Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 4195318, 15 pages
https://doi.org/10.1155/2019/4195318

Research Article

Memetic Variable Clustering and Its Application

JiaCheng Ni [1,2] **and Li Li** [1,3]

[1]*College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China*
[2]*TRIGr China, OCTO, DELL EMC, Shanghai 200443, China*
[3]*Shanghai Institue of Intelligent Science and Technology, Tongji University, Shanghai 201804, China*

Correspondence should be addressed to Li Li; lili@tongji.edu.cn

Received 5 June 2019; Revised 24 September 2019; Accepted 29 October 2019; Published 18 November 2019

Academic Editor: Pietro Bia

Clustering analysis is an important and difficult task in data mining and big data analysis. Although being a widely used clustering analysis technique, variable clustering did not get enough attention in previous studies. Inspired by the metaheuristic optimization techniques developed for clustering data items, we try to overcome the main shortcoming of *k*-means-based variable clustering algorithm, which is being sensitive to initial centroids by introducing the metaheuristic optimization. A novel memetic algorithm named MCLPSO (Memetic Comprehensive Learning Particle Swarm Optimization) based on CLPSO (Comprehensive Learning Particle Swarm Optimization) has been studied under the framework of memetic computing in our previous work. In this work, MCLPSO is used as a metaheuristic approach to improve the *k*-means-based variable clustering algorithm by adjusting the initial centroids iteratively to maximize the homogeneity of the clustering results. In MCLPSO, a chaotic local search operator is used and a simulated annealing- (SA-) based local search strategy is developed by combining the cognition-only PSO model with SA. The adaptive memetic strategy can enable the stagnant particles which cannot be improved by the comprehensive learning strategy to escape from the local optima and enable some elite particles to give fine-grained local search around the promising regions. The experimental result demonstrates a good performance of MCLPSO in optimizing the variable clustering criterion on several datasets compared with the original variable clustering method. Finally, for practical use, we also developed a web-based interactive software platform for the proposed approach and give a practical case study—analyzing the performance of semiconductor manufacturing system to demonstrate the usage.

1. Introduction

Clustering analysis or clustering is the task of grouping a set of objects in such a way that, according to certain similarity, objects in the same group (called a cluster) are more similar than objects falling in different groups (clusters). Clustering analysis is used widely in the data preprocessing step and data mining step of KDD (Knowledge Discovery in Databases, KDD) [1] (Figure 1), and especially it is the main task of exploratory data mining and unsupervised machine learning. Recently, clustering analysis is pointed out as a powerful metalearning to accurately analyze the big data [2]. Clustering analysis also plays an important role in many other fields, including pattern recognition, image analysis, information retrieval, and bioinformatics. Besides its importance, clustering analysis is also a challenging task because the unsupervised nature of clustering analysis implies that the structural characteristics of the dataset are not known, except if there is some domain knowledge about the dataset available in advance [3].

Because of the importance and difficulty of clustering analysis, a lot of clustering algorithms are proposed in the literature. Some popular clustering algorithms, for example, *k*-means clustering, suffer from the shortcoming that is being sensitive to outliers; therefore, metaheuristic methods such as evolutionary algorithms and swarm intelligence algorithms are used widely to improve the clustering algorithms from the optimization perspective. Almost all the metaheuristic based improvements for clustering algorithms in the literature are devoted to cluster the data items, but clustering analysis for variables is also a common technique

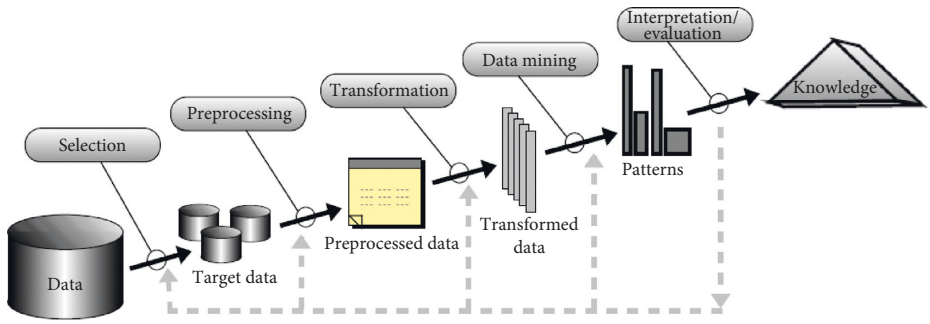

Figure 1: Steps of KDD process.

for statistical data analysis for dimension reduction or (unsupervised) feature selection especially in practical statistical data analysis activities. The most famous one is the VARCLUS procedure in SAS, and there are also some other versions of variable clustering methods implemented in R and SPSS. In contrast to its wide application, the contributions of research to the variable clustering techniques are not sufficient. Also, the k-means-based variable clustering algorithms suffer from the same shortcoming that is being sensitive to initial centroids.

We studied the metaheuristic approach for variable clustering algorithm based on our previous work. In our previous research, MCLPSO [4] is studied to improve CLPSO [5] from two aspects: one is the chaotic local search and the other is the SA-based local search. Firstly, we integrate the chaotic local search operator to CLPSO to enable the stagnant particles to escape from the local optima. An SA-based local search operator combined with the "cognition-only" model is developed to enhance the local search ability of the elite members. The experimental results demonstrate that MCLPSO is competitive in optimizing the multimodal functions. In this work, MCLPSO is reorganized under a novel metaheuristic paradigm-memetic computing. Furthermore, MCLPSO is used to optimize k-means-based variable clustering algorithm as a metaheuristic approach. The experimental results demonstrate that MCLPSO can improve the k-means based variable clustering algorithm effectively. We also developed a web-based interactive software platform to implement this approach and give a practical case study—analyzing the performance of a semiconductor manufacturing system by MCLPSO-based variable clustering.

The main contribution of this work includes the following:

(i) A novel memetic algorithm MCLPSO proposed in our previous research is described under a more sound and general theoretical framework-memetic computing.

(ii) To our best knowledge, it is the first time to use metaheuristic method to improve the results for variable clustering. Also, the improved variable clustering is used to deal with some complex tasks.

(iii) To facilitate the practical use of the MCLPSO-based variable clustering algorithm, we developed an interactive software system for this approach and give a real-world case study.

The rest of the paper is organized as follows: In Section 2, we review the related work. In Section 3, we describe our previous work MCLPSO in detail under the memetic computing framework. In Section 4, MCLPSO is used to optimize the k-means-based variable clustering problem. In Section 4, some experimental results on several datasets are presented and discussed. In Section 5, a web-based interactive software system developed for clustering variables is introduced. Finally, we give a final conclusion in Section 6.

2. Related Work

As mentioned in Section 1, clustering analysis is an important and difficult task. In the literature, dozens of clustering algorithms are proposed for multiple clustering analysis applications. These clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods [6]. Despite the classification of methods, the main objective of a clustering algorithm is maximizing both the homogeneity within each cluster and the heterogeneity among different clusters [6]. From optimization perspective, if the homogeneity within each cluster and the heterogeneity among different clusters can be measured by a certain clustering criterion, the metaheuristic algorithms including EA (Evolutionary Algorithms, EA) such as GA (Genetic Algorithms, GA) and swarm intelligence algorithms such as PSO (Particle Swarm Optimization, PSO) can be applied to improve the clustering results by adjusting the hyperparameters for those clustering algorithms which are sensitive to their hyperparameters.

