J Inf Process Syst, Vol.15, No.4, pp.986~1016, August 2019
https://doi.org/10.3745/JIPS.04.0131

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

# Mobile User Interface Pattern Clustering Using Improved Semi-Supervised Kernel Fuzzy Clustering Method

Wei Jia*,**, Qingyi Hua*, Minjun Zhang*, Rui Chen*, Xiang Ji*, and Bo Wang*,***

**Abstract**

Mobile user interface pattern (MUIP) is a kind of structured representation of interaction design knowledge. Several studies have suggested that MUIPs are a proven solution for recurring mobile interface design problems. To facilitate MUIP selection, an effective clustering method is required to discover hidden knowledge of pattern data set. In this paper, we employ the semi-supervised kernel fuzzy c-means clustering (SSKFCM) method to cluster MUIP data. In order to improve the performance of clustering, clustering parameters are optimized by utilizing the global optimization capability of particle swarm optimization (PSO) algorithm. Since the PSO algorithm is easily trapped in local optima, a novel PSO algorithm is presented in this paper. It combines an improved intuitionistic fuzzy entropy measure and a new population search strategy to enhance the population search capability and accelerate the convergence speed. Experimental results show the effectiveness and superiority of the proposed clustering method.

**Keywords**

Intuitionistic Fuzzy Entropy Measure, Mobile User Interface Pattern, Particle Swarm Optimization, Population Search Strategy, Semi-Supervised Kernel Fuzzy C-Means

## 1. Introduction

A mobile user interface pattern (MUIP) is a proven solution to a common recurring mobile interface development problem. Meaningful design information such as design problem, functional goal and solution, included in the MUIP, can increase reusability, efficiency, quality and modularization in interface development. It is more useful for developers to design high usability interface with short time and share their knowledge with others [1-3]. In recent years, many attempts have been made toward the pattern selection. Gomes et al. [4] proposed a case-based reasoning (CBR) method for selecting patterns. In [5], the ontology-based technique was employed to represent knowledge on patterns and a set of question-answer pairs was used for reasoning suitable patterns. Hasheminejad and Jalili [6] presented a pattern selection method based on the text classification approach. But to our knowledge, little work has been made for the MUIP data analysis. In fact, data analysis plays a significant role in determining the applicability of a MUIP, because it provides useful information for pattern selection. Especially when


Manuscript received January 11, 2018; first revision March 15, 2018; second revision May 8, 2018; accepted July 2, 2018.
**Corresponding Author:** Qingyi Hua (huaqy@nwu.edu.cn)
* School of Information Science and Technology, Northwest University, Xi'an, China (jia.401@163.com, huaqy@nwu.edu.cn, {fantast_831, nwuchenrui}@126.com, {jxiang1113, xiyouwangbo}@163.com)
** Xinhua College, Ningxia University, Yinchuan, China (jia.401@163.com)
*** School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an, China (xiyouwangbo@163.com)

there is a large supply of labeled data, unlabeled data and overlapping functional goals between MUIPs, data analysis can effectively extract hidden knowledge from the MUIP data set. Thus an effective data analysis method is needed that facilitates the MUIP selection in the design phase.

Semi-supervised clustering, as an important data analysis method, is widely used in many applications to solve real-world problems. It uses a few labeled samples to guide the clustering process. Various methods have been proposed in the semi-supervised clustering area. However, some clustering methods such as density-based spatial clustering of applications with noise (DBSCAN) [7] and graphics processing unit accelerated DBSCAN (GDBSCAN) [8] fail to discover overlapping clusters because they belong to non-overlapping clustering. Some other clustering methods such as balanced iterative reducing and clustering using hierarchies (BIRCH) [9] and clustering algorithm based on randomized search (CLARANS) [10] only work well for convex or spherical clusters of uniform size.


**Fig. 1.** Several overview plus data patterns.

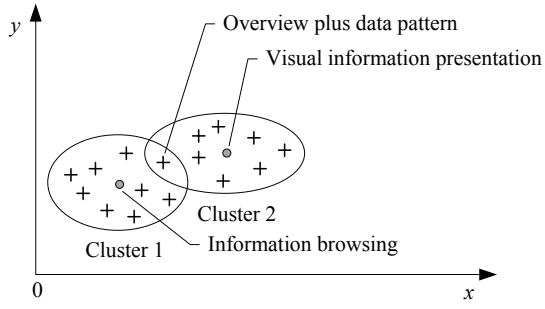


**Fig. 2.** MUIP clustering result.

In general, most MUIPs contain multiple sub functional goals that belong to different clusters. For instance, Fig. 1 shows several overview plus data patterns that use charts to summarize the most important information, and a table below with the detailed data. Two sub-functional goals are included in the overview plus data pattern. One is to browse the details of data and the other is to provide visual data. When information browsing and visual information presentation are selected as cluster centers, we can see from Fig. 2 that cluster 1 and cluster 2 cannot be reasonably distinguished due to overlapping area between clusters. Furthermore, researches on the functional goals of MUIPs show that there are a large

number of non-linear relationships between MUIPs [11]. In this case, MUIP data are non-linear and the distribution of MUIP data is irregular in space. Due to the above reasons, MUIP data are not linearly separable in input space and it is difficult for the above clustering methods to identify the arbitrary shaped clusters.

Recently, kernel fuzzy c-means (KFCM) clustering [12] has been received much attention in the semi-supervised clustering area [13]. The kernel function is utilized to perform a non-linear mapping from the original feature space to a high dimensional feature space. Due to the idea of kernel substitution, semi-supervised kernel fuzzy c-means (SSKFCM) clustering can identify non-spherical clusters and has a powerful ability to deal with non-linear data. In addition, fuzzy logic is used in SSKFCM clustering to determine the association of a data point to a cluster with different membership degree. Thus, if a MUIP simultaneously lies in more than one cluster, SSKFCM clustering can detect overlapping clusters. With the advantages above, the application of SSKFCM clustering to MUIP clustering seems natural and appropriate.

However, SSKFCM clustering is sensitive to the kernel parameter, the initial cluster centers and the number of clusters. Due to these defects, it is easily trapped in local minima. Many researchers treat this problem as an optimization problem and metaheuristic optimization algorithms such as genetic algorithm (GA) [14], simulated annealing (SA) [15], and particle swarm optimization (PSO) [16] are used to improve the clustering performance. Compared with other optimization algorithms, the PSO algorithm is more simple and easy to implement. Therefore, we pay more attention to the PSO algorithm in this study. In recent years, many PSO-based clustering methods have been proposed. Yu et al. [17] used the PSO algorithm to obtain the global optimal number of clusters. Mekhmoukh and Mokrani [18] improved the performance of clustering by utilizing the PSO algorithm to initialize the cluster centers. Liu et al. [19] presented an improved KFCM clustering method where, in the original KFCM clustering method, they employed the PSO algorithm for kernel parameter optimization.

The major drawback of PSO algorithm is that it easily falls into local optima and has slow convergence speed. Many variant PSO algorithms have been proposed in order to improve the performance of the PSO algorithm. Sedki and Ouazar [20] proposed a hybrid algorithm based on differential evolution (DE) [21] and PSO algorithm (DEPSO) which integrated DE and PSO algorithm at specified interval of iterations. In [22], a hybrid of chaotic optimization algorithm [23] and PSO algorithm (CPSO) was proposed to enhance the global optimal solution of the PSO algorithm. In [24], the authors proposed a PSO algorithm combined cellular automata (CA) [25] with PSO algorithm (CAPSO) to improve local search ability and global optimization ability. Each particle was considered as a cell and updated its current status with neighbors' states in the algorithm. Wang et al. [26] proposed an intuitionistic fuzzy entropy-based PSO algorithm (EPSO) which utilized intuitionistic fuzzy sets (IFSs) to describe particles and attempted to take advantage of intuitionistic fuzzy entropy [27] as an index to measure the state of the particle swarm. Their work had shown some preliminary results in improving the performance of the PSO algorithm through the entropy of IFSs. However, existing improved PSO algorithms still have some shortcomings in terms of measuring and maintaining the diversity of the population.

Inspired by the above discussions, we propose an improved SSKFCM clustering method for MUIP clustering. Firstly, a hybrid PSO algorithm combining intuitionistic fuzzy entropy, opposition-based learning (OL) [28], DE and PSO algorithm (EODPSO) is presented that aims to optimize clustering parameters. This work is motivated by the variant PSO algorithms. In EODPSO algorithm, an improved intuitionistic fuzzy entropy measure and a new population search strategy are proposed to obtain better

optimization result. Then, we propose a MUIP clustering method and use the EODPSO algorithm to optimize the clustering parameters. Finally, experiments using MUIP data set are reported and the results are compared to other relevant clustering methods.

In summary, the contributions of our work are summarized as follows:

(1) In order to reasonably measure the diversity of particle swarm, we present an improved entropy measure of IFSs that based on geometric interpretation of IFSs.

(2) We propose an improved PSO algorithm to enhance the exploration of the PSO algorithm. Specifically, the improved entropy measure is used to measure the diversity of particle swarm and a new population search strategy is used to strengthen the global convergence.

(3) We propose a SSKFCM clustering method for MUIP clustering where kernel parameter, initialize clustering centers and number of clusters are optimized through the improved PSO algorithm.

The remainder of this paper is organized as follows: Section 2 briefly introduces the SSKFCM clustering method and the PSO algorithm. Section 3 presents the details of the proposed method. We use numerical experiments to verify the performance of the proposed method in Section 4. Finally, a conclusion and future works are given in Section 5.

## 2. Preliminaries

### 2.1 Semi-Supervised Kernel Fuzzy C-Means Clustering

Fuzzy c-means (FCM) clustering [29] is a typical clustering method which allows points to belong to two or more clusters with different membership values, ranging from $[0,1]$. Let $Z=\{z_1,z_2,\cdots,z_n\}$, where $z_i \in \mathbb{R}^D$ is a data set of size $n$ in $D$-dimensional space. FCM clustering partitions this data set into $e'$ subsets. The aim of the FCM clustering is to minimize the objective function which is the generalized form of the least-squared errors function:

$$J(U,B)=\sum_{i=1}^{n}\sum_{j=1}^{e'}u_{i,j}^{m}\left\|z_i-b_j\right\|^2 \tag{1}$$

where $\left\|z_i-b_j\right\|$ denotes the Euclidean distance between $z_i$ and $b_j$, $m$ is the fuzzy controlling parameter and is generally chosen as 2, $u_{i,j}$ is the membership value of $z_i$ with the respect to cluster $j$ and $u_{i,j}$ must satisfy the following conditions:

(1) The membership value ranges between zero and one, that is, $u_{i,j}\in[0,1]$.

(2) The sum of the membership values for each data point must be one, that is, $\sum_{j=1}^{e'}u_{i,j}=1$.

(3) The sum of the all membership values in a cluster must be smaller than the number of data $n$, that is, $0<\sum_{i=1}^{n}u_{i,j}<n$.

In KFCM clustering, a kernel function is used to transfer the original feature space to a kernel-induced distance, and the Euclidean distance of FCM is replaced with kernel-induced distance. The KFCM clustering minimizes the following objective:

$$J(U,B)=\sum_{i=1}^{n}\sum_{j=1}^{e'}u_{i,j}^{m}\left\|\Phi(z_i)-\Phi(b_j)\right\|^2 \tag{2}$$

where $\Phi$ is a non-linear function mapping $z_i$ from the input space $Z$ to a new space $F$ with higher or even infinite dimensions.

The kernel-induced distance is calculated as:

$$\left\|\Phi(z_i)-\Phi(b_j)\right\| = K(z_i,z_i)-K(z_i,b_j)+K(b_j,b_j) \tag{3}$$

where $K(z_i,b_j)$ is the kernel function which is defined as the inner product in the new space $F$ with: $K(z_i,b_j)=\Phi(z_i)^T\Phi(b_j)$, for $z_i$ , $b_j$ in input space $Z$.

We can see that kernel function can be directly constructed in original input space without knowing the concrete form of $\Phi$ . Among the various kernel functions, Gaussian kernel is a typical kernel function which is expressed as follows:

$$K(z_i,b_j)=\exp\left(-\frac{\left\|z_i-b_j\right\|^2}{2\sigma^2}\right) \tag{4}$$

where $\sigma>0$ is the parameter of Gaussian kernel.

For Gaussian kernel, we have $K(z_i,z_i)=1$ . Thus, it is obvious that objective function of KFCM could be defined as:

$$J(U,B)=\sum_{i=1}^{n}\sum_{j=1}^{e'}u_{i,j}^{m}\left\|2-2K(z_i,b_j)\right\|^2 \tag{5}$$

By optimizing Eq. (5), the fuzzy partition matrix and cluster centers can be obtained as follows:

$$u_{i,j}=\frac{\left(1/\left(1-K(z_i,b_j)\right)\right)^{1/(m-1)}}{\sum_{k=1}^{e'}\left(1/\left(1-K(z_i,b_k)\right)\right)^{1/(m-1)}} \tag{6}$$

$$b_j=\frac{\sum_{i=1}^{n}u_{i,j}^{m}K(z_i,b_j)x_i}{\sum_{i=1}^{n}u_{i,j}^{m}K(z_i,b_j)} \tag{7}$$

By integrating supervised and unsupervised information into the objective function of KFCM, SSKFCM clustering method expands KFCM clustering into semi-supervised domain. This method takes advantage of supervised information, such as labeled data or pair-wise constraints, to guide the clustering process. The labeled and unlabeled data can be denoted in a whole matrix form as follows:

$$Z=\left\{z_1^{l'},\cdots,z_{n_{l'}}^{l'}\mid z_1^{u'},\cdots,z_{n_{u'}}^{u'}\right\}=Z^{l'}\cup Z^{u'} \tag{8}$$

where the superscript $l'$ and $u'$ are the labeled and unlabeled data respectively, $n_{l'}$ and $n_{u'}$ are the number of the labeled and unlabeled data respectively. The total number of data is $n_{l'}+n_{u'}$.

