E-03	3.3000E-03	6.1332E-02	8.7301E-01	2.2891E-02
	Std.Dv	1.4000E-03	2.8000E-03	3.1000E-03	1.9000E-03	3.4000E-03	4.7000E-03	3.2498E-02	3.3678E-03	3.1450E-03
	Best	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F11	Mean	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**
	Std.Dv	6.6809E-47	6.6809E-47	6.6809E-47	6.6809E-47	6.6809E-47	6.6809E-47	6.6809E-47	6.6809E-47	6.6809E-47
	Best	1.3498E-31	1.3498E-31	1.3498E-31	1.3498E-31	1.3498E-31	1.3498E-31	1.3498E-31	1.3498E-31	1.3498E-31
F12	Mean	3.4162E-26	3.1238E-27	8.3526E-28	8.2447E-29	2.0339E-31	**7.3070E-45**	6.3526E-24	7.1236E-23	2.3871E-21
	Std.Dv	5.2158E-26	4.5494E-27	1.0814E-27	1.6046E-28	6.2809E-31	2.5848E-44	5.6803E-17	2.5634E-19	5.7874E-17
	Best	1.3391E-27	1.1122E-29	2.5563E-29	1.9409E-31	2.2001E-33	1.4726E-48	3.5453E-19	1.5122E-19	7.5576E-29
F13	Mean	-8.778E+02	-1.003E+03	-1.143E+03	-1.157E+03	**-1.507E+03**	-1.235E+03	-2.563E+03	-1.128E+03	-1.843E+03
	Std.Dv	2.3739E+02	2.5771E+02	3.4093E+02	4.4529E+02	2.7453E+02	4.8710E+02	2.4577E+02	3.4745E+02	2.4093E+02
	Best	-1.484E+03	-1.510E+03	-1.771E+03	-1.667E+03	-2.249E+03	-1.918E+03	-1.771E+03	-1.576E+03	-1.771E+03
F14	Mean	3.7954E-07	9.3134E-08	2.5944E-07	5.6892E-07	1.1472E-16	**8.9730E-25**	7.5944E-07	1.5563E-07	3.5944E-07
	Std.Dv	1.5075E-06	5.0161E-07	1.1170E-06	1.8935E-06	6.2836E-16	4.7045E-24	4.1170E-06	2.1935E-06	4.6808E-06
	Best	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F15	Mean	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Std.Dv	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Best	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F16	Mean	2.3569E-46	1.6864E-48	1.0218E-49	1.8531E-52	1.7582E-57	**9.4375E-81**	1.0283E-42	1.1618E-43	1.1159E-41
	Std.Dv	8.9929E-46	5.4810E-48	4.7154E-49	5.6808E-52	4.7069E-57	5.1651E-80	5.8854E-49	5.5169E-49	5.8074E-49
	Best	6.2330E-51	5.6229E-53	1.8914E-55	1.8607E-56	6.9800E-63	1.8083E-94	1.8935E-55	1.9880E-55	1.8563E-55

Table 3. The comparisons of the ranks of the results with respect to k values.

		Order Numbers								
SS	k	1	2	3	4	5	6	7	8	9
30	2	8	0	0	0	2	6	0	0	0
30	3	7	0	2	0	6	1	0	0	0
30	4	7	0	1	8	0	0	0	0	0
30	5	7	2	6	0	0	1	0	0	0
30	7	9	6	0	0	1	0	0	0	0
30	10	13	1	0	1	0	1	0	0	0
60	12	7	0	0	0	0	0	4	3	2
60	15	7	0	0	0	0	0	3	4	2
60	20	7	0	0	0	0	0	2	2	5

Table 4. Experimental results of PSO-ISK and SPSO.