As a best-known and most commonly used clustering technique, k-means clustering suffers from the deficiency that is being sensitive to its k value and initial centroids. In the literature, several metaheuristic-based clustering methods are proposed to overcome this deficiency. Maulik and Bandyopadhyay proposed a GA-based clustering technique to exploit the searching capability of genetic algorithms so that the clustering metric can be optimized by searching for appropriate cluster centroids [7]. Van der Merwe and Engelbrecht introduced two PSO-based clustering algorithms [8]. In PSO-based clustering algorithm, PSO is used to find the optimum centroids directly. In hybrid PSO and k-means clustering algorithm, the result of k-means is used to initialize PSO-based clustering for quick convergence. The results of algorithm were compared to k-

means algorithm and the conclusion is that the proposed approaches gave better convergence and low quantization error in comparison to k-means algorithm. Esmin improved PSO-based clustering algorithm by modifying the evaluation function and the modification brought good improvements to the clustering results [9]. Ahmadyfard proposed a two-stage clustering algorithm [10] in which PSO is used at the first stage to find optimum centroids directly; then these optimized centroids are used to initialize k-means at the second stage. The combined method has the advantage of both PSO and k-means methods if the algorithm switch to k-means when the PSO are closed to the global optimum.

Recently, memetic algorithms are used as a novel metaheuristic paradigm to improve clustering algorithms. A memetic algorithm (MA) is an EA that includes one or more local search operators to improve the individuals within its evolution cycles [11]. In MAs, "memes" refer to the local search operators which are used to enhance local search ability of EAs [12]. Moscato introduced the concept of meme to EA firstly by combining the SA with the crossover operator in the genetic algorithm to solve the TSP (Travelling Salesman Problem, TSP) problem [13]. MA is inspired by the concept of a meme, which represents a unit of cultural evolution that can exhibit local refinement. The population evolution is cooperated with the individual learning, and the memetic model is a more detailed explanation for the adaption in the natural system than the genetic model [12]. Most EAs can find the regions around the local optima, but some versions of EA including PSO exhibit the deficiency of lacking local search abilities. MA is proposed to overcome this deficiency. The promising regions throughout the search space can be found by global search operators and the local search operators can give fine-grained search around these search regions [12]. The global search cooperates with the local search to find the global optima. Ong extends the notion of MA and defines memetic computation (MC) [14]. The concept of memes used in MC is more general than the concept of memes used in MA. In MC, a meme can denote a learning strategy, an operator or a local search procedure.

Sheng proposed an approach for simultaneous clustering and feature selection using a niching memetic algorithm, NMA_CF [15]. In NMA_CF, encode both feature selection and cluster centers with different numbers of clusters; local search operations are introduced to refine feature selection and cluster centers encoded in the chromosomes; and niching method is integrated to preserve the population diversity and prevent premature convergence. The experimental results demonstrated that simultaneous global clustering and feature subset optimization mechanism is effective in approaching the problem. Recently Sheng improved NMA_CF by introducing multiple local search operation and adaptive niching strategy [16]. In our previous research [17], we proposed a novel memetic algorithm GS-MPSO and use GS-MPSO to optimize the initial centroids for k-means clustering. In GS-MPSO, k-means clustering algorithm is integrated into function evaluation so that the improvement of clustering results is significant.

Although most clustering algorithms are devoted to cluster data items, variable clustering is also a widely used technique in practical statistical analysis activities. The function of variable clustering is provided in almost all the statistical tools such as R, SPSS, and SAS. The most famous one is the VARCLUS procedure implemented in SAS. In VARCLUS, the similarity of variables is measured by Pearson correlation, and the centroid is computed by the first principal component of the variables in the cluster. The variables are clustered to hierarchical clusters by hierarchical clustering. Almost in all the variable clustering algorithms, PCA (Principal Component Analysis, PCA) is used to compute the representative of variables in a cluster. Vigneau proposed a variable clustering algorithm named Clustering around Latent Variables to segment quantitative variables [18, 19]. Chavent proposed ClustOfVar to cluster variables with mixed type [20]. In [20], PCAMIX is used to calculate centroids and a k-means-based variable clustering and a hierarchical variable clustering are studied to optimize the homogeneity criterion.

3. Memetic Comprehensive Learning PSO

In many applications, the MAs are more competitive both in effectiveness and efficiency than the traditional EAs. But the method of designing an MA with a good performance is intricate. To design a competitive MA, the local search components should be kept in balance with the global search component to achieve a balance between exploration and exploitation. In some MAs, the excessive use of the local search can lead to a loss of diversity in the population. If the local search is applied to the candidate which is a local optimum or the local search depth is too high, the computing time may be wasted because of the unnecessary local search. The local search operators should cooperate with the evolutionary operators to find a balance between global search and local search. Therefore, the following design and parameterization issues of MA are considered [21]:

(i) How often should local search be applied?

(ii) On which solutions should local search be used?

(iii) How long should local search be run?

The memetic strategy used in MCLPSO will give answers to the design issues of MA. We propose an adaptive memetic strategy based on the status and quality of particles.

Although some MAs have been proved to be effective, the framework of MA was found too specific to describe some complex hybrid algorithms. Some researchers try to develop a more general and more formal definition for MA. For example, Nguyen presents a probabilistic memetic framework to model the process of MA [22]. Ong defines memetic computation as a paradigm that uses the notion of meme(s) as units of information encoded in computational representations for problem-solving [14]. A MC is composed of several interactive memes and MC uses these memes to solve the complex problems. In MC, a meme can denote an operator, a learning strategy, or a local search procedure, so the concept of memes used in MC is extended. Icca gave a thorough analysis for MC and introduced

"Ockham's Razor" theorem which is stated as "Entities should not be multiplied unnecessarily" [23]. Icca pointed out that simplicity will help to design an efficient and compact memetic computational algorithm from the perspective of Ockham's Razor theorem and summarized that four kinds of memes perform different exploration in MC:

(i) Stochastic long-distance exploration

(ii) Stochastic moderate-distance exploration

(iii) Deterministic short-distance exploration

(iv) Random long-distance exploration

In our previous work, we have developed some novel memetic algorithms under the framework of MC and theses memetic algorithms are applied to data clustering [17] and missing data estimation [24]. In this work, we develop MCLPSO by following the analysis of MC in [14] and design the following "memes" as in [24].

(i) Stochastic long-distance exploration: comprehensive learning strategy

(ii) Stochastic moderate-distance exploration: chaotic local search

(iii) Deterministic short-distance exploration: SA local search

The diversity can benefit from random long-distance exploration. But random long-distance exploration may lower the quality of swarm when the comprehensive learning strategy is used. So random long-distance exploration is disabled in MCLPSO to keep the swarm stable.

Based on the above discussion, we will discuss the memes used in MCLPSO in detail and propose the memetic strategy for MCLPSO.

3.1. Classification of the Particles. In MCLPSO, the CLPSO is responsible for the global search. The chaotic local search operator is applied to the stagnant particle to improve the stagnant particle and the SA-based local search operator performs fine-grained search around the promising regions.

At each iteration of CLPSO, the ith particle's solution x_i will be updated by adding a velocity v_i which is calculated by learning from $pbest_{fi(d)d}$ at each dimension d. $pbest_j$ is the best solution found by the jth particle so far. $f_i = [f_i(1), f_i(2), \ldots, f_i(D)]$ defines the ith particle's corresponding learning exemplars at each dimension. Some variables are introduced to classify the particles for the purpose of designing an adaptive memetic strategy. The classification depends on the searching status of the particle.