A matrix of the fuzzy c-partition of $Z$ in Eq.(8) is expressed as:

$$U=\left\{U^{l'}=\left\{u_{i,j}^{l'}\right\}\middle|U^{u'}=\left\{u_{i,j}^{u'}\right\}\right\} \tag{9}$$

where the value of $u_{i,j}^{l'}$ is known before and typically is set to 1 if the data $z_i$ is labeled with class *j*, and 0 otherwise.

The initial of set of cluster centers is obtained from $U^{l'}$ as follows:

$$b_j^0=\frac{\sum_{i=1}^{n_{l'}}\left(u_{i,j}^{l'}\right)^m z_i^{l'}}{\sum_{i=1}^{n_{l'}}\left(u_{i,j}^{l'}\right)^m} \tag{10}$$

Then, the membership $u_{ij}^{u'}$ in $U^{u'}$ is calculated:

$$u_{i,j}^{u'}=\frac{\left(1/\left(1-K\left(z_i^{u'},b_j\right)\right)\right)^{1/(m-1)}}{\sum_{k=1}^{e'}\left(1/\left(1-K\left(z_i^{u'},b_k\right)\right)\right)^{1/(m-1)}} \tag{11}$$

Finally, the cluster centers are written:

$$b_j=\frac{\sum_{i=1}^{n_{l'}}\left(u_{i,j}^{l'}\right)^m K\left(z_i^{l'},b_j\right)x_i^{l'}+\sum_{i=1}^{n_{u'}}\left(u_{i,j}^{u'}\right)^m K\left(z_i^{u'},b_j\right)x_i^{u'}}{\sum_{i=1}^{n_{l'}}\left(u_{i,j}^{l'}\right)^m K\left(z_i^{l'},b_j\right)+\sum_{i=1}^{n_{u'}}\left(u_{i,j}^{u'}\right)^m K\left(z_i^{u'},b_j\right)} \tag{12}$$

## 2.2 Particle Swarm Optimization

The aim of PSO algorithm is to solve optimization problems and avoid trapping into local minima. In PSO algorithm, a set of potential solutions is represented as particles that fly in *D*-dimensional space. For a search problem in a *D*-dimensional space, $p_i(t)$ is the best position that the $i^{\text{th}}$ particle has ever visited, $p_g(t)$ is the best position that the whole population have visited. The velocity of a particle $i$ is updated:

$$v_i(t+1)=wv_i(t)+c_1r_1\left(p_i(t)-y_i(t)\right)+c_2r_2\left(p_g(t)-y_i(t)\right) \tag{13}$$

where $y_i(t)$ and $v_i(t)$ are the position and the velocity of particle $i$ at iteration $t$, respectively, $w$ is inertia weight. It represents the inheritance of the next flight velocity from the current flight velocity. $c_1$ and $c_2$ are acceleration coefficients. They represent particle's self-learning capability and capability of learning from excellent companion respectively. In general, $c_1=c_2=2$. $r_1$ and $r_2$ are two independently uniformly distributed random variables with range $[0,1]$.

The position of a particle $i$ is updated as follows:

$$y_i(t+1)=y_i(t)+v_i(t) \tag{14}$$

Entropy of an IFS describes the fuzziness degree of the IFS. Atanassov [30] generalized the concept of fuzzy sets and defined the IFS as follows:

**DEFINITION 1:** *An IFS* $A$ *in a finite set* $X=\{x_i|i=1,2,\cdots,n\}$ *is given by:*

$$A=\{\langle x,\mu_A(x),\gamma_A(x)\rangle|x\in X\} \tag{15}$$

*where* $\mu_A: X\to[0,1]$ *is the membership function of* $A$ *and* $\gamma_A: X\to[0,1]$ *is the non-membership function of* $A$, *with the condition* $0\le\mu_A(x)+\gamma_A(x)\le 1$, $\forall x\in X$. *The numbers* $\mu_A(x)$ *and* $\gamma_A(x)$ *represent the membership degree and non-membership degree of the element* $x$ *to the set* $A$, *respectively.*

**DEFINITION 2:** *For each IFS* $A$ *in* $X$, *if*

$$\pi_A(x)=1-\mu_A(x)-\gamma_A(x) \tag{16}$$

*then* $\pi_A(x)$ *is called intuitionistic index of* $x$ *in* $A$. *It is a hesitancy degree of* $x$ *to* $A$, *and* $0\le\pi_A(x)\le 1$ *for* $\forall x\in X$.

**DEFINITION 3:** *The complementary set* $A^c$ *of* $A$ *is defined as*:

$$A^c=\{\langle x,\gamma_A(x),\mu_A(x)\rangle|x\in X\} \tag{17}$$

Wang et al. [26] utilized IFS to describe particles and taken the entropy of the IFS as the measurement of the state of particle swarm and the mutation parameter. Its main idea is that there is an effective radius $R_i'$ for the best position $p_i(t)$ of each particle $x_i$ at iteration $t$. If a particle is in the range of $R_i'$, then it gather near $p_i(t)$, otherwise, it does not gather near $p_i(t)$ and is considered as an isolated point. The related definition of entropy of IFSs in PSO algorithm is defined as follows:

**DEFINITION 4:** *At iteration* $t$, *the best position* $p_i(t)$ *of each particle is selected as the aggregation point.* $F_{k,i}'$ *is the distance function which calculates the distance from each particle to aggregation point* $p_i(t)$. *The effective radius of* $p_i(t)$ *is* $R_i'=m'\max(F_{k,i}')$, *where* $m'$ *is random value in range* $[0,0.5]$.

**DEFINITION 5:** *Let* $A'$ *is a particle swarm in a finite set* $X=\{x_i|i=1,2,\cdots,n\}$. *At iteration* $t$, *if* $F_{k,i}'\le R_i'$, *then the particle* $k$ *belongs to the effective radius of* $p_i(t)$ *and* $T_{k,i}'=T_{k,i}'+1$. *If* $F_{k,i}'>R_i'$, *then the particle* $k$ *is an isolated point and* $T_{x_i}''=T_{x_i}''+1$. *Then membership degree* $\mu_{A_{k,i}'}^t(x_i)=\frac{T_{k,i}'}{n}$, *intuitionistic index* $\pi_{A_{k,i}'}^t(x_i)=\frac{T_x''}{n}$ *and non-membership degree* $\gamma_{A_{k,i}'}^t(x_i)=1-\mu_{A_{k,i}'}^t(x_i)\ -\pi_{A_{k,i}'}^t(x_i)$, *where* $\mu_{A_{k,i}'}^t(x_i)$ *is the membership degree of* $p_i(t)$ *at iteration* $t$. $\pi_{A_{k,i}'}^t(x_i)$ *is the intuitionistic index of an isolated point.* $\gamma_{A_{k,i}'}^t(x_i)$ *is the non-membership degree of* $p_i(t)$ *at iteration* $t$.

# 3. The Proposed Method for Mobile User Interface Pattern Clustering

## 3.1 Improved Intuitionistic Fuzzy Entropy Measure

Many recent studies have been done on intuitionistic fuzzy entropy measure from different points of

view. Zeng and Li [31] discussed the relationship between similarity measure and entropy of IFSs. Their study showed that similarity measures and entropies of IFSs can be transformed to each other. Based on this point of view, they defined the entropy measure as:

$$E_{Zeng}(A)=\frac{1}{n}\sum_{i=1}^{n}\left|\mu_A(x_i)-\gamma_A(x_i)\right| \tag{18}$$

Li et al. [32] investigated the systematic transformation of the entropy into the similarity measure for IFSs and discussed the sufficient conditions of this transformation. Then, the entropy measure was defined as:

$$E_{Li}(A)=1-\frac{1}{2n}\sum_{i=1}^{n}\left(\left|\mu_A(x_i)-\gamma_A(x_i)\right|^3+\left|\mu_A(x_i)-\gamma_A(x_i)\right|\right) \tag{19}$$

Verma and Sharma [33] proposed an exponential entropy of IFSs based on the concept of exponential fuzzy entropy. The entropy measure was given by:

$$E_{Verma}(A)=\frac{1}{n\left(\sqrt{e}-1\right)}\sum_{i=1}^{n}\left[\theta e^{1-\theta}+(1-\theta)e^{\theta}-1\right] \tag{20}$$

in which $\theta=\left(\frac{\mu_A(x_i)+1-\gamma_A(x_i)}{2}\right)$.

In [27], an entropy measure for IFSs based on geometric interpretation was shown as:

$$E_{Wang}(A)=\frac{1}{n}\sum_{i=1}^{n}\frac{\min\left(\mu(x_i),\gamma(x_i)\right)+\pi_A(x_i)}{\max\left(\mu(x_i),\gamma(x_i)\right)+\pi_A(x_i)} \tag{21}$$

In [34], an entropy measure based on a trigonometric function was given by:

$$E_{Wei}(A)=\frac{1}{n}\sum_{i=1}^{n}\cos\frac{\mu_A(x_i)-\gamma_A(x_i)}{2\left(1+\pi_A(x_i)\right)}\pi \tag{22}$$

A different axiomatic definition of entropy for IFSs was given in [35], and the entropy measure was defined as:

$$E_{Gao}(A)=\frac{1}{n}\sum_{i=1}^{n}\frac{1-\left|\mu_A(x_i)-\gamma_A(x_i)\right|^2+\pi_A(x_i)}{2} \tag{23}$$

Liu and Ren [36] discussed the hesitation information of IFSs and given the entropy measure for IFSs as:

$$E_{Liu}(A)=\frac{1}{n}\sum_{i=1}^{n}\cos\frac{\left(\mu_A(x_i)-\gamma_A(x_i)\right)\left(1-\pi_A(x_i)\right)}{2}\pi \tag{24}$$

However, the existing entropy measures cannot reasonably measure the fuzziness degree of IFSs in some cases. To solve this problem, it is necessary to introduce an improved intuitionistic fuzzy entropy measure. In [37], the authors presented a geometric interpretation of an IFS and defined the distances between intuitionistic fuzzy sets. Inspired by this study, we propose an effective intuitionistic fuzzy

entropy measure based on geometric interpretation to overcome the disadvantages of existing entropy measures. The proposed entropy measure takes fully into account many sides, including intuitionistic index, membership degree, non-membership degree and relationships among them.

A geometric interpretation of an IFS is shown in Fig. 3, each element of an IFS $X$ is mapped into a point in an equilateral triangle $A'B'D'$ in their opinion. The triangle $A'B'C'$ is an orthogonal projection of the triangle $A'B'D'$. Segment $AB$ represents a fuzzy set described by two parameters $\mu$ and $\gamma$. All points which are above the segment $A'B'$ have a hesitancy margin greater than 0. The most undefined is point $D'$. When the hesitancy margin for $D'$ is equal to 1, it is difficult to judge if this point belongs or does not belong to the set. When the distance from $D'$ to $A'$ is equal to the distance to $B'$, the fuzziness degree for $D'$ is equal to 100%.

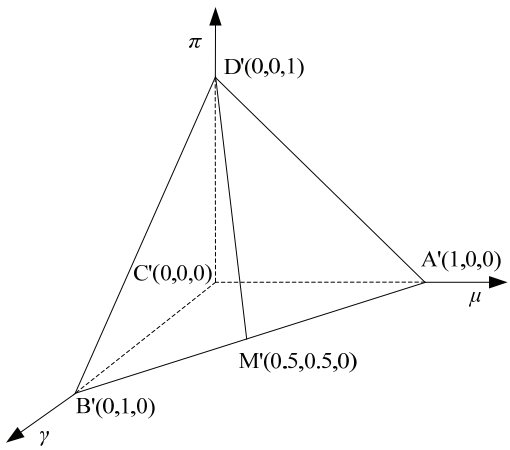


**Fig. 3.** A 3-dimensional representation of an IFS.

In Fig. 3, we draw a perpendicular line from point $D'$ on a line segment $A'B'$, two line intersect at point $M'$, and a perpendicular bisector $D'M'$ is obtained. Obviously, $\mu_A(x)$ equals $\gamma_A(x)$ on any point of $D'M'$, and triangle $D'B'M'$ is complementary with triangle $D'A'M'$. It is worth noticing that $E(A)$ is higher with higher $\pi_A(x)$ and is lower with higher $\mu_A(x)$ or $\gamma_A(x)$. From Fig. 3, we can safely come to the following conclusions about entropy of IFSs $E(A)$:

(1) $E(A)=1$, iff $\mu_A(x)=\gamma_A(x)=0$, $\pi_A(x)=1$ for $\forall x \in X$;

(2) $E(A)=0$, if $\mu_A(x)=1$, $\gamma_A(x)=\pi_A(x)=0$ or $\gamma_A(x)=1$, $\mu_A(x)=\pi_A(x)=0$ for $\forall x \in X$;

(3) $E(A)$ is only equal to $E(A^c)$ on $A'B'D'$;

(4) $E(A)$ becomes higher with the point of $D'M'$ approaches $D'$.

Based on the existing studies and conclusions above [37,38], we give an axiomatic definition of entropy measure of IFSs as follows:

**DEFINITION 6:** *Let* $X=\{x_i | i=1,2,\cdots,n\}$ *be a finite universe of discourse. A real function* $E$ : $IFS(X) \to [0,1]$ *is called an entropy measure of an* $IFS(X)$, *if* $E$ *has the following properties*:

(1) $E(A)=0$ *iff* $A$ *is a crisp set*;

(2) $E(A)=0$ *iff* $\mu_A(x_i)=\gamma_A(x_i)=0$ *for* $\forall x_i \in X$ ;

(3) $E(A)\le E(G)$ *if* $A$ *is less fuzzy than* $G$ , *i.e., when* $\pi_A(x_i)=\pi_G(x_i)$ , $\mu_A(x_i)\le\mu_G(x_i)\le\gamma_G(x_i)\le\gamma_A(x_i)$ *or* $\mu_A(x_i)\ge\mu_G(x_i)\ge\gamma_G(x_i)\ge\gamma_A(x_i)$ , *or when* $\pi_A(x_i)\le\pi_G(x_i)$, $|\mu_A(x_i)-\gamma_A(x_i)|=|\mu_G(x_i)-\gamma_G(x_i)|$ *for* $\forall x_i \in X$ ;

(4) $E(A)=E(A^c)$ .