No.		$D=2$		$D=3$		$D=4$		$D=5$	
		PSO-ISK	SPSO	PSO-ISK	SPSO	PSO-ISK	SPSO	PSO-ISK	SPSO
F1	Mean	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	1.0066E-15	**8.8818E-16**	1.3619E-15
	Std.Dv	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	6.4863E-16	0.000E+00	1.2283E-15
	Best	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**
F2	Mean	**2.3558E-31**	**2.3558E-31**	**1.5705E-31**	**1.5705E-31**	**1.1779E-31**	**1.1779E-31**	**9.4233E-32**	**9.4233E-32**
	Std.Dv	4.4539E-47	4.4539E-47	6.6809E-47	6.6809E-47	2.2270E-47	2.2270E-47	3.3404E-47	3.3404E-47
	Best	**2.3558E-31**	**2.3558E-31**	**1.5705E-31**	**1.5705E-31**	**1.1779E-31**	**1.1779E-31**	**9.4233E-32**	**9.4233E-32**
F3	Mean	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**
	Std.Dv	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Best	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**	**1.4998E-33**
F4	Mean	**4.3749E-24**	4.3766E-23	**2.7739E-32**	2.2949E-30	**1.9604E-39**	4.4524E-37	**6.9479E-45**	1.6874E-42
	Std.Dv	6.3687E-24	6.1939E-23	4.8375E-32	1.9356E-30	2.1818E-39	3.3483E-37	9.1952E-45	5.0311E-42
	Best	**1.0547E-25**	2.1405E-24	**2.2733E-33**	2.0168E-31	**1.8073E-40**	2.3476E-38	**3.8004E-46**	6.9861E-44
F5	Mean	**4.2508E-45**	5.6123E-43	**6.8775E-62**	3.5694E-58	**2.1270E-75**	6.2298E-71	**3.5208E-87**	4.5479E-82
	Std.Dv	1.1077E-44	1.2871E-42	1.4487E-61	6.6597E-58	8.7209E-75	1.9580E-70	8.5092E-87	9.9633E-82
	Best	**2.7873E-48**	1.8630E-46	**2.7389E-64**	1.5468E-60	**2.9610E-78**	9.4018E-73	**4.4471E-89**	5.9303E-85
F6	Mean	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Std.Dv	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Best	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
F7	Mean	**1.2001E-46**	3.4455E-45	**8.0768E-64**	1.2565E-59	**2.5996E-77**	1.1334E-72	**1.4407E-88**	1.8118E-83
	Std.Dv	4.0789E-46	6.7876E-45	1.5188E-63	2.9314E-59	5.7645E-77	3.2176E-72	3.2367E-88	3.9677E-83
	Best	**5.2497E-50**	2.4111E-48	**5.3344E-66**	4.0227E-62	**6.8847E-80**	1.9770E-76	**3.6587E-91**	1.4291E-85
F8	Mean	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	3.3200E-02	**0.000E+00**	6.6300E-02
	Std.Dv	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.8170E-01	0.000E+00	3.6330E-01
	Best	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
F9	Mean	2.4272E-14	**3.2465E-15**	3.6807E-05	**1.3008E-05**	1.2000E-03	**9.5862E-04**	3.9750E-01	**3.7000E-03**
	Std.Dv	1.0694E-13	9.7337E-15	7.0469E-05	2.2182E-05	6.1715E-04	3.8606E-04	1.2031E+00	4.8000E-03
	Best	1.3079E-18	**3.0321E-19**	3.1007E-08	**2.6281E-09**	7.7701E-05	**1.3200E-06**	**1.1000E-03**	1.4000E-03
F10	Mean	**1.6198E-05**	2.4889E-04	**3.1000E-03**	4.0000E-03	**8.4000E-03**	1.1600E-02	**1.0100E-02**	2.0900E-02
	Std.Dv	1.1000E-03	1.3000E-03	3.9000E-03	4.3000E-03	5.9000E-03	9.7000E-03	3.3000E-03	1.4400E-02
	Best	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
F11	Mean	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**
	Std.Dv	6.6809E-47	6.6809E-47	6.6809E-47	6.6809E-47	6.6809E-47	6.6809E-47	6.6809E-47	6.6809E-47
	Best	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**	**1.3498E-31**
F12	Mean	**4.3788E-23**	3.3364E-22	**5.3633E-31**	4.7849E-29	**1.0159E-36**	1.8177E-34	**5.5303E-41**	1.1451E-38
	Std.Dv	6.6221E-23	2.9324E-22	5.3548E-31	5.2987E-29	1.6594E-36	2.5375E-34	5.4592E-41	1.0219E-38
	Best	**2.5663E-24**	6.1440E-24	**5.6450E-32**	4.0340E-30	**9.0958E-38**	4.1100E-36	**5.8581E-42**	1.1972E-39
F13	Mean	**-1.3834E+03**	3.9479E+00	**-1.4008E+03**	1.9740E+01	**-1.3527E+03**	3.9479E+01	**-1.3012E+03**	8.2964E+01
	Std.Dv	2.0234E+01	2.1624E+01	3.0851E+01	4.4894E+01	5.0740E+01	5.6787E+01	9.6430E+01	9.9169E+01
	Best	**-2.1097E+03**	-2.2737E-13	**-1.9583E+03**	-4.5475E-13	**-2.1391E+03**	-4.5475E-13	**-2.3650E+03**	-4.5475E-13
F14	Mean	**2.0354E-17**	1.5153E-07	**1.7006E-06**	1.8626E-05	1.3670E-06	**1.1241E-33**	**2.0354E-17**	**2.0354E-17**
	Std.Dv	1.1148E-16	6.3352E-07	9.2609E-06	5.8837E-05	7.4689E-06	5.7849E-33	1.1148E-16	1.1148E-16
	Best	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**5.3649E-55**	**5.3649E-55**
F15	Mean	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Std.Dv	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Best	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
F16	Mean	2.2583E-06	**1.4711E-37**	**3.2974E-58**	2.0298E-53	**2.3592E-72**	1.5357E-66	**0.000E+00**	3.2340E-77
	Std.Dv	8.7048E-06	4.5894E-37	6.0140E-58	3.3541E-53	1.0278E-71	3.6323E-66	0.000E+00	1.0896E-76
	Best	**0.000E+00**	7.4520E-41	**7.5932E-61**	5.4273E-56	**3.1786E-75**	1.5154E-68	**0.000E+00**	3.9712E-80