(i) For the ith particle, flag_i is used to record the number of generations the ith particle has not improved its $pbest_i$. If $\text{flag}_i \geq m$, f_i is reassigned and flag_i is reset to 0. m is the refreshing gap and set at 7 [5].

(ii) For the ith particle, stagnant_i is used to record the number of reassignments of f_i and the $pbest_i$ has not been improved during this period, i.e., $pbest_i$ has not been improved for $m * \text{stagnant}_i + \text{flag}_i$ generations. If $\text{stagnant}_i \geq \text{stagnant}_{\max}$, the particle i is stagnant.

(iii) For the ith particle, improve_i is used to record the number of generations that the $pbest_i$ has been improved continuously, i.e., $pbest_i$ has been changed continuously for improve_i generations.

(iv) A particle i with the best $pbest_i$ in the population is a promising particle if $\text{improve}_i \geq \text{improve}_{\max}$.

3.2. Stochastic Long-Distance Exploration-Comprehensive Learning Strategy. The CLPSO is adapted from the original PSO by using a novel velocity updating equation (1) which is called comprehensive learning strategy:

$$v_{id} = w * v_{id} + c * \text{rand}() * \left(p\text{best}_{fi(d)d} - x_{id}\right), \quad (1)$$

where w is the inertia weight, c is the weight of comprehensive learning, $\text{rand}()$ will generate a random number in [0, 1] according to the uniform distribution, and $f_i = [f_i(1), f_i(2), \ldots, f_i(D)]$ defines the ith particle's corresponding learning exemplars at each dimension. At dth dimension, the ith particle should follow $pbest_{fi(d)d}$ which denotes the dth value of the $f_i(d)$ th particle's best solution found so far. Pc_i is the probability that the ith particle will learn from other particles' $pbest$ which is empirically defined as

$$Pc_i = 0.05 + 0.45 * \frac{(\exp(10(i-19)/ps - 1) - 1)}{(\exp(10) - 1)}, \quad (2)$$

where ps means the population size.

The selection of learning exemplars of the ith particle can be implemented by the following steps. For each dimension of the ith particle, a random number is generated between 0 and 1 according to the uniform distribution. If this random number is larger than Pc_i, the corresponding dimension will learn from its own $pbest_i$, otherwise it will learn from another particle's $pbest$ and then two particles in the swarm which excludes the ith particle are chosen randomly and the one with a better $pbest$ will be selected as the exemplar for particle i to learn at that dimension. This process is summarized in Figure 2. For efficiency, the ith particle is allowed to refresh its learning exemplars f_i until the particle ceases improving for m generations and m is called the refreshing gap.

In the CLPSO, each particle will learn from $pbest_{fi}$ which is derived from different particles' historical best position. The updating strategy (1) is proved to yield a larger potential search space than that of the original PSO by the analysis of search behavior [5]. The swarm's diversity can be kept by the comprehensive learning strategy. Therefore, the performance is improved when solving complex multimodal problems. But this improvement is obtained at the cost of the convergence speed because the effect of the current global best position is weakened. If all the particles share the similar $pbest$ with the current global best position, the comprehensive learning is not able to enable the swarm to escape from the local optimum. As other EAs, CLPSO also lacks of the ability of local search. In this study, the CLPSO is investigated under the framework of the MC. Two local search operators are introduced to overcome these deficiencies.

```
procedure Refresh_learning_exemplar(particle i)
begin
  for d = 1 to D
    if(rand() < Pc_i)
      randomly choose particle j and particle k
      from the population excluding particle i
      if(f(pbest_j < f(pbest_k))
        f_i(d) = j
      else f_i(d) = k
      endif
    else f_i(d) = i
    endif
  endfor
end
```

Figure 2: Pseudocode of choosing particle i's learning exemplars.

3.3. Stochastic Moderate-Distance Exploration—Chaotic Local Search. We study the chaotic local search operator to improve the stagnant particle i which cannot improve its $pbest_i$ by comprehensive learning strategy. The *Chaotic_local_search* is adapted from the chaotic local search operator in [25]. The logistic equation (3) is used to generate the chaotic sequence. In (3), μ is the control factor and x is the chaotic variable. Although (3) is deterministic, it exhibits chaotic dynamics when $\mu = 4$ and $x_k \notin \{0, 0.25, 0.5, 0.75, 1\}$. So, (4) is used to generate the chaotic sequence for the dth dimension of particle i. The sequence generated by (4) is sensitive to the initial value. A minute difference in the initial value of the chaotic variable would result in a considerable difference in its long behavior. Equation (5) is used to normalize the initial value of chaotic variable in (4). The stagnant particle i is perturbed with probability P_{Chaotic} by the denormalized value of a chaotic variable. The denormalized value is derived from (6):

$$x_{k+1} = \mu * x_i\left(1 - x_k\right), \quad 0 \le x_0 \le 1, \tag{3}$$

$$cx_{id}^{k+1} = 4cx_{id}^{k}\left(1 - cx_{id}^{k}\right), \quad d = 1, 2, \ldots, D, \tag{4}$$

$$cx_{id}^{k} = \frac{x_d^k - x_{\min,d}}{x_{\max,d} - x_{\min,d}}, \quad d = 1, 2, \ldots, D, \tag{5}$$

$$x_{id}^{k} = x_{\min d} + cx_{id}^{k}\left(x_{\max,d} - x_{\min,d}\right), \quad d = 1, 2, \ldots, D. \tag{6}$$

In *Chaotic_local_search*, x_i is reset by $pbest_i$ and then x_i is normalized between 0 and 1 by (4) to initialize the chaotic vector cx_i. $[x_{\min,d}, x_{\max,d}]$ is the range of the dth dimension of the search space. A chaotic sequence is generated for each dimension by (5) and cx_{id} is the chaotic variable for the dth value of particle i. k is the iteration number. cx_{id} evolves by (5) iteratively, and the track of cx_{id} during the evolution can travel ergodically over the whole search space. During the evolution process of the chaotic variables, the position x_i is perturbed with probability P_{Chaotic} by $x_{ir}^{k'}$ to escape from the local optimum. $x_{ir}^{k'}$ is denormalized from cx_{ir}^{k}. The details of *Chaotic_local_search* are described in Figure 3. *Chaotic_ls_length* represents the number of iterations.

```
procedure Chaotic_local_search (particle i)
begin
    for d = 1 to D
        pos_id = pbestid
        cx_id^0 is initialized by (5)
    endfor
    pos_i' = pos_i
    for k = 1 to Chaotic_ls_length
        for d = 1 to D
            cx_id^k is computed by (4)
        endfor
        for r = 1 to D
            if(rand() < P_Chaotic)
                pos_ir^k' is derived from cx_ir^k by (6)
                replace the rth value of pos_i' by pos_ir^k'
            endif
        endfor
        if(f(pos_i') < f(pos_i))
        then
            pos_i is updated by pos_i'
            pbest_i is updated by pos_i'
        endif
    endfor
    for d = 1 to D
        v_id = 0
    endfor
    stagnant_i = 0
    flag_i = 0
end
```

Figure 3: Pseudocode of *Chaotic_local_search*.