Through above analysis, we develop an effective entropy measure as follows:

$$E(A)=\frac{1}{n}\sum_{i=1}^{n}\frac{2-2|\mu_A(x_i)-\gamma_A(x_i)|+\pi_A^{\,2}(x_i)}{3+\mu_A(x_i)+\gamma_A(x_i)+|\mu_A(x_i)-\gamma_A(x_i)|} \tag{25}$$

**THEOREM 1:** *Let* $X=\{x_i|i=1,2,\cdots,n\}$ *be a finite universe of discourse. A real function* $E$ : $IFS(X)\to[0,1]$ *defined by the Eq.* (25) *is an entropy measure of the IFS.*

***Proof*:** As an intuitionistic fuzzy entropy measure, it should satisfy the axioms 1–4 in Definition 6.

(1) When $A$ is a crisp set, we have $\mu_A(x_i)=1$ , $\gamma_A(x_i)=\pi_A(x_i)=0$ or $\gamma_A(x_i)=1$ , $\mu_A(x_i)=\pi_A(x_i)=0$ . Hence $E(A)=0$ . On the other hand, suppose $E(A)=0$ . Then $\frac{2-2|\mu_A(x_i)-\gamma_A(x_i)|+\pi_A^{\,2}(x_i)}{3+\mu_A(x_i)+\gamma_A(x_i)+|\mu_A(x_i)-\gamma_A(x_i)|}=0$ . So we have $2-2|\mu_A(x_i)-\gamma_A(x_i)|+\pi_A^{\,2}(x_i)=0$ . Since $0\le 2-2|\mu_A(x_i)-\gamma_A(x_i)|\le 2$ and $0\le\pi_A^{\,2}(x_i)\le 1$ , we then get $0\le 2-2|\mu_A(x_i)-\gamma_A(x_i)|=0$ and $\pi_A^{\,2}(x_i)=0$ . Thus we obtain $\mu_A(x_i)=1$, $\gamma_A(x_i)=\pi_A(x_i)=0$ or $\gamma_A(x_i)=1$, $\mu_A(x_i)=\pi_A(x_i)=0$ . Therefore $A$ is a crisp set.

(2) We now suppose $E(A)=1$. Then we get $\frac{2-2|\mu_A(x_i)-\gamma_A(x_i)|+\pi_A^{\,2}(x_i)}{3+\mu_A(x_i)+\gamma_A(x_i)+|\mu_A(x_i)-\gamma_A(x_i)|}=1$ . Thus we can obtain $1+\mu_A(x_i)+\gamma_A(x_i)+3|\mu_A(x_i)-\gamma_A(x_i)|=\pi_A^{\,2}(x_i)$ . Since $0\le\pi_A^{\,2}(x_i)\le 1$ and $1\le 1+\mu_A(x_i)+\gamma_A(x_i)+3|\mu_A(x_i)-\gamma_A(x_i)|$, we then get $\mu_A(x_i)=\gamma_A(x_i)=0$ . Let us suppose $\mu_A(x_i)=\gamma_A(x_i)=0$, we have $E(A)=1$ .

(3) When $\pi_A(x_i)=\pi_G(x_i)$ , and $\mu_A(x_i)\le\mu_G(x_i)\le\gamma_G(x_i)\le\gamma_A(x_i)$ or $\mu_A(x_i)\ge\mu_G(x_i)\ge\gamma_G(x_i)\ge\gamma_A(x_i)$ . So we get $1-|\mu_A(x_i)-\gamma_A(x_i)|\le 1-|\mu_G(x_i)-\gamma_G(x_i)|$ and $\mu_A(x_i)+\gamma_A(x_i)=\mu_G(x_i)+\gamma_G(x_i)$ . Then we can have $2-2|\mu_A(x_i)-\gamma_A(x_i)|+\pi_A^{\,2}(x_i)\le 2-2|\mu_G(x_i)-\gamma_G(x_i)|+\pi_G^{\,2}(x_i)$ and $3+\mu_A(x_i)+\gamma_A(x_i)+|\mu_A(x_i)-\gamma_A(x_i)|\ge 3+\mu_G(x_i)+\gamma_G(x_i)+|\mu_G(x_i)-\gamma_G(x_i)|$ . Thus $E(A)\le E(G)$ . When $\pi_A(x_i)\le\pi_G(x_i)$ and $|\mu_A(x_i)-\gamma_A(x_i)|=|\mu_G(x_i)-\gamma_G(x_i)|$, we obtain $\mu_A(x_i)+\gamma_A(x_i)\ge\mu_G(x_i)+\gamma_G(x_i)$. Then we can get $2-2|\mu_A(x_i)-\gamma_A(x_i)|+\pi_A^{\,2}(x_i)\le 2-2|\mu_G(x_i)-\gamma_G(x_i)|+\pi_G^{\,2}(x_i)$ and $3+\mu_A(x_i)+\gamma_A(x_i)+|\mu_A(x_i)-\gamma_A(x_i)|\ge 3+\mu_G(x_i)+\gamma_G(x_i)+|\mu_G(x_i)-\gamma_G(x_i)|$ . So $E(A)\le E(G)$ .

(4) It is clear that $A^c=\{\langle x_i,\gamma_A(x_i),\mu_A(x_i)\rangle|x_i\in X\}$ and $\pi_{A^c}(x_i)=\pi_A(x_i)$ . Then $E(A^c)=$

$\frac{1}{n}\sum_{i=1}^{n}\frac{2-2\left|\gamma_A(x_i)-\mu_A(x_i)\right|+\pi_A^{\,2}(x_i)}{3+\gamma_A(x_i)+\mu_A(x_i)+\left|\gamma_A(x_i)-\mu_A(x_i)\right|}$. Thus, we have $E(A^c)=E(A)$.

Hence, we complete the proof of Theorem 1.

In the following, we demonstrate the performance of $E(A)$ by the comparison of $E(A)$ and the existing intuitionistic fuzzy entropy measures.

Let us calculate the entropy for IFSs: $A_1=\{\langle x_i,0,0\rangle|x_i\in X\}$, $A_2=\{\langle x_i,0.5,0.5\rangle|x_i\in X\}$, $A_3=\{\langle x_i,0.5,0.4\rangle|x_i\in X\}$, $A_4=\{\langle x_i,0.2,0.1\rangle|x_i\in X\}$, $A_5=\{\langle x_i,0.3,0.3\rangle|x_i\in X\}$, $A_6=\{\langle x_i,0.4,0.4\rangle|x_i\in X\}$, $A_7=\{\langle x_i,0.3,0.2\rangle|x_i\in X\}$, $A_8=\{\langle x_i,0.4,0.3\rangle|x_i\in X\}$, $A_9=\{\langle x_i,0.2,0.6\rangle|x_i\in X\}$ and $A_{10}=\{\langle x_i,0,0.5\rangle|x_i\in X\}$. According to the geometric interpretation of IFSs, it is clear that the fuzziness degrees of these IFSs are different. Table 1 presents the calculation results of different entropy measures.

**Table 1.** Comparison of the different entropy measures

| **Entropy measure** | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ | $A_7$ | $A_8$ | $A_9$ | $A_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $E_{Wang}(A)$ | 1 | 1 | 0.8333 | 0.8889 | 1 | 1 | 0.875 | 0.8571 | 0.5 | 0.5 |
| $E_{Zeng}(A)$ | 1 | 1 | 0.9 | 0.9 | 1 | 1 | 0.9 | 0.9 | 0.6 | 0.5 |
| $E_{Li}(A)$ | 1 | 1 | 0.9495 | 0.9495 | 1 | 1 | 0.9495 | 0.9495 | 0.768 | 0.6875 |
| $E_{Verma}(A)$ | 1 | 1 | 0.9905 | 0.9905 | 1 | 1 | 0.9905 | 0.9905 | 0.8463 | 0.7588 |
| $E_{Wei}(A)$ | 1 | 1 | 0.9898 | 0.9957 | 1 | 1 | 0.9945 | 0.9927 | 0.8662 | 0.8662 |
| $E_{Gao}(A)$ | 1 | 0.5 | 0.5 | 0.74 | 0.58 | 0.52 | 0.62 | 0.54 | 0.44 | 0.5 |
| $E_{Liu}(A)$ | 1 | 1 | 1 | 0.9999 | 1 | 1 | 0.9999 | 0.9999 | 0.9995 | 0.998 |
| $E(A)$ | 1 | 0.5 | 0.4525 | 0.6735 | 0.6 | 0.5368 | 0.5694 | 0.4974 | 0.2952 | 0.3125 |

From Table 1, we find that $E_{Zeng}(A)$, $E_{Li}(A)$ and $E_{Verma}(A)$ cannot distinguish the fuzziness degrees of $A_3$, $A_4$, $A_7$ and $A_8$. Moreover, their calculation results show that the fuzziness degrees of $A_2$, $A_5$ and $A_6$ are equal to 1, which is obviously unreasonable. The reason is that they omit intuitionistic index and only consider the difference between membership degree and non-membership degree. Although $E_{Wang}(A)$, $E_{Wei}(A)$, $E_{Gao}(A)$ and $E_{Liu}(A)$ consider the intuitionistic index, they do not take full account of the relationship between known information and unknown information. For this reason, their calculation results do not clearly show the differences between these IFSs in terms of fuzziness and intuitionism.

By comparing the calculation results as listed in Table 1, the proposed entropy measure $E(A)$ can definitely distinguish the fuzziness degrees of these IFSs. Further, for $A_3$, $A_4$, $A_7$ and $A_8$, we can see that differences between the membership degrees and the non-membership degrees are the same. In this case, the greater value of $\pi_A(x_i)$, the fuzzier the corresponding element $A_i$. Thus uncertain information of $A_4$ is the biggest, uncertain information of $A_7$ is more than that of $A_8$, and uncertain information of $A_3$ is the least. The calculation results of the proposed entropy measure show that $E(A_3)<E(A_8)<E(A_7)<E(A_4)$, as we would expect. Hence, the proposed entropy measure is not only mathematically correct but also reasonable.

## 3.2 Improved Particle Swarm Optimization Algorithm

In this study, EODPSO is proposed in this study to optimize clustering parameters. To improve the performance of the PSO algorithm, we use a hybrid strategy to avoid local minima and accelerate convergence speed. The hybrid strategy contains the following three aspects.

(1) Chaotic opposition-based learning (COL) [39] is used to obtain a better initial population.
(2) CA is combined with PSO algorithm to enhance local search ability.
(3) We present a new population search strategy that combines OL and DE to extend the performance of global search.

### 3.2.1 Generation of initial population

A reasonable initial population plays a vital role in helping to avoid premature local optimization and accelerate convergence speed. COL is a population initialization method which combines chaotic systems and OL to obtain initial population. Thus COL is employed in EODPSO algorithm to generate an initial population.

In [39], the authors used the Sine map in their chaotic system. Nevertheless, the Sine map has two problems. Firstly, its chaotic range is limited to a finite interval. Secondly, the data distribution of output chaotic sequences is non-uniform. To avoid these problems, we use an improved 1-dimensional chaotic system [40], called Tent-Sine system, in COL to obtain better result. The Tent-Sine system uses the Tent and Sine maps as seed maps. The Tent-Sine system is defined as follows:

$$Q_{i+1}=\begin{cases}\mathrm{mod}\left(\dfrac{r'Q_i}{2}+\dfrac{(4-r')\left(\sin(\pi Q_i)\right)}{4},1\right) & Q_i<0.5\\ \mathrm{mod}\left(\dfrac{r'(1-Q_i)}{2}+\dfrac{(4-r')\left(\sin(\pi Q_i)\right)}{4},1\right) & Q_i\geq 0.5\end{cases} \tag{26}$$

where $Q_i$ is the output chaotic sequence, $r'$ is a parameter with range of $(0,4]$, mod is the modulo operation, $i$ is the iteration number, $\begin{cases}\dfrac{r'Q_i}{2} & Q_i<0.5\\ \dfrac{r'(1-Q_i)}{2} & Q_i\geq 0.5\end{cases}$ is Tent map, $\dfrac{(4-r')\left(\sin(\pi Q_i)\right)}{4}$ is Sine map.

At iteration $t$, the opposite search of particle $i$ is carried out as follows:

$$\overline{y}_{i,j}(t)=y_{\min,j}(t)+y_{\max,j}(t)-y_{i,j}(t) \tag{27}$$

where $y_{i,j}(t)$ is current position of particle $i$, $\overline{y}_{i,j}(t)$ is the result of reverse search of particle $i$, $y_{\min,j}(t)$ and $y_{\max,j}(t)$ are lower and upper bounds for the dimension $j$, respectively.

COL algorithm consists of two main parts. In the first part, we use Tent-Sine system to generate an initial population. In the second part, we use OL to generate an opposite population. Then these two populations are amalgamated and a new initial population is generated.

Suppose all particles fly in *D*-dimensional space. The steps of COL are described by Algorithm 1.

**Algorithm 1. COL algorithm**

Input: Set the population size $S$ and the maximum number of iterations $K_{\max}$.
Output: Initial population.

```
for i = 1 to S do
  for j = 1 to D do
    Randomly generate the value of Q_{0,j} ∈ (0,1);
    for k = 1 to K_max do
      Q_{k,j} = Q_{k-1,j};
    end for
    y_{i,j}(t) = y_{min,j}(t) + Q_{k,j}(y_{max,j}(t) − y_{min,j}(t));
  end for
end for
for i = 1 to S do
  for j = 1 to D do
    ȳ_{i,j}(t) = y_{min,j}(t) + y_{max,j}(t) − y_{i,j}(t);
  end for
end for
Select S fittest individuals from the set of {y(t) ∪ ȳ(t)} as current population.
```

### 3.2.2 Enhancement of local search ability

CA is a space and time-discrete dynamical system that evolves according to a set of local rules and neighbor cells. It is composed of five basic components, which are cell, cell space, cell state, neighborhood, and transition rule. As is shown in Fig. 4, Moore model is one of the well-known neighborhoods in 2-dimensional CA. The black cell is the center one, and those gray ones are its neighbors. At a discrete time step, the update of a cell state obtains by taking into account the states of cells in its local neighborhood. All cells communicate and change states synchronously.