Table 5. Numerical comparisons of the results of PSO-ISK against SPSO for mean values.

	$D = 2$	$D = 3$	$D = 4$	$D = 5$
Better	7	8	9	9
Equal	7	7	5	6
Worse	2	1	2	1

Table 6. Comparative results of the PSO-ISK with PSO-variants.

Algorithm		F1	F2	F3	F4	F5	F8	F9	F10
PSO-ISK	Mean	**7.57E-15**	**1.57E-32**	**1.50E-33**	**2.87E-55**	**1.39E-98**	3.99E+01	**9.28E+00**	2.66E-03
	Std.Dv	**1.18E-15**	**5.59E-48**	**0.00E+00**	**3.26E-55**	**2.00E-98**	1.55E+01	**6.66E+00**	4.02E-03
	Rank	**1**	**1**	**1**	**1**	**1**	6	**1**	5
CLPSO	Mean	3.66E-07	6.45E-14	1.25E-12	2.51E-08	1.58E-12	**9.09E-05**	1.14E+01	9.02E-09
	Std.Dv	7.57E-08	3.70E-14	9.45E-13	5.84E-09	7.70E-13	**1.25E-04**	9.85E+00	8.57E-09
	Rank	7	7	6	7	8	**1**	2	3
HPSO-TVAC	Mean	7.29E-14	2.71E-29	2.79E-28	9.03E-20	2.83E-33	9.43E+00	2.39E+01	9.75E-03
	Std.Dv	3.00E-14	1.88E-29	2.18E-28	9.58E-20	3.19E-33	3.48E+00	2.65E+01	8.33E-03
	Rank	4	4	3	3	3	3	5	8
FIPSO	Mean	2.33E-07	1.96E-15	2.70E-14	2.76E-08	2.42E-13	6.51E+01	2.51E+01	9.01E-12
	Std.Dv	7.19E-08	1.11E-15	1.57E-14	9.04E-09	1.73E-13	1.34E+01	5.10E-01	1.84E-11
	Rank	6	6	4	8	7	8	6	2
SPSO-40	Mean	3.73E-02	7.47E-02	1.76E-03	1.74E-53	2.29E-96	4.10E+01	1.35E+01	7.48E-03
	Std.Dv	1.90E-01	3.11E+00	4.11E-03	1.58E-53	9.48E-96	1.11E+01	1.46E+01	1.25E-02
	Rank	8	8	7	2	2	7	3	7
LPSO	Mean	8.20E-08	8.10E-16	3.26E-13	1.70E-10	3.34E-14	3.51E+01	2.81E+01	1.53E-03
	Std.Dv	6.73E-08	1.07E-15	3.70E-13	1.39E-10	5.39E-14	6.89E+00	2.18E+01	4.32E-03
	Rank	5	5	5	6	6	5	7	4
DMS-PSO	Mean	1.84E-14	2.51E-30	2.64E-03	1.57E-18	2.65E-31	2.72E+01	4.16E+01	6.21E-03
	Std.Dv	4.35E-15	1.02E-29	4.79E-03	3.79E-18	6.25E-31	6.02E+00	3.03E+01	8.14E-03
	Rank	3	3	8	4	4	4	8	6
LFPSO	Mean	1.68E-14	4.67E-31	1.51E-28	2.64E-17	4.69E-31	4.54E+00	2.38E+01	**8.14E-17**
	Std.Dv	4.84E-15	9.01E-31	8.00E-28	6.92E-17	2.50E-30	1.03E+01	3.17E-01	**4.46E-16**
	Rank	2	2	2	5	5	2	4	**1**