3.4. Deterministic Short-Distance Exploration—SA Based Local Search. CLPSO is used as the global search component for MCLPSO because the diversity can be kept by comprehensive learning. But the lack of ability to local refinement in CLPSO can lead to missing the local optima. To solve this problem, a novel local search operator by combining the cognition-only model [26] with SA is developed in our previous work [17] to enhance the local search ability of the CLPSO. The details of this SA-based local search operator are described in Figure 4.

In Figure 4, T is the temperature variable and T_0 is the initial temperature. *SA_ls_length* represents the number of iterations. $pbest_i'$ can be obtained by introducing a Cauchy perturbation to the rth dimension of $pbest_i$ according to (7) in which $[A_r, B_r]$ is the range of *the* rth parameter and u is generated randomly subject to the uniform distribution between 0 and 1. $pbest_i$ is perturbed with a probability P_{SA} each time for the purpose of "fine-grained" local search around the promising regions. The $pbest_i$ is updated in a greedy way, but the new position x_i' which is generated by the cognition-only model is accepted subject to the Metropolis rule (Kirkpatrick, 1983). A local search around the promising region can be performed. Thus, the ability of local refinement of PSO can be enhanced by the *SA_local_search*.

$$pbest_{ir}' = pbest_{ir} + T \operatorname{sgn}(u - 0.5)\left[\left(\frac{1+1}{T}\right)^{|2u-1|} - 1\right] \cdot \left(B_r - A_r\right), \quad pbest_{ir}' \in [A_r, B_r]. \tag{7}$$

```
procedure SA_local_search (particle i)
begin
  initilization T = T_0
  repeat ls_length times
    pbest_i' = pbest_i
    for r = 1 to D
        if(rand() < P_SA)
            replace rth dimension of pbest_i by (7)
        endif
    endfor
    if(f(pbest_i') < f(pbest_i))
        pbest_i is updated by pbest_i'
        v_i is updated by v_id = w*v_id + c_1*rand()*(pbest_id' − pos_id)
        pos_i' is generated by pos_id' = pos_id + v_id
        Δf = f(pos_i') − f(pos_i)
        if(Δf < 0)
            pos_i is updated by pos_i'
        else if(rand() < exp(−Δf/T))
            pos_i is updated by pos_i'
        endif
        if(f(pos_i') < f(pbest_i))
            pbest_i is updated by pos_i'
        endif
    endif
    T = T_0/lg (1 + j)
  endrepeat
  reset v_i and improve_i to 0
end
```

Figure 4: Pseudocode of SA_local_search.

3.5. Adaptive Memetic Strategy for CLPSO. MCLPSO can be presented as a combination of the CLPSO with *Chaotic_local_search* and *SA_local_search*. The memetic strategy used in MCLPSO can be described as follows:

Adaptive Memetic Strategy 1: *SA_local_search* is only applied to the promising particle to give fine-grained local search around the promising regions, and the *Chaotic_local_search* should be applied to the stagnant particle which cannot improve its own *pbest* by the comprehensive learning strategy to enable the stagnant particles to escape from the local optima.

Although, in some other MAs, the local search is applied to all particles, we adopted *Adaptive Memetic Strategy 1* in the MCLPSO because of the high cost of local search, and the frequent application of local search will result in a disastrous loss in diversity. In *Adaptive Memetic Strategy 1*, the swarm evolves along with the local refinement around the promising regions and the chaotic local search of the stagnant particles. A pseudocode for MCLPSO is described in Figure 5.

Adaptive Memetic Strategy 1 can give answers to two of the design and parameterization issues mentioned in the last section. The local search operators will be applied adaptively according to current particle's quality and status. The *SA_local_search* is always applied to the promising candidate solutions and the *Chaotic_local_search* is always applied to a stagnant particle which cannot improve its *pbest* by the comprehensive learning strategy. For the third question, the depth of *Chaotic_local_search* and *SA_local_search*, we believe a moderate value of *SA_ls_length* is sufficient for *SA_local_search* to find the local optimum because the local search is always applied to the

```
procedure MCLPSO()
begin
for each particle i
    initialize each particle
end
do
    for each particle i
     if(flag_i = m)then
         call the procedure Refresh_learning_exemplar (i)
     endif
     calculate particle velocity according equation (1)
     update particle position by x_id = x_id + v_id
     if the f(pos_i) < f(pbest_i)
     then pbest_i = pos_i
       flag_i = 0
       improve_i++
       no_improve_i = 0
     else
       flag_i++
       no_improve_i++
       improve_i = 0
     endif
    endfor
     for each particle i
     if i is stagnant&rand() < prob_chaotic
     then call the procedure Chaotic_local_search(i) (Figure 2)
     else if i is promising
     then call the procedure SA_local_search(i) (Figure 3)
     endif
    endfor
    choose the particle i with the best f(pbest_i) as gbest
while maximum iterations is not attained
output the f(gbest)
end
```

Figure 5: Pseudocode of MCLPSO.

particles with high quality and the value of *Chaotic_ls_length* is set a same value to balance the exploration and exploitation.

In MCLPSO, the velocity of particle i is restrained by $\min(v_{\max,d}, \max(v_{\min,d}, v_{id}))$ within $[v_{\min,d}, v_{\max,d}]$ which is the range of the dth velocity value. And, $f(x_i)$ is evaluated only if x_i is inside the search bounds. All the *pbest* are kept inside the search bounds, and the particle will be attracted back to the search bounds by the learning exemplars.

4. Clustering of Variables Based on MCLPSO

The main objective of this work is to improve k-means-based variable clustering algorithm by MCLPSO. As mentioned in section1, the k-means clustering method is sensitive to the initial centroids and is easy to be trapped into local optima. But k-means is still the most popular clustering algorithm because of its effectiveness and efficiency. Some variable clustering algorithms are implemented by k-means. In this section, we introduce k-means-based variable clustering

algorithms in this section at first. Then MCLPSO is used to optimize the initial centroids for the k-means based variable clustering algorithm.

Some notations used in variable clustering are defined as follows:

(i) $X = (X_1, X_2, \ldots, X_N)$ is a N-dimension multivariate random variable, in which X_i is a continuous random variable

(ii) $x_i = (x_{i1}, x_{i2}, \ldots, x_{iN})$ is an observation of X

(iii) $S = \{x_i\}_{i=1}^{M}$ is a dataset composed of M observations of X

We consider hard partitioning clustering in this work, so each variable belongs to only one cluster. Based on the above notations, we can give a formal description for variable clustering furthermore:

Definition 1. Clustering of variables can be defined as K partition on the variable set, Partition_K:

$$\begin{aligned} \text{Partition}_K &= \left(\text{Cluster}_1, \text{Cluster}_2, \ldots, \text{Cluster}_K\right), \\ &\forall k, \text{Cluster}_k \subseteq X,\ 1 \le k \le K. \end{aligned} \tag{8}$$

Partition_K should satisfy the following constraints:

$$\begin{aligned} &\forall k, \text{Cluster}_k \neq \varnothing, \quad 1 \le k \le K, \\ &\text{Cluster}_i \cap \text{Cluster}_j = \varnothing, \quad 1 \le i,\ j \le K,\ i \neq j, \\ &\bigcup_{k=1}^{K} \text{Cluster}_k = X. \end{aligned} \tag{9}$$

In Sections 5 and 6, we choose some datasets generated by the sensors of complex manufacturing system, so the variables discussed in this section are quantitative. In [20], some variable clustering methods are developed to cluster variables with mixed types.