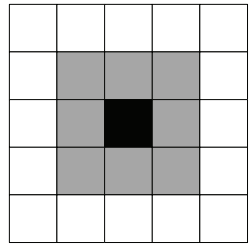

**Fig. 4.** Moore model.

Current studies have shown that CA is a powerful method to enhance the local search ability of the PSO algorithm [24]. In EODPSO algorithm, CA is integrated in particle update to avoid local minima, and each particle is defined as a single cell and is distributed in 2-dimensional space. The CA model for EODPSO algorithm is described as follows:

(1) Cell: each particle is defined as a single cell;
(2) Cell space: the set of all cells;

(3) Cell state: cell state is defined as the particle's position. The state of cell $i$ is defined as $s_i(t)=y_i(t)$;

(4) Neighborhood: the neighborhood of particle $i$ is defined as $H(i)=\{i+1,i+2,\cdots,i+h\}$, where $h$ is the number of neighbors;

(5) Transition rule: $s_i(t)=f\left(s_i(t),s_{i+1}(t),s_{i+2}(t),\cdots,s_{i+h}(t)\right)$.

### 3.2.3 Population search strategy

In order to accurately judge when population search strategy is implemented, we use the entropy of IFSs as population diversity index and utilize the proposed entropy measure to achieve the state of particle swarm in the EODPSO algorithm. According to Theorem 1, the proposed intuitionistic fuzzy entropy measure in PSO algorithm is defined as follows:

**DEFINITION 7:** *Let $A'$ is a particle swarm in a finite set $X=\{x_i|i=1,2,\cdots,n\}$. At iteration $t$, the intuitionistic fuzzy entropy measure in PSO algorithm is defined as*:

$$E^t(A')=\frac{1}{n}\sum_{i=1}^{n}\frac{2-2\left|\mu^t_{A'_{k,i}}(x_i)-\gamma^t_{A'_{k,i}}(x_i)\right|+{\pi^t_{A'_{k,i}}}^2(x_i)}{3+\mu^t_{A'_{k,i}}(x_i)+\gamma^t_{A'_{k,i}}(x_i)+\left|\mu^t_{A'_{k,i}}(x_i)-\gamma^t_{A'_{k,i}}(x_i)\right|} \tag{28}$$

*where $E^t(A')$ is in range $[0,1]$.*

It is known from Definition 7 that when the entropy value equals 0, all particles converge to one point. When entropy value equals 1, all particles reach the maximum dispersion. In addition, $\beta$ is used as a threshold to decide whether the population search strategy is implemented. When the entropy value is smaller than $\beta$, it means that EODPSO algorithm falls into local optima. In this case, we use the proposed population search strategy to enhance global search ability.

DE algorithm is an evolutionary algorithm based on real number coding and the evolution of population. It guides the population towards the vicinity of the global optimum by repeating cycles of mutation, crossover and selection. DE algorithm starts with a population of NP $D$-dimensional search variable vectors. During the iteration $t+1$, for each individual $y_i(t)=\{y_{i_1}(t),y_{i_2}(t),\cdots y_{i_n}(t)\}$ $(i\in\{1,2,\cdots,NP\})$, a mutation vector $\overline{v}_i(t)$ is computed by using the following mutation operator:

$$\overline{v}_i(t)=y_{\alpha_1}(t)+F\left(y_{\alpha_2}(t)-y_{\alpha_3}(t)\right) \tag{29}$$

where $\alpha_1$, $\alpha_2$ and $\alpha_3$ are random unequal integers within the set $\{1,\cdots,i-1,i+1,\cdots,N\}$. $F$ is the mutation control parameter.

The trial vector $y'_{i,j}(t+1)$ is generated by using the following crossover operation:

$$y'_{i,j}(t+1)=\begin{cases}\overline{v}_{i,j}(t) & \text{if } \left(rand_j\le CR\right) \text{ or } \left(j=j_{rand}\right)\\ y_{i,j}(t) & \text{otherwise}\end{cases} \tag{30}$$

where $j=1,2,\cdots,N$, $y'_{i,j}(t+1)$, $\overline{v}_{i,j}(t)$ and $y_{i,j}(t)$ are $j^{\text{th}}$ parameter for the $i^{\text{th}}$ trial, mutant and vectors. $rand_j$ is a uniformly distributed random number within $[0,1]$. $CR$ is a cross control parameter and is

usually selected from within $(0,1]$. It controls the diversity of population and helps the DE algorithm to escape from local optima. $j_{rand}$ is a randomly chosen index within $[0,N]$. It ensures that $y'_{i,j}(t+1)$ gets at least one parameter from $\overline{v}_i(t)$.

Moreover, as discussed in Section 3.2.1, OL is also an effective algorithm for searching population. The main idea of OL is to consider an estimate and its corresponding opposite estimate simultaneously. Our new population search strategy makes use of the merits of DE and OL algorithm. It not only helps the PSO algorithm to escape from local optima but also accelerates the convergence speed of the PSO algorithm. In the new population search strategy, OL and DE are applied to obtain a new population. Specifically, we firstly use OL to generate an opposite population, and then we use mutation and crossover to generate a mutation population. Finally, we select *S* fittest particles from the current population, opposite population and mutation population as a new population.

### 3.2.4 EODPSO algorithm

In EODPSO algorithm, each particle is defined as cell and updates its current status with neighbors' states. Based on the evolution mechanism of CA, COL is used to generate a reasonable initial population. Then, at each iteration, each particle's velocity and position are updated by using the evolution mechanism of CA. When the entropy value of the population is smaller than threshold $\beta$, then OL and DE are applied to generate an opposite population and a mutation population, respectively. A new population is obtained by selecting *S* fittest particles from the current population, opposite population and mutation population. At the next iteration, the new population is considered as an initial population.

To accelerate convergence speed and improve optimization quality effectively, inertia weight $w$ is calculated by using a linear decreasing way. $w$ is calculated as follows:

$$w(t) = w_{\max} - \frac{t(w_{\max} - w_{\min})}{T_{\max}} \tag{31}$$

where $w_{\max}$ and $w_{\min}$ represent the maximum and minimum of $w$ respectively, $t$ is the current iterative number and $T_{\max}$ is the maximum number of iterations.

EODPSO algorithm is shown as follows:

**Algorithm 2. EODPSO algorithm**

Input: Set the parameters including $w_{\min}$, $w_{\max}$, $c_1$, $c_2$, $\beta$, the maximum number of iterations $T_{\max}$ and the number of neighbors $h$.

Output: $p_g$.

1. Randomly initialize $r_1$ and $r_2$;
2. Generate the initial particles with positions $y_i(t)$ by using Algorithm 1 and random velocities $v_i(t)$ in the cell space;
3. For each initial particle, set $p_i(t) = y_i(t)$ and $s_i(t) = y_i(t)$;
4. Evaluate each initial particle;
5. Identify the best position $p_g(t)$;
6. $t = 1$;

7. while $(t < T_{\max})$
8. Calculate $w(t)$ by using Equation (31);
9. for $i = 1$ to $S$ do
10. for $j = 1$ to $h$ do
11. if fitness$(s_i(t))$ <fitness$(s_{i+h}(t))$
12. Update $s_i(t)$;
13. end if
14. end for
15. $v_i(t+1) = wv_i(t) + c_1 r_1 (p_i(t) - y_i(t))$
$+c_2 r_2 (p_g(t) - y_i(t))$;
16. $y_i(t+1) = y_i(t) + v_i(t)$;
17. $s_i(t+1) = y_i(t+1)$;
18. if fitness$(p_i(t))$ <fitness$(s_i(t+1))$
19. Update $p_i(t)$;
20. end if
21. if fitness$(p_g(t))$ <fitness$(p_i(t))$
22. Update $p_g(t)$;
23. end if
24. end for
25. Calculate the entropy value of population according Equation (28);
26. if $E(A) < \beta$
27. Generate opposite population $OP$ by using Equation (27);
28. Generate mutation population $MP$ by using Equation (29) and (30);
29. Generate a new population according to current population, $OP$ and $MP$;
30. Update $p_i(t)$ and $p_g(t)$;
31. end if
32. $t = t + 1$;
33. end while

## 3.3 Mobile User Interface Pattern Clustering

In this section, we propose a SSKFCM clustering method to cluster the MUIP data. In the framework of MUIP clustering, EODPSO algorithm is used to optimize clustering parameters.

First of all, EODPSO algorithm is exploited to optimize the kernel parameter σ of Gaussian kernel function. During the process of optimizing parameter σ, a fitness function is defined to assess the generated solutions. One of the clustering evaluation metric, Davies-Bouldin index (DBI) [41], is used in this fitness function to measure the influence of σ on clustering quality. The lower value of DBI signifies better clustering result. The fitness function is defined as follows:

$$f_1(x) = \frac{1}{1+\overline{R}} \tag{32}$$

where $\overline{R}$ is DBI.

The DBI is defined as follows:

$$\overline{R} \equiv \frac{1}{e'} \sum_{i=1}^{e'} R_i \tag{33}$$

where $e'$ is the number of clusters. $R_i \equiv$ maximum of $R_{i,j}$.

$$R_{i,j} \equiv \frac{d_i + d_j}{M_{i,j}} \tag{34}$$

where $d_i$ and $d_j$ are the scatter with the $i^{\text{th}}$ center and $j^{\text{th}}$ center, respectively. $M_{i,j}$ is the distance between the $i^{\text{th}}$ cluster center and the $j^{\text{th}}$ cluster center.

We can see that the smaller the value of $\overline{R}$, the greater the fitness value $f_1(x)$ and the better value of parameter.

Then, considering that the cluster centers and the number of clusters also influence the clustering effect directly, the EODPSO algorithm is reused to initialize the cluster centers and DBI is used as a judgment index to determine the optimal number of clusters. In [42], they discussed the optimal number of clusters. The results of their study show that the optimal number of clusters is in range $\left[0, \sqrt{n}\right]$, where $n$ is the size of the data set $Z=\{z_1, z_2, \cdots, z_n\}$.

A fitness function is defined to assess the generated cluster centers during the initialization phase. The definition of the fitness function is expressed below:

$$f_2(x) = \frac{1}{1 + J(U,B)} \tag{35}$$

where $J(U,B)$ is the objective function of KFCM, as shown in Eq. (5).

It has been expressed clearly that the smaller the value of $J(U,B)$, the better the effect of the clustering and the greater the fitness value of $f_2(x)$.

Fig. 5 illustrates the framework of MUIP clustering method. The MUIP clustering includes two stages. In the first stage, initial cluster centers are generated by using EODPSO algorithm. In the second stage, the kernel parameter is optimized by utilizing EODPSO algorithm. Then, the fuzzy partition matrix and cluster centers are updated. If the value of DBI is greater than maximum number of clusters $e_{\max}$, clustering method is stopped and the fuzzy partition matrix and cluster centers are output. Otherwise, clustering method continues to optimize the kernel parameter and update the fuzzy partition matrix and cluster centers.

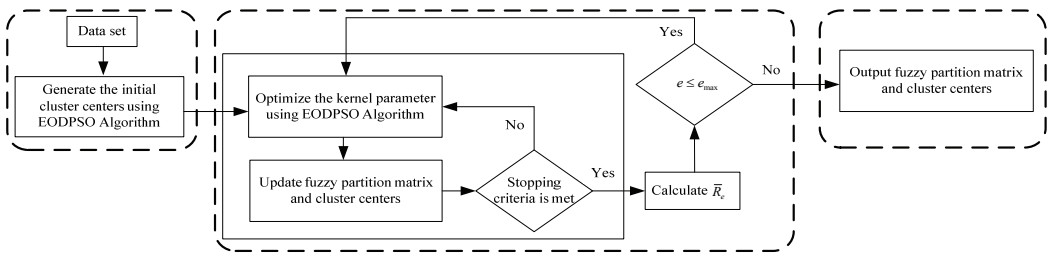


**Fig. 5.** Framework of MUIP clustering.

Algorithm 3 shows the specific steps of MUIP clustering. MUIP clustering consists of two loops. The inner loop is used to optimize the parameter σ and update $u_{i,j}$ and $b_j$. The outer loop is used to determine the fuzzy partition matrix and cluster centers. At each iteration of the inner loop, the clustering method first optimizes the parameter σ by using EODPSO algorithm and fitness function $f_1(x)$. Then, the fuzzy partition matrix $u_{i,j}$ and cluster centers $b_j$ are updated. At each iteration of the outer loop, the value of DBI is calculated and used to determine the number of clusters.

**Algorithm 3. MUIP clustering**

Input: Set the parameters including $\eta$, $m$, $F$, $CR$, the maximum number of iterations $q_{\max}$ and the maximum number of clusters $e_{\max} = \sqrt{n}$.

Output: $U$ and $B$.

1. Generate the initial cluster centers by using Algorithm 2 and Equation (35);
2. $q = 1$;
3. for $e = 2$ to $e_{\max}$ do
4. while $\left( \left\| U_q^u - U_{q-1}^u \right\| \geq \eta \text{ or } q < q_{\max} \right)$
5. Optimize the parameter $\sigma$ by using Algorithm 2 and Equation (32);
6. Update $u_{i,j}$ by using Equation (11);
7. Update $b_j$ by using Equation (12);
8. $q = q + 1$;
9. end while
10. Calculate $\overline{R}_e$ by using Equation (33) and (34);
11. end for
12. Identify the minimum $\overline{R}_e$, and then select its corresponding $U$ and $B$ as clustering result.

# 4. Experiments and Analysis

## 4.1 Data Collection

In general, developers select MUIPs according to functional goals. Therefore, we exploit the MUIP clustering method to cluster MUIPs based on the functional goals. Our data set is constructed by collecting 5,397 MUIPs from [43,44] and seven MUIPs websites (http://www.mobile-patterns.com/, http://inspired-ui.com/, http://ui-patterns.com/, http://pttrns.com/, https://www.cocoacontrols.com/, http://ios-patterns.com/, http://mobiledesign.com/). To facilitate testing, we define 51 types of functional goals based on our previous study on MUIP data collection [11].