Table 7. The comparisons of the ranks of the results.

	PSO-ISK	CLPSO	HPSO-TVAC	FIPSO	SPSO-40	LPSO	DMS-PSO	LFPSO
Mean Rank	2.125	5.125	4.125	5.875	5.5	5.375	5	2.875
Final Rank	1	5	3	8	7	6	4	2

4 Conclusion

Although PSO is one of the most efficient SI algorithms, it has the problem of getting stuck at local minimums due to early convergence and weak global search capability. In this study, the K-Means clustering method is inserted into the operating routine of PSO. The proposed method is named Particle Swarm Optimization with a New Intensification Strategy Based on K-Means (PSO-ISK). PSO-ISK aims to improve the intensification of the swarm by dividing the particles into several clusters. The total quality of the swarm is increased by improving the weakest particles in each cluster. The comparison between the SPSO2007 and PSO-ISK is conducted on 16 benchmark test functions with 2, 3, 4, and 5 dimensions. For all dimensions and almost all benchmark test functions, PSO-ISK obtained better results than SPSO. Furthermore, the performance of the PSO-ISK algorithm was analyzed by comparing seven PSO variants. The experimental results proved that PSO-ISK is more robust in most of the test functions, and it generally achieves better results. Until now, no such strategy has been proposed to improve PSO-ISK will be adapted in the future with metaheuristics based on swarm intelligence such as GWO and KHO. Because enhancing performance by improving the worst results is one of the most important components of achieving success.

5 Author contribution statements

In the scope of this study, the Tahir SAĞ in the formation of the idea, the design and the writing of codes; Ayşegül İHSAN in the assessment of obtained results, supplying the used and examining the results other than the literature review.

6 Ethics committee approval and conflict of interest statement

There is no need to obtain permission from the ethics committee for the article prepared. There is no conflict of interest with any person/institution in the article prepared.

7 References

[1] MiarNaeimi F, Azizyan G, Rashki M. "Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems". *Knowledge-Based Systems*, 213, 1-17, 2021.

[2] Chou JS, Truong DN. "A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean". *Applied Mathematics and Computation*, 389, 1-47, 2021.

[3] Ahmadianfar I, Bozorg-Haddad O, Chu X. "Gradient-based Optimizer: A new metaheuristic optimization algorithm". *Information Sciences,* 540, 131-159, 2020.

[4] Askari Q, Younas I, Saeed M. "Political optimizer: A novel socio-inspired meta-heuristic for global optimization". *Knowledge-Based Systems,* 195, 1-25, 2020.

[5] Shadravan S, Naji HR, Bardsiri VK. "The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems". *Engineering Applications of Artificial Intelligence,* 80, 20-34, 2019.

[6] Arora S, Singh S. "Butterfly optimization algorithm: a novel approach for global optimization". *Soft Computing,* 23(3), 715-734, 2019.

[7] Kiran MS. "TSA: Tree-seed algorithm for continuous optimization". *Expert Systems with Applications,* 42(19), 6686-6698, 2015.

[8] James JQ, Li VO. "A social spider algorithm for global optimization". *Applied Soft Computing,* 30, 614-627, 2015.

[9] Kennedy J, Eberhart R. "Particle swarm optimization". *Proceedings of ICNN'95-International Conference on Neural Networks*, Perth, WA, Australia, 27 November-01 December 1995.