4.1. Principal Component Analysis—PCA. Centroid update rule is critical to k-means-based variables clustering. In almost all the variable clustering algorithms in the literature, PCA is used to compute the first principal component as the centroid for a group of variables in a cluster. In this section, we give a brief introduction to PCA at first.

PCA is a widely used dimension reduction method. The essentiality of PCA is the coordinate transformation. The projection of data on the new coordinate can maximize the variance. PCA transforms x_i to x_i' by projecting x_i on new coordinate U' in (13), the dimension of x_i' is less than N:

$$x_i' = x_i U', \tag{10}$$

U' is a submatrix of U and U' is obtained by deleting some columns from U. U is a $N \times N$ orthogonal matrix, U_j is the jth column of U, and U_j is defined as the jth eigenvector of the sample covariance matrix C. C is the sample covariance matrix of dataset S defined by (11), $C = (c_{ij})_{N \times N}$:

$$c_{ij} = \frac{1}{M-1} \sum_{k=1}^{M} \left(x_{ki} - \mu_{Xi}\right)\left(x_{kj} - \mu_{Xj}\right). \tag{11}$$

From (12), we can get that C is a real symmetric matrix. By the properties of real symmetric matrix, we can get that there are N real eigenvalues of C ($\lambda_1, \lambda_2, \ldots, \lambda_N$) and it is possible that $\lambda_i = \lambda_j$, $1 \le i \neq j \le N$. The eigenvectors of C ($U_1, U_2, \ldots, U_N$) corresponding to ($\lambda_1, \lambda_2, \ldots, \lambda_N$) are real vectors. Eigenvectors corresponding to different eigenvalues are orthogonal to each other.

$$\lambda_j U_j = C U_j,\ j = 1, 2, \ldots, N, \lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_N, \tag{12}$$

where λ_j is the eigenvalue of C, U_j is the eigenvector corresponding to λ_j. Let $\lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_N$, then $U_1, U_2, \ldots, U_N$ is sorted by their corresponding eigenvalues. The projection of S on U_1 direction has the largest variance. The projection of S on U_2 has the second largest variance, and so on. These eigenvectors are all orthogonal to each other. We can choose T eigenvectors with T maximum eigenvalues $U' = (U_1, U_2, \ldots, U_T)$. The S' is the reduction dataset and S' is the projection of S on U'.

$$S' = SU'. \tag{13}$$

Based on the discussion above, we can summarize the steps to calculate the FPC (First Principal Component) for S:

(1) Calculate the sample covariance matrix C for S

(2) Calculate the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$ for C by Jacobi method

(3) Choose the largest eigenvalue value λ_1 and U_1 is the eigenvector corresponding to λ_1

(4) Compute the projection of S on U_1 and get the $\text{FPC} = SU_1$

The Pseudocode of FPC is described in Figure 6.

4.2. Variable Clustering Based on KMEANSVAR. We use MCLPSO to optimize the k-means-based variable clustering algorithm KMEANSVAR which is same as CLV_kmeans in R packge ClustVarLV [19]. In KMEANSVAR, the variables are clustered iteratively and the key components of KMEANSVAR are defined as follows:

(1) *Similarity*. In variable clustering, the similarity between variables is usually defined by correlation coefficient. In KMEANSVAR, Pearson correlation (14) is used to measure the similarity between the variables. If the two variables are highly correlated, the variables will be closer to each other, and vice versa. The similarity between variables is defined by (15).

$$\text{Pearson}\left(X_i, X_j\right) = \frac{\sum_{k=1}^{M} \left(x_{ki} - \mu_{Xi}\right)\left(x_{kj} - \mu_{Xj}\right)}{\sqrt{\sum_{k=1}^{M} \left(x_{ki} - \mu_{Xi}\right)^2} \sqrt{\sum_{k=1}^{M} \left(x_{kj} - \mu_{Xj}\right)^2}}, \tag{14}$$

$$d\left(X_i, X_j\right) = 1 - \text{Pearson}\left(X_i, X_j\right)^2. \tag{15}$$

function *FPC*(*Dataset: S*)
begin
compute covariance matric C for S
compute eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$ *for C, and* $\lambda_1 \geq \lambda_2 \geq, \ldots, \geq \lambda_N$
get eigenvectors $U_1, U_2, \ldots, U_N$
return the SU_1 *as the first principle component of S*
end

Figure 6: Pseudocode of FPC.

(2) *Update of Centroid*. In KMEANSVAR, the centroid of a cluster of variables is always kept as the FPC of the variables in the cluster.

$$\text{Cluster}_k = \{X_{k1}, X_{k2}, \ldots, X_{kP}\}, \tag{16}$$

$S\text{Cluster}_k$ is the samples composed by M observations of the random vector (X_{k1}, X_{k2}, ..., X_{kP}). $S\text{Cluster}_k$ can be obtained by keeping X_{k1}, X_{k2}, ..., X_{kP} and deleting X-{X_{k1}, X_{k2}, ..., X_{kP} } from S. FPC ($S\text{Cluster}_k$) is the centroid of Cluster$k$.

(3) *Clustering Criterion*. The quality of clustering result is measured by clustering criterion. A high-quality clustering of variables can maximize the clustering criterion. In [20], a clustering criterion is proposed for both quantitative and qualitative variables. In this work, we only take quantitative variables into consideration. In KMEANSVAR, the clustering criterion is defined by the homogeneity of variables in each cluster. H (Cluster_k) denotes the homogeneity of the variable cluster Cluster_k which is defined by (17). In (17), centroid_k is the cenroid of Cluster_k obtained by FPC ($S\text{Cluster}_k$).

$$H(\text{Cluster}_k) = \sum_{X_{ki} \in \text{Cluster}_k} \text{Pearson}^2(X_{ki}, \text{centroid}_k). \tag{17}$$

H (Partition_K) denotes the homogeneity of a clustering of variables Partition_K, which is defined by

$$H(\text{Partition}_k) = \sum_{\text{Cluster}_k \in \text{Partition}_K} H(\text{Cluster}_k). \tag{18}$$

Based on the discussion above, we can give the steps of KMEANSVAR in detail.

(1) Initialize K cluster centroids for centroid_1, ..., centroid_K for clusters Cluster_1, ..., Cluster_K

(2) Clear the clusters Cluster_1, ..., Cluster_K

(3) For each $X_i \in X$, find its nearest cluster $\text{Cluster}_{\text{nearest}}$, and assign X_i to $\text{Cluster}_{\text{nearest}}$

$$\text{Cluster}_{\text{nearest}} = \arg\min_{\text{Cluster}_k \in \text{Partition}_K} (d(X_i, \text{Cluster}_k)),$$
$$\text{Cluster}_{\text{nearest}} = \text{Cluster}_{\text{nearest}} \cup X_i. \tag{19}$$

The distance between a variable and a cluster is defined by the distance between the variable and the centroid of the cluster as

$$d(X_i, \text{Cluster}_k) = d(X_i, \text{centroid}_k). \tag{20}$$

(4) Computer $\text{centroid}_k = \text{FPC}$ ($S\text{Cluster}_k$) as the new centroid for Cluster_k

(5) Iteratively do (2) to (4) until the maximum iterations is reached

The Pseudocode of KMEANSVAR is described in Figure 7.