## 4.2 Evaluation Metrics

In our experiments, DBI, accuracy [45], adjusted rand index (ARI) [46] and normalized mutual information (NMI) [47] are employed to evaluate the performance of the experimented methods. DBI is used to measure the within-cluster scatter and the inter-cluster separation. Since ARI is not sensitive to the number of clusters, it is used to compare two partitions with different cluster numbers. Accuracy and

NMI are used to measure the quality and efficiency of the cluster method, respectively. Note that the greater values of accuracy, ARI and NMI signify better clustering result. DBI is defined above and the other three metrics are defined as follows:

$$Accuracy = \frac{\sum_{i=1}^{n} \delta\left(a_i, \mathrm{map}\left(\overline{a}_i\right)\right)}{n} \tag{36}$$

where $n$ is the total number of data, $a_i$ and $\overline{a}_i$ are the true class label and the obtained cluster label, $\delta\left(a,\overline{a}\right)$ is a function that equals 1 if $a=\overline{a}$ and equals 0 otherwise. Function $\mathrm{map}\left(\cdot\right)$ is a permutation function that maps each cluster label to a class label. The optimal matching can be found with Hungarian algorithm [48].

$$ARI = \frac{\sum_{i=1}^{e_1}\sum_{j=1}^{e_2}\binom{n_{i,j}}{2} - \binom{n}{2}^{-1}\sum_{i=1}^{e_1}\binom{n_i}{2}\sum_{j=1}^{e_2}\binom{n_j}{2}}{\frac{1}{2}\left[\sum_{i=1}^{e_1}\binom{n_i}{2} + \sum_{j=1}^{e_2}\binom{n_j}{2}\right] - \binom{n}{2}^{-1}\sum_{i=1}^{e_1}\binom{n_i}{2}\sum_{j=1}^{e_2}\binom{n_j}{2}} \tag{37}$$

where $n$ is the total number of data, $n_{i,j}$ is the number of data that belong to cluster $i$ and cluster $j$, $n_i$ denotes the number of data that belong to cluster $i$, $n_j$ denotes the number of data that belong to cluster $j$. $e_1$ is the number of clusters that are obtained after running a clustering method. $e_2$ is the number of clusters that come from priori partition.

$$NMI = \frac{\sum_{i=1}^{e'}\sum_{j=1}^{e'} n_{i,j}\log\frac{nn_{i,j}}{n_i n_j}}{\sqrt{\sum_{i=1}^{e'} n_i \log\frac{n_i}{n}\sum_{j=1}^{e'} n_j \log\frac{n_j}{n}}} \tag{38}$$

where $e'$ is the number of clusters, $n$ is the total number of data, $n_{i,j}$ denotes the number of data in cluster $i$ as well as in cluster $j$, $n_i$ and $n_j$ denote the number of data in cluster $i$ and cluster $j$, respectively.

## 4.3 Parameters Setting

Before applying our method to MUIP data, several parameters need to be set. The parameters of our method are set as: $S=50$, $K_{\max}=300$, $w_{\max}=0.9$, $w_{\min}=0.1$, $c_1=c_2=2$, $T_{\max}=500$, $h=8$, $\eta$=0.0001, $m=2$, $q_{\max}=800$, $F=0.5$, $CR=1$. Considering that the change of entropy value $\beta$ affects the performance of the EODPSO algorithm and further affects the clustering result of the proposed clustering method, we evaluate the critical $\beta$ by using the above evaluation metrics. We increase $\beta$ gradually from 0.1 to 0.9 and the proposed clustering method is run 20 times for each value of $\beta$. We randomly select 10% data as labeled data and the rest as unlabeled data each time. Fig. 6 reports the mean values and standard deviations of the four evaluation metrics for different values of $\beta$. It should be noted that "Mean" indicates the mean value and "Std" indicates the standard deviation in this paper. It is clear that these four evaluation metrics change significantly when changing the value of $\beta$ from 0.1 to 0.7, but

they change very slightly when changing the value of $\beta$ from 0.7 to 0.9. At the same time, we can see from the results that values of Accuracy, ARI and NMI are close to their maximums and values of DBI is close to its minimum after $\beta$=0.7. This means when $\beta$<0.7, EODPSO algorithm falls into local optima. In this case, it requires a population search strategy to jump out of local optima. On the contrary, when $\beta \geq 0.7$, EODPSO algorithm does not fall into local optima. In this case, the population diversity is not much affected by the increasing value of $\beta$. All these results indicate that the critical value of $\beta$ is 0.7. Therefore, we set $\beta$=0.7 in our clustering method.

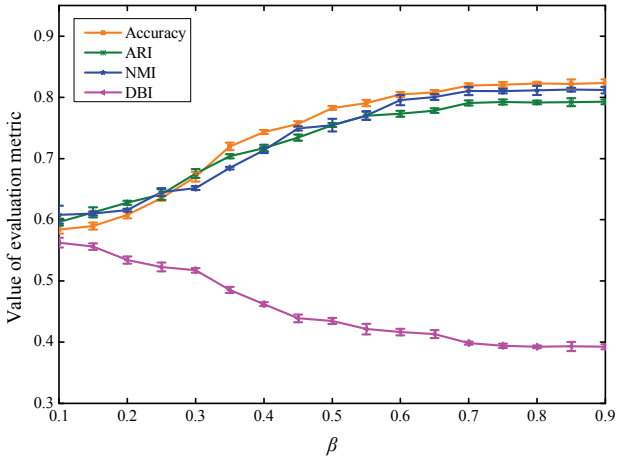


**Fig. 6.** The effect of parameter $\beta$ on the proposed clustering method.

## 4.4 Results and Comparisons

In this paper, the experimental studies focus on the clustering effect and the convergence of the proposed MUIP clustering method. In order to make it evident to show the performance of EODPSO clustering method, other relevant clustering methods are used as comparisons. Specifically, DBSCAN, GDBSCAN, BIRCH and CLARANS are used in semi-supervised clustering method to cluster MUIP data. Additionally, GA, SA, PSO, and four variant PSO algorithms mentioned in this paper, namely DEPSO, CPSO, CAPSO, and EPSO, are used to replace EODPSO algorithm in the framework of MUIP clustering. Then the seven alternative SSKFCM clustering methods are applied to cluster MUIP data.

The first experiment is to compare the clustering effect of those clustering methods in different ratios of labeled data. All methods are run 20 times independently on MUIP data. Fig. 7 shows the mean values of four evaluation metrics of the clustering method accompanied with standard deviations. The mean value indicates the clustering effect of the methods and the standard deviation indicates the stability of the methods. Mean values of four evaluation metrics are clearly evident that PSO-based SSKFCM clustering methods usually outperform other semi-supervised clustering methods. The reasons are as follows. Firstly, density-based spatial clustering methods cannot discover overlapping clusters. Secondly, BIRCH and CLARANS clustering methods only work well for convex or spherical clusters. Thirdly, GA and CA easily fall into local optima and have slow convergence speed. Moreover, it can be seen that EODPSO clustering method surpasses all other PSO-based clustering methods. Compared to other PSO algorithms, EODPSO algorithm has better global search ability because it utilizes a hybrid strategy to

avoid local minima and accelerate convergence speed. Thus EODPSO clustering method achieves better results than other PSO-based clustering methods.

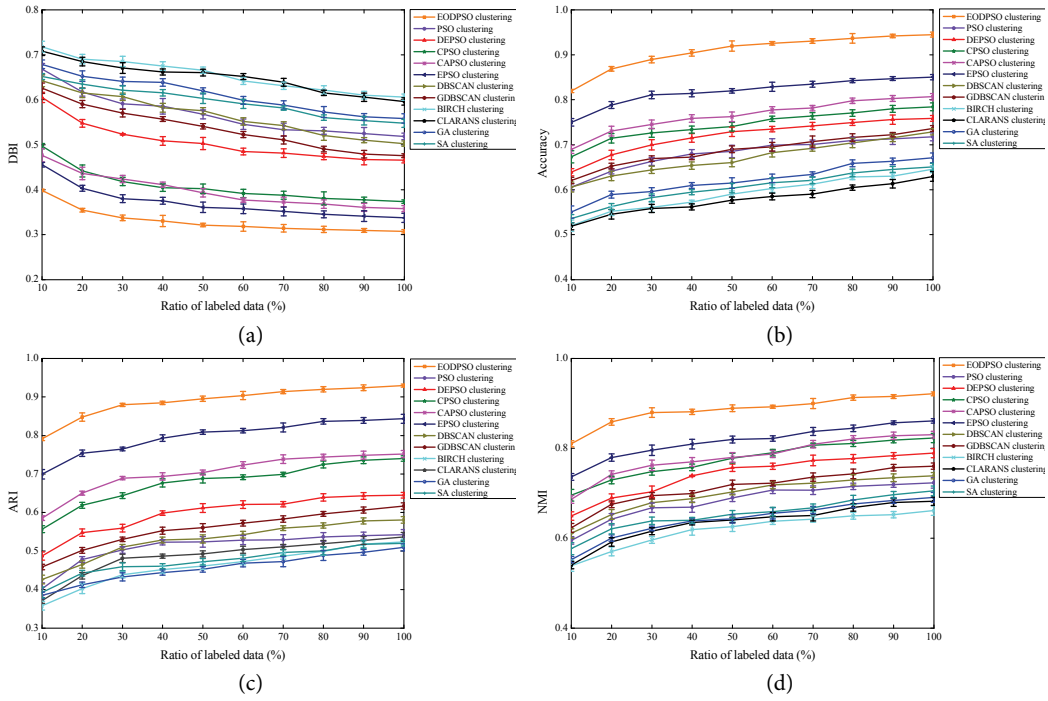


**Fig. 7.** Comparison of all clustering methods under different ratio of labeled data: (a) DBI, (b) accuracy, (c) ARI, and (d) NMI.

It is worth noting that the performances of all methods are less dependent on labeled data as the ratio of labeled data increases. The reason is that when the number of labeled data reaches a certain ratio, the whole MUIP data space can be represented better and the influence of labeled data on clustering effect decreases gradually.

Tables 2–7 and Fig. 8 show more details on the performance of the clustering methods. The best results are highlighted in bold in tables. As can be seen from Tables 2–5, the overall standard deviations of four evaluation metrics of EODPSO clustering method are lower than that of other methods. It means that EODPSO clustering method has not only better clustering effect but also better stability than all other clustering methods.

**Table 2.** Mean and standard deviation results of DBI for all clustering methods

| Clustering method | | Ratio of labeled data (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| EODPSO | Mean | **0.3985** | **0.3546** | **0.3373** | **0.3305** | **0.3215** | **0.3182** | **0.3143** | **0.3117** | **0.3094** | **0.3076** |
| | Std | **0.0025** | **0.0041** | **0.0063** | 0.0124 | **0.0039** | 0.0105 | 0.0085 | **0.0068** | **0.0042** | **0.0038** |
| PSO | Mean | 0.6681 | 0.6174 | 0.5911 | 0.5855 | 0.5676 | 0.5456 | 0.5335 | 0.5309 | 0.5248 | 0.5185 |
| | Std | 0.0125 | 0.0091 | 0.0108 | 0.0174 | 0.0098 | 0.0116 | 0.0145 | 0.0078 | 0.0137 | 0.0074 |
| DEPSO | Mean | 0.6052 | 0.5477 | 0.5234 | 0.5086 | 0.5028 | 0.4847 | 0.4813 | 0.4739 | 0.4672 | 0.4655 |
| | Std | 0.0115 | 0.0085 | 0.00145 | 0.0095 | 0.0135 | 0.0074 | 0.0095 | **0.0068** | 0.0116 | 0.0061 |

| Clustering method | | Ratio of labeled data (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| CPSO | Mean | 0.4968 | 0.4419 | 0.4182 | 0.4047 | 0.4021 | 0.3917 | 0.3878 | 0.3806 | 0.3772 | 0.3737 |
| | Std | 0.0059 | 0.0133 | 0.0093 | 0.0085 | 0.0105 | 0.0093 | 0.0089 | 0.0145 | 0.0062 | 0.0041 |
| CAPSO | Mean | 0.4765 | 0.4363 | 0.4238 | 0.4114 | 0.3933 | 0.3771 | 0.3727 | 0.3683 | 0.3607 | 0.3579 |
| | Std | 0.0146 | 0.0139 | 0.0082 | 0.0058 | 0.0092 | **0.0047** | 0.0106 | 0.0095 | 0.0087 | 0.0069 |
| EPSO | Mean | 0.4557 | 0.4033 | 0.3801 | 0.3754 | 0.3609 | 0.3577 | 0.3517 | 0.3452 | 0.3414 | 0.3375 |
| | Std | 0.0052 | 0.0066 | 0.0085 | 0.0077 | 0.0117 | 0.0106 | 0.0099 | 0.0078 | 0.0119 | 0.0101 |
| DBSCAN | Mean | 0.6423 | 0.6152 | 0.6067 | 0.5823 | 0.5758 | 0.5518 | 0.5425 | 0.5204 | 0.5101 | 0.5025 |
| | Std | 0.0132 | 0.0105 | 0.0082 | 0.0097 | 0.0068 | 0.0081 | 0.0087 | 0.0105 | 0.0058 | 0.0059 |
| GDBSCAN | Mean | 0.6258 | 0.5905 | 0.5703 | 0.5571 | 0.5409 | 0.5225 | 0.5108 | 0.4905 | 0.4792 | 0.4757 |
| | Std | 0.0115 | 0.0079 | 0.0087 | **0.0051** | 0.0059 | 0.0068 | 0.0091 | 0.0062 | 0.0084 | 0.0044 |
| BIRCH | Mean | 0.7177 | 0.6905 | 0.6852 | 0.6751 | 0.6658 | 0.6419 | 0.6312 | 0.6218 | 0.6102 | 0.6057 |
| | Std | 0.0125 | 0.0105 | 0.0112 | 0.0095 | 0.0071 | 0.0078 | 0.0083 | 0.0092 | 0.0085 | 0.0068 |
| CLARANS | Mean | 0.7081 | 0.6849 | 0.6705 | 0.6618 | 0.6603 | 0.6517 | 0.6389 | 0.6154 | 0.6058 | 0.5962 |
| | Std | 0.0097 | 0.0095 | 0.0118 | 0.0068 | 0.0074 | 0.0066 | 0.0087 | **0.0059** | 0.0093 | 0.0082 |
| GA | Mean | 0.6788 | 0.6525 | 0.6409 | 0.6385 | 0.6197 | 0.5986 | 0.5884 | 0.5728 | 0.5621 | 0.5584 |
| | Std | 0.0095 | 0.0115 | 0.0093 | 0.0085 | 0.0068 | 0.0087 | 0.0094 | 0.0121 | 0.0075 | 0.0103 |
| SA | Mean | 0.6518 | 0.6347 | 0.6214 | 0.6156 | 0.6033 | 0.5909 | 0.5817 | 0.5604 | 0.5536 | 0.5477 |
| | Std | 0.0067 | 0.0095 | 0.0094 | 0.0072 | 0.0117 | 0.0102 | **0.0047** | 0.0061 | 0.0093 | 0.0088 |

The best results are highlighted in bold.