[10] Haklı H, Uğuz H. "A novel particle swarm optimization algorithm with levy flight". *Applied Soft Computing,* 23, 333-345, 2014.

[11] Lei L, Min X, Xiaokui L. "Research on hybrid PSO algorithm with appended intensification and diversification". *Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)*, Shenyang, China, 20-22 December 2013.

[12] Liang, JJ, Qin AK, Suganthan PN, Baskar S. "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions". *IEEE Transactions on Evolutionary Computation*, 10(3), 281-295, 2006.

[13] Ratnaweera A, Halgamuge SK, Watson HC. "Self-organizing hierarchical Particle Swarm Optimizer with Time-varying acceleration coefficients". *IEEE Transactions on Evolutionary Computation,* 8(3), 240-255, 2004.

[14] Mendes R, Kennedy J, Neves J, "The fully informed Particle Swarm: simpler, maybe better". *IEEE Transactions on Evolutionary Computation,* 8(3), 204-210, 2004.

[15] Kennedy J, Mendes R. "Population structure and particle swarm performance". *Proceedings of the 2002 Congress on Evolutionary Computation CEC'02, (Cat. No.02TH8600)*, Honolulu, HI, USA, 12-17 May 2002.

[16] Liang JJ, Suganthan PN. "Dynamic multi-swarm Particle Swarm Optimizer". *Proceedings 2005 IEEE Swarm Intelligence Symposium,* Pasadena, CA, USA, 08-10 June 2005.

[17] Solaiman BF, Sheta A. "Energy optimization in wireless sensor networks using a hybrid K-means PSO clustering algorithm". *Turkish Journal of Electrical Engineering & Computer Sciences*, 24(4), 2679-2695, 2016.

[18] Gao H, Li Y, Kabalyants P, Xu H, Martinez-Bejar R. "A novel hybrid PSO-K-Means clustering algorithm using gaussian estimation of distribution method and Lévy Flight". *IEEE Access,* 8, 122848-122863, 2020.

[19] Mahajan M, Kumar S, Pant B. "Prediction of environmental pollution using hybrid PSO-K-Means approach". *International Journal of E-Health and Medical Communications (IJEHMC),* 12(2), 65-76, 2021.

[20] Sun Y, Liu G, Zheng D, Zou H, Liu Z, Liu J. "A self-adaptive Particle Swarm Optimization based K-means (SAPSO-K) clustering method to evaluate fabric tactile comfort". *The Journal of the Textile Institute,* 4(1), 1-12, 2021.

[21] Younus ZS, Mohamad D, Saba T, Alkawaz MH, Rehman A, Al-Rodhaan M, Al-Dhelaan A. "Content-based image retrieval using PSO and K-means clustering algorithm". *Arabian Journal of Geosciences,* 8(8), 6211-6224, 2015.

[22] Liu S, Zou Y. "An improved hybrid clustering algorithm based on Particle Swarm Optimization and K-means''. *IOP Conference Series: Materials Science and Engineering*, Singapore, 27-29 February 2020.

[23] Jamali-Dinan SS, Soltanian-Zadeh H, Bowyer SM, Almohri H, Dehghani H, Elisevich K, Nazem-Zadeh MR. "A combination of particle swarm optimization and minkowski weighted K-Means clustering: application in lateralization of temporal lobe epilepsy". *Brain Topography*, 33(4), 519-532, 2020.

[24] Tarkhaneh O, Isazadeh A, Khamnei HJ. "A new hybrid strategy for data clustering using cuckoo search based on Mantegna Levy distribution, PSO and K-means". *International Journal of Computer Applications in Technology,* 58(2), 137-149, 2018.

[25] MacQueen, J. "Some methods for classification and analysis of multivariate observations". *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, Los Angeles, USA, 1 January 1967.

[26] Particle Swarm Central. ''SPSO 2007 Matlab''. http://www.particleswarm.info/Programs.html (07.07.2020).

[27] Karaboga D, Akay B. "A comparative study of artificial bee colony algorithm". *Applied mathematics and computation,* 214(1), 108-132, 2009.

[28] Zhan Z, Zhang J, Li Y, Shi Y. "Orthogonal learning particle swarm optimization". *IEEE Transactions on Evolutionary Computation*, 15(6), 832-847, 2010.