4.3. Variable Clustering Based on MCLPSO. Although KMEANSVAR can cluster the variables efficiently, KMEANSVAR is as sensitive to initial centroids as some other k-means-based methods. The clustering criterion is easy to be trapped to the local optima and the quality of clustering cannot be guaranteed. To overcome this shortcoming, MCLPSO is used to optimize the initial centroids for KMEANSVAR and MCLPSO-KMEANSVAR is proposed. In MCLPSO-KMEANSVAR, the solution is coded as the initial centroids for kmeansvar, and KMEANSVAR is embeded into the objective function of MCLPSO. MCLPSO optimizes the following:

(1) Coding of the solution: particle i's solution is coded as a D-dimension vector (21), $D = K * M$, K is the number of clusters, M the number of observations. The kth component of solution_i-centroid_{ik} denotes that the centroid of the kth cluster centroid_k is initialized by centroid_{ik}, and solution_i determines the initial centroids for the clusters. The pos_i and $pbest_i$ of particle i can be denoted as solution_i:

$$\text{solution}_i = (\text{centroid}_{i1}, \text{centroid}_{i2}, \ldots, \text{centroid}_{iK}). \tag{21}$$

(2) Objective function: in order to improve the quality of the clustering of variables, MCLPSO-KMEANSVAR optimizes H (Partition_K) by optimizing the initial centroids for KMEANSVAR. solution_i can be decomposed to K initial centroids: centroid_{i1}, ..., centroid_{iK}. The clustering of variables can be obtained by call KMEANSVAR parameterized by centroid_{i1}, ..., centroid_{iK}, i.e., $\text{Partition}_K = \text{KMEANSVAR}$ (centroid_{i1}, ..., centroid_{iK}), and the clustering criterion H (Partition_K) can be obtained by (20). $1/H$ (Partition_K) is defined as the value of the objective function f. KMEANSVAR is embeded into the objective function of MCLPSO; therefore, the clustering result of KMEANSVAR can be optimized by adjusting the initial centroids for KMEANSVAR (Figure 8).

5. Experiment

As the clustering criterion of the clustering results has been defined in Section 4, we give some experiment results in this section. We evaluate the performance of the proposed algorithm MCLPSO-KMEANSVAR and compare it with some other variable clustering methods. MCLPSO-

```
function kmeansvar(K initial centroids: centroid_1, ..., centroid_K)
 do
    for k = 1 to K
        Cluster_k = Ø
    endfor
    for i = 1 to N
        Cluster_nearest = argmin_{Cluster_k ∈ Partition_K} (d(X_i, Cluster_k))
        Cluster_nearest = {X_i} ∪ Cluster_nearest
    endfor
    for k = 1 to K
        recalculate the centroid_k by FPC (S_Clusterk)
    endfor
 while(the stop criterion is not met)
 return Partition_K = {Cluster_1, ..., Cluster_K} as result
end
```

Figure 7: Pseudocode of KMEANSVAR.

```
function f (solution: sol)
begin
decompose sol and get centroid_1, ..., centroid_K
Partition_K = KMEANSVAR(centroid_1, ..., centroid_K)
return the result of 1/H(Partition_K) as the value of f
end
```

Figure 8: Objective function of MCLPSO-KMEANSVAR.

KMEANSVAR is compared with CLPSO-KMEANSVAR (KMEANSVAR initialized by CLPSO) and the original version of KMEANSVAR with random initialization. In [18–20], cutting the dendrogram is recommended to initialize the k-means-based variable clustering method, but it is also stated that the hierarchical variable clustering lacks scalability when the number of candidate variables increases because its O (N^2) complexity in which N is the number of candidate variables. We choose a more scalable initialization k-means++ initialization [27] in which the first cluster center is chosen uniformly at random from the data points that are being clustered, after which each subsequent cluster center is chosen from the remaining data points with probability proportional to its squared distance from the point's closest existing cluster center. It is easy to apply k-means++ initialization to KMEANSVAR and obtain KMEANSVAR++. The variable clustering methods and some experimental setting are listed in Table 1.

5.1. Datasets. We choose several real-world datasets as the benchmark datasets to test the variable clustering method in Table 1. The detailed information of the datasets is listed in Table 2. D_1 are chosen from UCI datasets. D_2 and D_3 are collected from the MES (Manufacturing Execution System) database of a large-scale semiconductor manufacturing system located at Shanghai. D_2 is composed of the values of the manufacturing performance variables and D_3 is composed of the values of manufacturing status variables. D_4 is the SECOM dataset which is described in [28]. A complex modern semiconductor manufacturing process of house line testing is normally under consistent surveillance via the monitoring of signals/variables collected from sensors and or process measurement points. SECOM is collected from the database of the FCS (Floor Control System) of the semiconductor manufacturing process. In D_1-D_4, only continuous variables are considered. The number of clusters for each dataset is set according to the number of variables.

In order to ensure the validity of the evaluation, D_1–D_4 are preprocessed before experiment. First, all the values are normalized between [0, 1] by Min-Max Normalization method. Especially, D_4 contains some null values because the FCS sometimes influenced by sensor drifting results in data loss. Therefore, we give following rules to clean D_4. After cleaning, a complete dataset D_4 consists of 1560 instances, and each instance has 440 variables. D_4 is a difficult variable clustering problem.

(i) Remove the variables with unchangeable data

(ii) Remove the variables with more than 50% missing data

(iii) Remove the data items with more than 30% missing data

5.2. Parameter Setting. There are many parameters in the MCLPSO. According to the "No Free Lunch" theorem [29], there do not exist a so-called optimal parameterization. We set the parameters by following some empirical rules mentioned in some studies [30, 31].

For the parameters in the global search component of MCLPSO, w decreases from 0.9 to 0.4 linearly, $c = 1.49445$, $m = 7$, the number of generations is set at 100, the population size is set at 20. For the parameters in the SA_local_search, $T_0 = 10$ to give a fine-grained local search. For the parameters in the "cognition-only" model, w decreases from 0.9 to 0.4 linearly along with the evolution cycles and $c_1 = 1.49445$. P_{SA} and P_{Chaotic} are both set to 0.1. For the other parameters, we found two heuristic rules by some tentative experiments. *Chaotic_ls_length* should be positively correlated with $\text{stagnant}_{\max}$ as $\text{stagnant}_{\max}$ determines the degree of stagnation of the particle. A high value of $\text{stagnant}_{\max}$ implies that a high value of *Chaotic_ls_length* is needed to enable the stagnant particle to escape from the local optimum. *SA_ls_length* is negatively correlated with $\text{improve}_{\max}$ because a high value of $\text{improve}_{\max}$ denotes a high quality of a promising particle. A moderate value of *SA_ls_length* is enough to detect the local optima. These parameters are set empirically: $\text{stagnant}_{\max} = 10$, $\text{improve}_{\max} = 3$, *Chaotic_ls_length* = 100, and *SA_ls_length* = 100.

In CLPSO, w decreases from 0.9 to 0.4 linearly and c is set at 1.49445 as recommended by [5].

In KMEANSVAR, the number of clusters for each dataset has been specified in Table 2. If KMEANSVAR is evaluated in the fitness function, the maximum number of iterations is set to 10.

5.3. Result and Discussion. The mean value and the standard deviation are recorded in Table 3 with the best result in bold.

First, we assess the effect of introducing the meta-heuristic optimization on variable clustering. From Table 3, we can find that the mean values of clustering criterion

Table 1: Variable clustering methods for comparison.