**Table 3.** Mean and standard deviation results of accuracy for all clustering methods

| Clustering method | | Ratio of labeled data (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| EODPSO | Mean | **0.8194** | **0.8686** | **0.8895** | **0.9043** | **0.9195** | **0.9256** | **0.9307** | **0.9365** | **0.9419** | **0.9448** |
| | Std | **0.0035** | **0.0052** | 0.0074 | 0.0072 | 0.0115 | 0.0041 | **0.0053** | 0.0107 | **0.0038** | **0.0056** |
| PSO | Mean | 0.6057 | 0.6409 | 0.6628 | 0.6799 | 0.6852 | 0.6994 | 0.7008 | 0.7095 | 0.7132 | 0.7176 |
| | Std | 0.0081 | 0.0057 | 0.0087 | 0.0093 | 0.0137 | 0.0142 | 0.0096 | 0.0075 | 0.0124 | 0.0088 |
| DEPSO | Mean | 0.6389 | 0.6774 | 0.6998 | 0.7153 | 0.7289 | 0.7347 | 0.7419 | 0.7493 | 0.7559 | 0.7587 |
| | Std | 0.0079 | 0.0103 | 0.0114 | 0.0096 | 0.0103 | 0.0062 | 0.0081 | 0.0066 | 0.0103 | 0.0065 |
| CPSO | Mean | 0.6731 | 0.7144 | 0.7262 | 0.7337 | 0.7404 | 0.7578 | 0.7642 | 0.7707 | 0.7798 | 0.7843 |
| | Std | 0.0136 | 0.0107 | 0.0098 | 0.0077 | 0.0096 | 0.0058 | 0.0072 | 0.0073 | 0.0076 | 0.0086 |
| CAPSO | Mean | 0.6896 | 0.7304 | 0.7453 | 0.7585 | 0.7622 | 0.7776 | 0.7815 | 0.7978 | 0.8029 | 0.8067 |
| | Std | 0.0087 | 0.0108 | 0.0097 | 0.0083 | 0.0107 | 0.0069 | 0.0059 | 0.0061 | 0.0065 | 0.0072 |
| EPSO | Mean | 0.7498 | 0.7884 | 0.8108 | 0.8143 | 0.8198 | 0.8289 | 0.8347 | 0.8425 | 0.8471 | 0.8502 |
| | Std | 0.0085 | 0.0077 | 0.0083 | 0.0079 | **0.0049** | 0.0102 | 0.0066 | **0.0047** | 0.0041 | 0.0059 |
| DBSCAN | Mean | 0.6059 | 0.6307 | 0.6442 | 0.6537 | 0.6605 | 0.6825 | 0.6919 | 0.7039 | 0.7156 | 0.7288 |
| | Std | 0.0112 | 0.0105 | 0.0078 | 0.0082 | 0.0093 | 0.0097 | 0.0065 | 0.0104 | 0.0085 | 0.0069 |
| GDBSCAN | Mean | 0.6205 | 0.6524 | 0.6693 | 0.6723 | 0.6899 | 0.6954 | 0.7072 | 0.7161 | 0.7224 | 0.7361 |
| | Std | 0.0071 | 0.0065 | **0.0061** | 0.0058 | 0.0074 | 0.0096 | 0.0124 | 0.0085 | 0.0049 | 0.0064 |
| BIRCH | Mean | 0.5201 | 0.5525 | 0.5609 | 0.5725 | 0.5905 | 0.6026 | 0.6122 | 0.6281 | 0.6303 | 0.6457 |
| | Std | 0.0128 | 0.0099 | 0.0113 | **0.0042** | 0.0071 | 0.0085 | 0.0095 | 0.0059 | 0.0067 | 0.0078 |
| CLARANS | Mean | 0.5187 | 0.5454 | 0.5584 | 0.5619 | 0.5767 | 0.5851 | 0.5905 | 0.6048 | 0.6133 | 0.6297 |
| | Std | 0.0078 | 0.0106 | 0.0092 | 0.0067 | 0.0069 | 0.0075 | 0.0081 | 0.0058 | 0.0095 | 0.0108 |
| GA | Mean | 0.5504 | 0.5891 | 0.5957 | 0.6097 | 0.6147 | 0.6255 | 0.6339 | 0.6587 | 0.6631 | 0.6709 |
| | Std | 0.0134 | 0.0077 | 0.0089 | 0.0065 | 0.0117 | 0.0095 | 0.0058 | 0.0075 | 0.0071 | 0.0109 |
| SA | Mean | 0.5357 | 0.5625 | 0.5823 | 0.5944 | 0.6036 | 0.6154 | 0.6208 | 0.6374 | 0.6452 | 0.6509 |
| | Std | 0.0089 | 0.0067 | 0.0084 | 0.0057 | 0.0104 | 0.0108 | 0.0074 | 0.0082 | 0.0065 | 0.0077 |

The best results are highlighted in bold.

**Table 4.** Mean and standard deviation results of ARI for all clustering methods

| Clustering method | | Ratio of labeled data (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| EODPSO | Mean | **0.7911** | **0.8479** | **0.8795** | **0.8846** | **0.8954** | **0.9037** | **0.9138** | **0.9198** | **0.9237** | **0.9296** |
| | Std | **0.0042** | 0.0107 | **0.0042** | **0.0045** | 0.0066 | 0.0106 | **0.0054** | 0.0069 | **0.0074** | **0.0038** |
| PSO | Mean | 0.4022 | 0.4781 | 0.5028 | 0.5229 | 0.5238 | 0.5284 | 0.5292 | 0.5367 | 0.5399 | 0.5426 |
| | Std | 0.0155 | 0.0074 | 0.0084 | 0.0069 | 0.0138 | 0.0117 | 0.0133 | 0.0112 | 0.0093 | 0.0136 |
| DEPSO | Mean | 0.4871 | 0.5479 | 0.5595 | 0.5986 | 0.6117 | 0.6208 | 0.6215 | 0.6393 | 0.6432 | 0.6451 |
| | Std | 0.0106 | 0.0096 | 0.0097 | 0.0057 | 0.0117 | 0.0101 | 0.0067 | 0.0097 | 0.0088 | 0.0072 |
| CPSO | Mean | 0.5569 | 0.6188 | 0.6438 | 0.6767 | 0.6881 | 0.6917 | 0.6991 | 0.7247 | 0.7358 | 0.7404 |
| | Std | 0.0091 | 0.0078 | 0.0075 | 0.0107 | 0.0114 | 0.0063 | 0.0057 | 0.0092 | 0.0087 | 0.0069 |
| CAPSO | Mean | 0.5855 | 0.6504 | 0.6893 | 0.6935 | 0.7039 | 0.7233 | 0.7386 | 0.7438 | 0.7489 | 0.7517 |
| | Std | 0.0058 | **0.0053** | 0.0044 | 0.0094 | 0.0063 | 0.0086 | 0.0112 | 0.0075 | 0.0105 | 0.0093 |
| EPSO | Mean | 0.6994 | 0.7539 | 0.7648 | 0.7931 | 0.8087 | 0.8125 | 0.8208 | 0.8367 | 0.8392 | 0.8433 |
| | Std | 0.0124 | 0.0083 | 0.0046 | 0.0086 | **0.0059** | **0.0058** | 0.0116 | 0.0071 | 0.0076 | 0.0115 |
| DBSCAN | Mean | 0.4261 | 0.4652 | 0.5104 | 0.5289 | 0.5317 | 0.5425 | 0.5596 | 0.5663 | 0.5782 | 0.5809 |
| | Std | 0.0112 | 0.0095 | 0.0068 | 0.0074 | 0.0105 | 0.0088 | 0.0062 | 0.0069 | 0.0081 | 0.0085 |
| GDBSCAN | Mean | 0.4585 | 0.5017 | 0.5305 | 0.5528 | 0.5607 | 0.5725 | 0.5831 | 0.5966 | 0.6065 | 0.6171 |
| | Std | 0.0069 | 0.0075 | 0.0058 | 0.0091 | 0.0107 | 0.0074 | 0.0083 | **0.0062** | 0.0079 | 0.0082 |
| BIRCH | Mean | 0.3582 | 0.4028 | 0.4387 | 0.4519 | 0.4611 | 0.4725 | 0.4868 | 0.4993 | 0.5177 | 0.5239 |
| | Std | 0.0115 | 0.0131 | 0.0085 | 0.0096 | 0.0091 | 0.0082 | 0.0079 | 0.0098 | 0.0106 | 0.0072 |
| CLARANS | Mean | 0.3717 | 0.4359 | 0.4812 | 0.4869 | 0.4927 | 0.5036 | 0.5108 | 0.5193 | 0.5276 | 0.5369 |
| | Std | 0.0073 | 0.0084 | 0.0109 | 0.0056 | 0.0078 | 0.0092 | 0.0064 | 0.0085 | 0.0105 | 0.0117 |
| GA | Mean | 0.3851 | 0.4125 | 0.4329 | 0.4441 | 0.4525 | 0.4688 | 0.4726 | 0.4886 | 0.4969 | 0.5098 |
| | Std | 0.0085 | 0.0071 | 0.0105 | 0.0069 | 0.0073 | 0.0098 | 0.0134 | 0.0128 | 0.0076 | 0.0094 |
| SA | Mean | 0.3925 | 0.4426 | 0.4592 | 0.4609 | 0.4722 | 0.4814 | 0.4961 | 0.5005 | 0.5174 | 0.5204 |
| | Std | 0.0139 | 0.0072 | 0.0103 | 0.0065 | 0.0082 | 0.0087 | 0.0073 | 0.0102 | 0.0095 | 0.0073 |

The best results are highlighted in bold.

**Table 5.** Mean and standard deviation results of NMI for all clustering methods

| Clustering method | | Ratio of labeled data (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| EODPSO | Mean | **0.8103** | **0.8587** | **0.8795** | **0.8812** | **0.8891** | **0.8925** | **0.8996** | **0.9129** | **0.9153** | **0.9213** |
| | Std | **0.0067** | **0.0072** | 0.0107 | 0.0059 | 0.0074 | **0.0035** | 0.0113 | **0.0059** | 0.0043 | **0.0041** |
| PSO | Mean | 0.5951 | 0.6422 | 0.6678 | 0.6691 | 0.6899 | 0.7072 | 0.7068 | 0.7155 | 0.7193 | 0.7232 |
| | Std | 0.0085 | 0.0128 | 0.0105 | 0.0113 | 0.0084 | 0.0079 | 0.0096 | 0.0071 | 0.0062 | 0.0117 |
| DEPSO | Mean | 0.6494 | 0.6895 | 0.7034 | 0.7388 | 0.7568 | 0.7603 | 0.7731 | 0.7771 | 0.7836 | 0.7892 |
| | Std | 0.0095 | 0.0091 | 0.0129 | 0.00132 | 0.0081 | 0.0069 | 0.0118 | 0.0093 | 0.0072 | 0.0101 |
| CPSO | Mean | 0.6958 | 0.7296 | 0.7481 | 0.7576 | 0.7782 | 0.7903 | 0.8069 | 0.8108 | 0.8182 | 0.8229 |
| | Std | 0.0131 | 0.0082 | 0.0076 | 0.0075 | 0.0109 | 0.0073 | **0.0047** | 0.0067 | 0.0058 | 0.0102 |
| CAPSO | Mean | 0.6866 | 0.7423 | 0.7622 | 0.7698 | 0.7799 | 0.7873 | 0.8093 | 0.8206 | 0.8278 | 0.8305 |
| | Std | 0.0087 | 0.0075 | 0.0115 | 0.0097 | 0.0108 | 0.0076 | 0.0085 | 0.0077 | 0.0083 | 0.0073 |
| EPSO | Mean | 0.7363 | 0.7795 | 0.7958 | 0.8094 | 0.8195 | 0.8218 | 0.8376 | 0.8445 | 0.8569 | 0.8611 |
| | Std | 0.0074 | 0.0083 | 0.0114 | 0.0105 | 0.0077 | 0.0059 | 0.0088 | 0.0073 | **0.0041** | 0.0052 |
| DBSCAN | Mean | 0.6109 | 0.6528 | 0.6792 | 0.6877 | 0.7031 | 0.7185 | 0.7225 | 0.7298 | 0.7346 | 0.7388 |
| | Std | 0.0102 | 0.0108 | 0.0078 | 0.0082 | 0.0065 | 0.0083 | 0.0095 | 0.0115 | 0.0086 | 0.0069 |
| GDBSCAN | Mean | 0.6232 | 0.6755 | 0.6947 | 0.6994 | 0.7196 | 0.7226 | 0.7359 | 0.7431 | 0.7568 | 0.7605 |
| | Std | 0.0098 | 0.0076 | 0.0071 | 0.0062 | 0.0098 | 0.0057 | 0.0094 | 0.0108 | 0.0074 | 0.0068 |
| BIRCH | Mean | 0.5389 | 0.5705 | 0.5963 | 0.6193 | 0.6258 | 0.6377 | 0.6425 | 0.6499 | 0.6526 | 0.6615 |
| | Std | 0.0124 | 0.0098 | 0.0074 | 0.0126 | 0.0109 | 0.0073 | 0.0068 | 0.0077 | 0.0071 | 0.0105 |
| CLARANS | Mean | 0.5402 | 0.5917 | 0.6158 | 0.6351 | 0.6408 | 0.6479 | 0.6504 | 0.6682 | 0.6794 | 0.6823 |
| | Std | 0.0081 | 0.0095 | 0.0077 | **0.0054** | **0.0061** | 0.0093 | 0.0117 | 0.0079 | 0.0083 | 0.0092 |
| GA | Mean | 0.5525 | 0.6004 | 0.6221 | 0.6387 | 0.6429 | 0.6554 | 0.6627 | 0.6763 | 0.6841 | 0.6909 |
| | Std | 0.0068 | 0.0074 | **0.0052** | 0.0059 | 0.0091 | 0.0127 | 0.0082 | 0.0088 | 0.0065 | 0.0112 |
| SA | Mean | 0.5769 | 0.6211 | 0.6387 | 0.6401 | 0.6535 | 0.6592 | 0.6675 | 0.6852 | 0.6968 | 0.7055 |
| | Std | 0.0133 | 0.0114 | 0.0083 | 0.0064 | 0.0081 | 0.0095 | 0.0086 | 0.0105 | 0.0074 | 0.0108 |

The best results are highlighted in bold.

Tables 6, 7 and Fig. 8 show the comparisons of clustering parameters calculated by different optimization algorithms in the framework of MUIP clustering. Table 6 lists the optimal values of σ. We see clearly that EODPSO clustering method gives the best optimal values and has better stability than other clustering methods. Furthermore, we note that since different ratios of labeled data influence the values of the fuzzy partition matrix and cluster centers, the values of the optimized σ vary with the number of labeled data.

**Table 6.** Comparison of optimal values of σ

| Clustering method | | Ratio of labeled data (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **10** | **20** | **30** | **40** | **50** | **60** | **70** | **80** | **90** | **100** |
| EODPSO | Mean | **1.3347** | **1.3339** | **1.3327** | **1.3331** | **1.3325** | **1.3323** | **1.3319** | **1.3315** | **1.3312** | **1.3309** |
| | Std | **0.0014** | **0.0017** | 0.0035 | 0.0024 | **0.0022** | **0.0019** | **0.0015** | **0.0023** | 0.0032 | **0.0021** |
| PSO | Mean | 2.1367 | 2.1362 | 2.1355 | 2.1348 | 2.1343 | 2.1341 | 2.1335 | 2.1332 | 2.1329 | 2.1324 |
| | Std | 0.0053 | 0.0059 | 0.0042 | 0.0035 | 0.0027 | 0.0031 | 0.0022 | 0.0059 | 0.0026 | 0.0034 |
| DEPSO | Mean | 1.9621 | 1.9615 | 1.9576 | 1.9497 | 1.9471 | 1.9425 | 1.9417 | 1.9415 | 1.9411 | 1.9408 |
| | Std | 0.0035 | 0.0024 | 0.0035 | 0.0047 | 0.0024 | 0.0039 | 0.0026 | 0.0025 | 0.0058 | 0.0047 |
| CPSO | Mean | 1.7553 | 1.7544 | 1.7541 | 1.7537 | 1.7532 | 1.7524 | 1.7519 | 1.7514 | 1.7512 | 1.7508 |
| | Std | 0.0032 | 0.0019 | 0.0042 | **0.0023** | 0.0029 | 0.0024 | 0.0067 | 0.0053 | **0.0022** | 0.0032 |
| CAPSO | Mean | 1.6455 | 1.6349 | 1.6338 | 1.6317 | 1.6234 | 1.6225 | 1.6221 | 1.6319 | 1.6217 | 1.6216 |
| | Std | 0.0027 | 0.0024 | 0.0073 | 0.0062 | 0.0033 | 0.0025 | 0.0023 | 0.0036 | 0.0054 | 0.0043 |
| EPSO | Mean | 1.4718 | 1.4632 | 1.4621 | 1.4583 | 1.4565 | 1.4539 | 1.4533 | 1.4527 | 1.4523 | 1.4521 |
| | Std | 0.0018 | 0.0047 | **0.0024** | 0.0068 | 0.0032 | 0.0051 | 0.0039 | 0.0028 | 0.0037 | 0.0026 |
| GA | Mean | 2.4657 | 2.4633 | 2.4601 | 2.4575 | 2.4557 | 2.4538 | 2.4527 | 2.4518 | 2.4515 | 2.4513 |
| | Std | 0.0021 | 0.0019 | 0.0035 | 0.0032 | 0.0058 | 0.0042 | 0.0039 | 0.0026 | 0.0027 | 0.0034 |
| SA | Mean | 2.2796 | 2.2771 | 2.2753 | 2.2695 | 2.2649 | 2.2573 | 2.2569 | 2.2557 | 2.2553 | 2.2551 |
| | Std | 0.0034 | 0.0057 | 0.0051 | 0.0048 | 0.0052 | 0.0026 | 0.0029 | 0.0035 | 0.0031 | 0.0027 |

The best results are highlighted in bold.

**Table 7.** Comparison of number of clusters

| Clustering method | | Ratio of labeled data (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **10** | **20** | **30** | **40** | **50** | **60** | **70** | **80** | **90** | **100** |
| EODPSO | Mean | **46.23** | **47.35** | **47.41** | **48.34** | **52.2** | **50.13** | **51.00** | **50.67** | **51.00** | **51.00** |
| | Std | 0.58 | **0.41** | **0.26** | 0.35 | **0.27** | **0.35** | **0.00** | **0.23** | **0.00** | **0.00** |
| PSO | Mean | 37.63 | 40.51 | 39.32 | 41.39 | 42.42 | 41.36 | 44.2 | 45.36 | 46.76 | 45.58 |
| | Std | 0.95 | 0.59 | 0.35 | 0.52 | 0.45 | 0.86 | 0.39 | 0.27 | 0.69 | 0.34 |
| DEPSO | Mean | 39.35 | 41.16 | 41.26 | 43.72 | 41.32 | 44.75 | 43.32 | 46.63 | 46.57 | 48.55 |
| | Std | 0.74 | 0.52 | 0.29 | 0.37 | 0.67 | 0.47 | 0.42 | 0.52 | 0.47 | 0.49 |
| CPSO | Mean | 40.5 | 41.37 | 43.39 | 42.05 | 43.24 | 45.17 | 47.35 | 47.26 | 45.82 | 52.14 |
| | Std | 0.67 | 0.55 | 0.34 | 0.41 | 0.84 | 0.38 | 0.68 | 0.49 | 0.58 | 0.68 |
| CAPSO | Mean | 43.15 | 45.84 | 42.35 | 45.41 | 47.19 | 47.63 | 48.29 | 53.74 | 53.06 | 49.3 |
| | Std | **0.54** | 0.48 | 0.38 | 0.56 | 0.33 | 0.46 | 0.35 | 0.38 | 0.31 | 0.36 |
| EPSO | Mean | 43.35 | 41.24 | 45.52 | 56.42 | 46.36 | 47.52 | 53.29 | 49.41 | 47.37 | 52.76 |
| | Std | 0.71 | 0.43 | 0.27 | **0.29** | 0.41 | 0.33 | 0.51 | 0.68 | 0.55 | 0.25 |
| GA | Mean | 35.96 | 31.18 | 32.26 | 38.69 | 40.82 | 34.4 | 34.25 | 35.57 | 33.68 | 38.36 |
| | Std | 0.89 | 0.75 | 0.45 | 0.39 | 0.32 | 0.59 | 0.73 | 0.54 | 0.59 | 0.73 |
| SA | Mean | 37.05 | 37.36 | 35.7 | 36.96 | 38.54 | 35.18 | 36.59 | 37.42 | 37.68 | 39.32 |
| | Std | 0.86 | 0.71 | 0.44 | 0.64 | 0.57 | 0.36 | 0.68 | 0.97 | 0.62 | 0.58 |

The best results are highlighted in bold.

Table 7 shows the comparison of the numbers of clusters. In our experiments, the true number of clusters in MUIP data collection is 51. It is observed that with the increase of labeled data, the numbers of clusters of all clustering methods tend to the true number of clusters. Especially, the obtained numbers of clusters from EODPSO clustering method are the most close to the true number of clusters. Moreover, EODPSO clustering method shows better stability than other clustering methods.

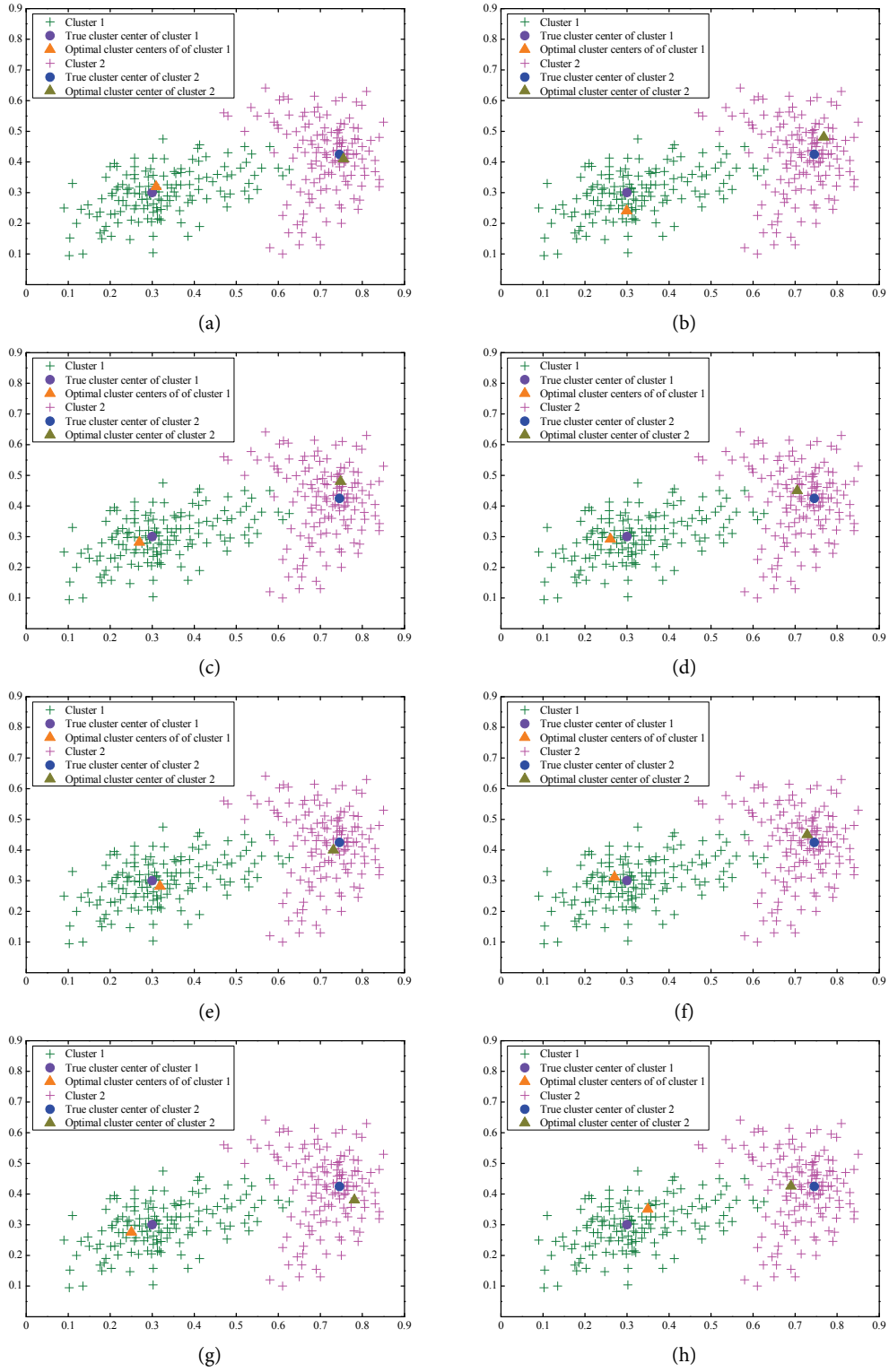


**Fig. 8.** Comparison of initial cluster centers: (a) EODPSO, (b) PSO, (c) DEPSO, (d) CPSO, (e) CAPSO, (f) EPSO, (g) GA, and (h) SA clusterings.

The results of the experiments mentioned above indicate that the overall performance of SSKFCM clustering method increases with an increasing number of labeled data and the influence of SSKFCM clustering method decreases. Thus, in order to compare the results of initial cluster centers, we randomly select 10% data as labeled data and the rest as unlabeled data in our experiments. Fig. 8 shows the comparison of initial cluster centers computed using different optimization algorithms in the framework of MUIP clustering. Two optimal initial cluster centers are generated by different optimization algorithms. We find that the optimal initial cluster centers from EODPSO clustering method are the most close to the true cluster centers compared to other clustering methods.

From Tables 6, 7 and Fig. 8, we find that EODPSO algorithm produces the best results in terms of optimization of clustering parameters, compared to other optimization algorithms. This is because, first, EODPSO algorithm enhances the local search ability and global search ability through CA and the new population search strategy. Second, COL is used to initialize the population of PSO, which avoids the randomness of the initial solutions and makes EODPSO algorithm more stable than other optimization algorithms.

To further analyze the performance of EODPSO algorithm in the framework of MUIP clustering, we compare the convergences of EODPSO, PSO, DEPSO, CPSO, CAPSO, EPSO, GA, and SA clustering methods in the second experiment. According to the analysis of the first experiment, we concentrate on convergence of these clustering methods with a small number of labeled data. Thus, we randomly select 10% labeled data and 90% unlabeled data, and all clustering methods are run 20 times independently on MUIP data. The mean and standard deviation of optimal values of the objective function are shown in Table 8, where the best results are highlighted in bold. It can be seen from Table 8 that EODPSO clustering method produces the best results in finding global optimum and has the best stability, compared to other clustering methods. It is quite understandable because EODPSO algorithm has better global search ability to optimize clustering parameters.