Clustering heuristic	Initialization strategy	Stop criterion	Number of test
MCLPSO-KMEANSVAR	Initialize K centroids by MCLPSO in which each particle is initialized by uniform distribution bounded by minimum and maximum at each dimension	Maximum 2000 evaluation of KMEANSVAR	100
CLPSO-KMEANSVAR	Initialize K centroids by CLPSO in which each particle is initialized by uniform distribution bounded by minimum and maximum at each dimension	Maximum 2000 evaluation of KMEANSVAR	100
KMEANSVAR	Choose K variables randomly from the candidate variables as initial centroids	No change of the clusters	100
KMEANSVAR++	Choose K variables subsequently from the candidate variables with probability proportional to its squared distance from the point's closest existing cluster center	No change of the clusters	100

Table 2: Dataset to evaluate MCLPSO-KMEANSVAR

Dataset	Sample number	Variable number	Cluster number	Dataset description
D_1	8192	22	2, 3, 4	A dataset of computer systems activity measures. The data were collected from a Sun Sparcstation 20/712 with 128 MB of memory running in a multiuser university department. This dataset is downloaded from http://tunedit.org/repo/UCI/numeric, cpu_act.arff.
D_2	3300	11	2, 3, 4	A dataset of performance of semiconductor manufacturing system. The data were collected from the MES system database of a large semiconductor manufacturing factory which located at Shanghai. Each variable records the values of manufacturing performance such as average machine utility of the manufacturing system.
D_3	550	67	5, 10, 20	A dataset of manufacturing environment of semiconductor manufacturing system. The data collected from the MES system database of a large semiconductor manufacturing factory which located at Shanghai. Each variable records the values of manufacturing environment such as number of wafers or lots waiting for a lithography machine.
D_4	1560	440	5, 10, 20	A dataset of monitoring data of semiconductor manufacturing system. The data were collected from monitoring of signals/variables collected from sensors and or process measurement points. This dataset is downloaded from https://archive.ics.uci.edu/ml/datasets/secom.

obtained by KMEANSVAR on D_1–D_4 are relatively poor because of the intrinsic deficiency of k-means clustering that is sensitive to initial centroids. KMEANSVAR is easy to be trapped into local optima and results in a relatively poor variable clustering. KMEANSVAR also shows a large standard deviation, so the performance of KMEANSVAR is not stable. On the simplest dataset D_2 with only 11 variables, the clustering criterion values obtained by KMEANSVAR are not satisfactory enough and the variance values remain large. KMEANSVAR++ can improve KMEANSVAR by choosing centroids with probability proportional to its squared distance from the point's closest existing cluster center. The improvement is definite but not so significant. CLPSO-KMEANSVAR can improve the clustering results significantly compared with KMEANSVAR. The mean values of the clustering criterion obtained by CLPSO-KMEANSVAR are improved, and the variance values of the clustering criterion are also reduced. Therefore, the clustering result can be improved more significantly by introducing the metaheuristic optimization than using k-means++ seeding.

Second, we analyze the effect of introducing the local search operators and adaptive memetic strategy to the population-based metaheuristic optimization on variable

Table 3: The result of the clustering of variables.

Methods	Dataset											
	D_1 ($K=2$)	D_1 ($K=3$)	D_1 ($K=4$)	D_2 ($K=2$)	D_2 ($K=3$)	D_2 ($K=4$)	D_3 ($K=5$)	D_3 ($K=10$)	D_3 ($K=20$)	D_4 ($K=5$)	D_4 ($K=10$)	D_4 ($K=20$)
MCLPSO-KMEANSVAR	**1.14E + 01 ± 1.46E − 01**	**1.21E + 01 ± 1.46E − 01**	**1.24E + 01 ± 1.46E − 01**	**6.69E + 00 ± 0.00E + 00**	**4.66E + 00 ± 1.64E − 02**	**4.46E + 00 ± 2.11E − 02**	**3.44E + 01 ± 2.12E − 01**	**4.48E + 01 ± 4.06E − 01**	**5.18E + 01 ± 5.29E − 01**	**4.36E + 01 ± 4.01E − 01**	**6.51E + 01 ± 4.22E + 00**	**1.06E + 02 ± 3.04E + 00**
CLPSO-KMEANSVAR	1.13E + 01 ± 1.49E − 01	1.19E + 01 ± 1.54E − 01	1.23E + 01 ± 1.60E − 01	**6.69E+00 ± 0.00E + 00**	**4.66E + 00 ± 1.64E − 02**	4.45E + 00 ± 2.09E − 02	**3.44E + 01 ± 2.10E − 01**	4.44E + 01 ± 3.92E − 01	5.13E + 01 ± 5.12E − 01	4.34E + 01 ± 6.01E − 01	6.41E + 01 ± 3.42E + 00	1.01E + 02 ± 3.04E + 00
KMEANSVAR	4.33E + 00 ± 1.14E + 0	9.24E + 00 ± 1.32E + 0	9.33E + 00 ± 1.14E + 0	5.44E + 00 ± 5.21E − 01	3.49E + 00 ± 6.43E − 01	3.93E + 00 ± 4.41E − 01	3.41E + 01 ± 4.40E − 01	4.32E + 01 ± 1.43E + 00	4.44E + 01 ± 1.99E + 00	3.99E + 01 ± 2.64E + 00	6.00E + 01 ± 4.64E + 00	4.94E + 01 ± 3.44E + 00
KMEANSVAR++	4.43E + 00 ± 0.97E − 01	9.97E + 00 ± 1.14E + 0	1.13E + 01 ± 8.16E − 01	6.13E + 00 ± 1.17E − 01	4.29E + 00 ± 5.71E − 01	4.22E + 00 ± 1.43E − 01	3.42E + 01 ± 4.20E − 01	4.39E + 01 ± 1.17E + 00	4.89E + 01 ± 1.01E + 00	4.13E + 01 ± 2.78E + 00	6.27E + 01 ± 4.36E + 00	6.43E + 01 ± 3.37E + 00
H	0	0	1	0	0	1	0	1	1	1	1	1

clustering. From Table 3, we can find that on D_1-D_2, the mean values of clustering criterion obtained by MCLPSO-KMEANSVAR are similar with the mean values of clustering criterion obtained by CLPSO-KMEANSVAR. The number of possible clustering results can be derived by the number of variables and the number of clusters. For example, the possible number of clustering results on D_2 is $C2$ 11, $C3$ 11, and $C4$ 11 when the number of clusters is 2, 3, and 4. Therefore, the difference between MCLPSO-KMEANSVAR's results and CLPSO-KMEANSVAR's results on D_1-D_2 is not significant because of the limited number of clusters and variables. When the number of variables and clusters increases, the advantage of MCLPSO-KMEANSVAR is more significant. On D_3, MCLPSO-KMEANSVAR has a better performance than CLPSO-KMEANSVAR and the improvement will be more significant when the number of clusters increases. The advantage of MCLPSO-KMEANSVAR is more significant on D_4—a complex real-world industry dataset with 440 variables. Therefore, the global search operators and the local search operators will take effect when dealing with dataset with large number of variables.