**Table 8.** Mean and standard deviation results of convergence for all SSKFCM methods

| Clustering | EODPSO | PSO | DEPSO | CPSO | CAPSO | EPSO | GA | SA |
|---|---|---|---|---|---|---|---|---|
| Mean | **4.6980** | 5.0784 | 4.9578 | 4.9156 | 4.8455 | 4.7566 | 5.2536 | 5.3352 |
| Std | **0.0043** | 0.0074 | 0.0066 | 0.0054 | 0.0059 | 0.0046 | 0.0072 | 0.0081 |

The best results are highlighted in bold.

The convergence processes of all methods are shown in Fig. 9. Obviously, EODPSO clustering method achieves the global optimal solution at around 40 iterations. By contrast, other clustering methods do not achieve global optimal solutions within 40 iterations. It indicates that the convergence speed of EODPSO clustering method is faster than other methods. The best fitness value is 4.698. In EODPSO algorithm, we use a hybrid strategy to avoid local minima and accelerate convergence speed. Therefore, when falling into local optimum, only EODPSO clustering method can jump out of local optima to continue searching for global optimum.

Through the above comparison, it can be concluded that the proposed clustering method has the best overall performance in terms of clustering effect and convergence. This is because the EODPSO clustering method has the following advantages over other clustering:

(1) Compared to other clustering methods, SSKFCM clustering method is more suitable to cluster MUIP data.
(2) In the framework of MUIP clustering, EODPSO algorithm performs better than PSO, DEPSO, CPSO, CAPSO, EPSO, GA, and SA clustering methods when optimizing the clustering parameters.

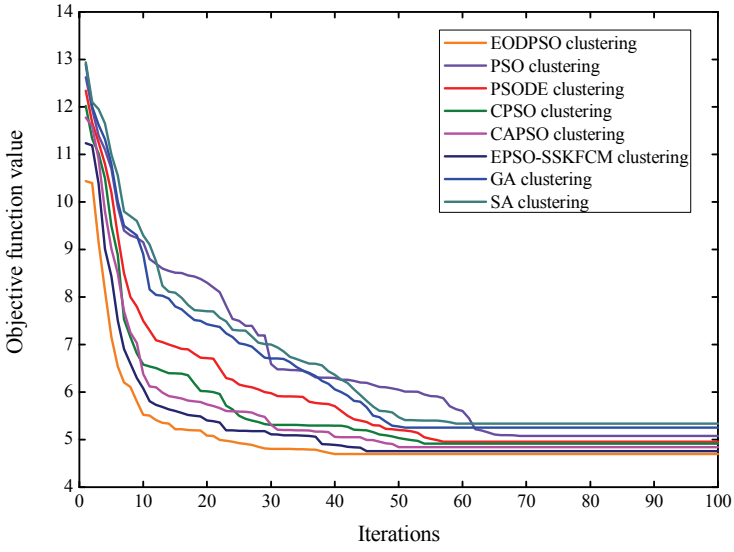


**Fig. 9.** The convergence comparison of all SSKFCM clustering methods.

## 5. Conclusion

In this paper, we propose a MUIP clustering method for the purpose of MUIP data analysis. By studying the advantages and drawbacks of SSKFCM clustering, we apply SSKFCM clustering to cluster MUIP data and utilize PSO algorithm to optimize SSKFCM clustering parameters. Motivated by the recent research on the variant PSO algorithm, we present a hybrid PSO algorithm, namely EODPSO, for the purpose of improving PSO algorithm. In EODPSO algorithm, COL is used to generate the initial population and CA is employed to enhance local search ability. Additionally, to further improve global search ability, EODPSO algorithm makes two improvements. Firstly, an improved intuitionistic fuzzy entropy measure is proposed to accurately measure the diversity of particle swarm. Secondly, a new population search strategy by combing OL and DE is proposed to strengthen the global convergence. The experiments compare the proposed clustering method with other relevant clustering methods on the MUIP data set. The results show that our clustering method achieves better results than other clustering methods in terms of clustering effect and convergence.

In the future work, in order to further improve the performance of our clustering method, we will collect more MUIP data to enrich the data set and strengthen the theoretical analysis of the relationship between labeled and unlabeled. We will use other evaluation metrics to evaluate the performance of our clustering method from different points. Moreover, we will investigate existing retrieval methods and intend to incorporate our clustering method into MUIP selection.

## Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 61272286), the Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20126101110006), Shaanxi Science and Technology Project (No. 2016GY-123), and the Science Research Foundation of Northwest University (No. 15NW31).

## References

[1] T. Wetchakorn and N. Prompoon, "Method for mobile user interface design patterns creation for iOS platform," in *Proceedings of 2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Songkhla, Thailand, 2015, pp. 150-155.

[2] T. D. Nguyen and J. Vanderdonckt, "User interface master detail pattern on Android," in *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, Copenhagen, Denmark, 2012, pp. 299-304.

[3] G. Thongmool and M. Phankokkruad, "Analysis of interaction user interface patterns and usability study in computer assisted instruction for Tablet PC," in *Proceedings of 2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Batu Ferringhi, Malaysia, 2014, pp. 472-477.

[4] P. Gomes, F. C. Pereira, P. Paiva, N. Seco, P. Carreiro, J. L. Ferreira, and C. Bento, "Using CBR for automation of software design patterns," in *Advances in Case-Based Reasoning*. Heidelberg: Springer, 2002, pp. 534-548.

[5] L. Pavlic, V. Podgorelec, and M. Hericko, "A question-based design pattern advisement approach," *Computer Science and Information Systems*, vol. 11, no. 2, pp. 645-664, 2014.

[6] S. M. H. Hasheminejad and S. Jalili, "Design patterns selection: an automatic two-phase method," *Journal of Systems and Software*, vol. 85, no. 2, pp. 408-424, 2012.

[7] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, Portland, OR, 1996, pp. 226-231.

[8] G. Andrade, G. Ramos, D. Madeira, R. Sachetto, R. Ferreira, and L. Rocha, "G-DBSCAN: a GPU accelerated algorithm for density-based clustering," *Procedia Computer Science*, vol. 18, pp. 369-378, 2013.

[9] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, Montreal, Canada, 1996, pp. 103-114.

[10] R. T. Ng and J. Han, "CLARANS: a method for clustering objects for spatial data mining," *IEEE Transactions on Knowledge & Data Engineering*, vol. 14, no. 5, pp. 1003-1016, 2002.

[11] W. Jia, Q. Hua, M. Zhang, R. Chen, and X. Ji, "User interface pattern language based on category theory," *Journal of Computer Aided Design & Computer Graphics*, vol. 29, no. 1, pp. 79-89, 2017.

[12] D. Q. Zhang and S. C. Chen, "Clustering incomplete data using kernel-based fuzzy c-means algorithm," *Neural Processing Letters*, vol. 18, no. 3, pp. 155-162, 2003.

[13] D. Zhang, K. Tan, and S. Chen, "Semi-supervised kernel-based fuzzy c-means," in *Neural Information Processing*. Heidelberg: Springer, 2004, pp. 1229-1234.

[14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* Reading, MA: Addison-Wesley, 1989.

[15] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statistical Science*, vol. 8, no. 1, pp. 10-15, 1993.

[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks* (ICNN), Perth, Australia, 1995, pp. 1942-1948.

[17] S. Yu, Y. M. Wei, J. Fan, X. Zhang, and K. Wang, "Exploring the regional characteristics of inter-provincial CO2 emissions in China: an improved fuzzy clustering analysis based on particle swarm optimization," *Applied Energy*, vol. 92, pp. 552-562, 2012.

[18] A. Mekhmoukh and K. Mokrani, "Improved fuzzy c-means based particle swarm optimization (PSO) initialization and outlier rejection with level set methods for MR brain image segmentation," *Computer Methods and Programs in Biomedicine*, vol. 122, no. 2, pp. 266-281, 2015.

[19] Y. Liu, F. Liu, T. Hou, and X. Zhang, "Kernel-based fuzzy C-means clustering method based on parameter optimization," *Journal of Jilin University (Engineering and Technology Edition)*, vol. 46, no. 1, pp. 246-251, 2016.

[20] A. Sedki and D. Ouazar, "Hybrid particle swarm optimization and differential evolution for optimal design of water distribution systems," *Advanced Engineering Informatics*, vol. 26, no. 3, pp. 582-591, 2012.

[21] R. Storn and K. Price, "Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.

[22] F. Liu and Z. Zhou, "A new data classification method based on chaotic particle swarm optimization and least square-support vector machine," *Chemometrics and Intelligent Laboratory Systems*, vol. 147, pp. 147-156, 2015.

[23] H. J. Lu, H. M. Zhang, and L. H. Ma, "A new optimization algorithm based on chaos," *Journal of Zhejiang University-Science A*, vol. 7, no. 4, pp. 539-542, 2006.

[24] Y. Dai, L. Liu, Y. Li, and J. Song, "An improved particle swarm optimisation based on cellular automata," *International Journal of Computing Science and Mathematics*, vol. 5, no. 1, pp. 94-106, 2014.

[25] J. L. Schiff, *Cellular Automata: A Discrete View of the World*. Hoboken, NJ: John Wiley & Sons, 2007.

[26] Y. Z. Wang, Y. J. Lei, L. Zhou, and R. L. Li, "Intuitionistic fuzzy discrete particle swarm algorithm," *Control and Decision*, vol. 27, no. 11, pp. 1735-1739, 2012.

[27] Y. Wang and Y. J. Lei, "A technique for constructing intuitionistic fuzzy entropy," *Control and Decision*, ol. 22, no. 12, pp. 1390-1394, 2007.

[28] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *Proceedings of International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, Vienna, Austria, 2005, pp. 695-701.

[29] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York, NY: Plenum, 1981.

[30] K. T. Atanassov, "Intuitionistic fuzzy sets," *Fuzzy Sets and Systems*, vol. 20, no. 1, pp. 87-96, 1986.

[31] W. Zeng and H. Li, "Relationship between similarity measure and entropy of interval valued fuzzy sets," *Fuzzy Sets and Systems*, vol. 157, no. 11, pp. 1477-1484, 2006.

[32] J. Li, G. Deng, H. Li, and W. Zeng, "The relationship between similarity measure and entropy of intuitionistic fuzzy sets," *Information Sciences*, vol. 188, pp. 314-321, 2012.

[33] R. Verma and B. D. Sharma, "Exponential entropy on intuitionistic fuzzy sets," *Kybernetika*, vol. 49, no. 1, pp. 114-127, 2013.

[34] C. P. Wei, Z. H. Gao, and T. T. Guo, "An intuitionistic fuzzy entropy measure based on trigonometric function," *Control and Decision*, vol. 27, no. 4, pp. 571-574, 2012.

[35] M. M. Gao, T. Sun, and J. J. Zhu, "An revised axiomatic definition and structural formula of intuitionistic fuzzy entropy," *Control and Decision*, vol. 29, no. 3, pp. 470-474, 2014.

[36] M. F. Liu and H. P. Ren, "A study of multi-attribute decision making based on a new intuitionistic fuzzy entropy measure," *System Engineering Theory and Practice*, vol. 35, no. 11, pp. 2909-2916, 2015.

[37] E. Szmidt and J. Kacprzyk, "Entropy for intuitionistic fuzzy sets," *Fuzzy Sets and Systems*, vol. 118, no. 3, pp. 467-477, 2001.
[38] K. Guo and Q. Song, "On the entropy for Atanassov's intuitionistic fuzzy sets: an interpretation from the perspective of amount of knowledge," *Applied Soft Computing*, vol. 24, pp. 328-340, 2014.
[39] W. F. Gao and S. Y. Liu, "A modified artificial bee colony algorithm," *Computers & Operations Research*, vol. 39, no. 3, pp. 687-697, 2012.
[40] Y. Zhou, L. Bao, and C. P. Chen, "A new 1D chaotic system for image encryption," *Signal Processing*, vol. 97, pp. 172-182, 2014.
[41] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 224-227, 1979.
[42] M. R. Rezaee, B. P. Lelieveldt, and J. H. Reiber, "A new cluster validity index for the fuzzy c-mean," *Pattern Recognition Letters*, vol. 19, no. 3-4, pp. 237-246, 1998.
[43] T. Neil, *Mobile Design Pattern Gallery: UI Patterns for Smartphone Apps*. Sebastopol, CA: O'Reilly, 2012.
[44] S. Hoober and E. Berkman, *Designing mobile interfaces*. Sebastopol, CA: O'Reill, 2012.
[45] W. Y. Chen, Y. Song, H. Bai, C. J. Lin, and E. Y. Chang, "Parallel spectral clustering in distributed systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no.3, pp. 568-586, 2010.
[46] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193-218, 1985.
[47] A. Strehl and J. Ghosh, "Cluster ensembles: a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583-617, 2002.
[48] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

**Wei Jia** https://orcid.org/0000-0002-3170-8269

He is an associate professor at the Xinhua College, Ningxia University, China. He is currently working towards his PhD degree in the School of Information Science and Technology from Northwest University, China. His research interests include human-computer interaction and user interface engineering.

**Qingyi Hua** https://orcid.org/0000-0002-2101-7579

He is a Professor and PhD Supervisor at the School of Information Science and Technology, Northwest University, China. His research interests include human-computer interaction and user interface engineering.

**Minjun Zhang** https://orcid.org/0000-0002-5776-0823

He is currently working towards his PhD degree in the School of Information Science and Technology from Northwest University, China. His research interests include human-computer interaction and user interface engineering.

**Rui Chen** https://orcid.org/0000-0003-1169-7678

He is currently working towards his PhD degree in the School of Information Science and Technology from Northwest University, China. His research interests include human-computer interaction and user interface engineering.

**Xiang Ji** https://orcid.org/0000-0003-4427-1171

She received her B.S. degree in Computer Application from Northwest University in 2001, received her M.S. degree in Computer Application Technology from Xidian University in 2005. She now is studying for a PhD in Northwest University. Her main research interests include wireless sensor network and human-computer interaction.

**Bo Wang** https://orcid.org/0000-0002-0524-9097

He is a lecturer at the School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, China. He is currently working towards his PhD degree in the School of Information Science and Technology from Northwest University, China. His research interests include human-computer interaction and user interface engineering.