Furthermore, we analyze the robustness of MCLPSO-KMEANSVAR when dealing with the complex real-word dataset. MCLPSO-KMEANSVAR can generally improve the quality of the clustering of variables by optimizing the clustering criterion; its variance values on D_1–D_4 are not reduced. To show the robustness of the above approaches, the boxplots of MCLPSO-KMEANSVAR, CLPSO-KMEANSVAR, and KMEANSVAR's result on D_4 are depicted in Figures 9–11. From Figure 9, we can find that KMEANSVAR lacks robustness because of its intrinsic deficiency. CLPSO-KMEANSVAR's results' distributions are flatter. The variable clustering results can be improved significantly by introducing metaheuristic optimization. Compared with CLPSO-KMEANSVAR's results, MCLPSO-KMEANSVAR's results' values of range and interquartile range are relatively higher, so the robustness cannot be improved by introducing the local search operators and adaptive memetic strategy. But in Figure 11, we find that MCLPSO-KMEANSVAR can avoid some extreme bad cases. In Figure 10, we find that the possibility to find satisfactory results is also higher.

To prove the improvement brought by MCLPSO-KMEANSVAR compared with CLPSO-KMEANSVAR is definite, nonparametric Wilcoxon rank sum tests are conducted between the MCLPSO's results and the CLPSO's results. The results of tests are presented in the last row of Table 3. If $h = 1$, the performances of the two algorithms are statistically different with 95% certainty. If $h = 0$, the performances are not statistically different. From Table 3, we find that MCLPSO-KMEANSVAR and CLPSO-KMEANSVAR are statistically different with the increase of K and the number of candidate variables.

5.4. Implementation and Computational Time. The algorithms discussed above are all implemented in JAVA 8, so we can use the multithread technique to accelerate the particle's comprehensive learning process. We run the codes on Intel i5-8365U CPU with a parallelism of 8, i.e., 8 particles can do comprehensive learning operation simultaneously.

When we use MCLPSO to optimize KMEANSVAR, we will run the KMEANSVAR with a maximum number of evaluations (calling KMEANSVAR) 2000 as stated in Table 1. In 5.2, we stated that when the KMEANSVAR is called in a fitness function, the number of iterations is restricted to 10. The computational time of MCLPSO-KMEANSVAR is

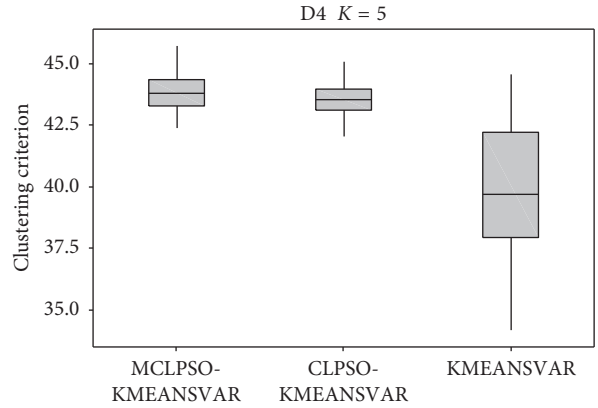

Figure 9: Boxplot of results on D_4 ($K = 5$).

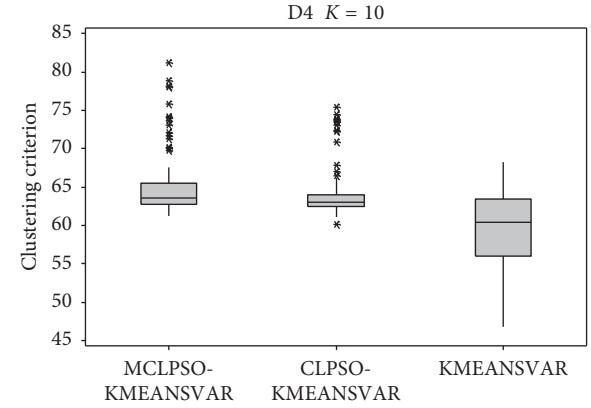

Figure 10: Boxplot of results on D_4 ($K = 10$).

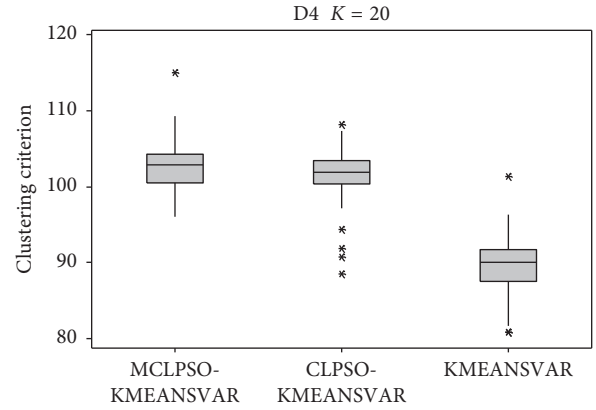

Figure 11: Boxplot of results on D_4 ($K = 20$).

Table 4: The result of computational time on D4 (minutes).

K methods	$K=5$	$K=10$	$K=20$
MCLPSO-KMEANSVAR	11.3	28.1	54.3
KMEANSVAR++	1.3	7.9	24.5
KMEANSVAR	0.3	0.7	1.7

about 40–60 times as much as the computational time of KMEANSVAR.

The computational time of D_4 with different K is listed in Table 4. Then computation time of MCLPSO-KMEANSVAR and KMEANSVAR shows a linear increase with respect to the increase of K, but KMEANSVAR++ increases dramatically because the initialization of KMEANSVAR++ is sensitive to K. So MCLPSO-KMEANSVAR is more scalable than KMEANSVAR++ (Table 5).

6. A Web-Based Interactive Software Platform

In Section 5, some datasets are used to evaluate MCLPSO-KMEANSVAR. Except D_1, D_2-D_4 are collected from the information system databases of semiconductor manufacturing factories. The relationship between D_2-D_4 is explained in Figure 12. The variable clustering analysis of the variables of D_2-D_4 is an important and practical work. It is helpful to find useful insights of manufacturing systems from different perspectives and improve the operation management by some further analysis such as performance analysis, optimal control, fault diagnosis.

For the purpose of practical usage, we have also developed a web-based interactive software platform based on MCPSO-KMEANSVAR. In this section, we introduce the usage of a software platform by demonstrating each step. The performance analysis of the semiconductor manufacturing system is introduced as a case study.

6.1. Performance of Semiconductor Manufacutring System. The semiconductor manufacturing system is a very complicated system and its performance can be affected by the manufacturing environment, scheduling rules, equipment failure rate, and rush order. The analysis of performance is useful to improve the operation management of semiconductor manufacturing system. In Table 3, we choose 8 performance variables, in which Y_1–Y_3 are long-term global performance, Y_4–Y_6 are short-term global performance, and Y_7–Y_8 are short-term local performance. The detailed description is presented in Table 3.

6.2. A Web-Based Interactive Variable Clustering System. First, the dataset of the performance history data should be uploaded (Figure 13).

Table 5: The performance of semiconductor manufacutring system.

Variable	Performance	Description
Y_1	MCT	Mean of wafers' cycle time
Y_2	$\text{MCT}_{\text{STDEV}}$	Standard deviation of wafers' cycle time
Y_3	Productivity	Ratio of output of wafers
Y_4	Mov	Total movement of wafers
Y_5	Turn	Average movement of wafers
Y_6	Utility	Average machine utility of all the machines
Y_7	$\text{Utility}_{\text{DF}}$	Average machine utility of the machines located in the diffusion area
Y_8	$\text{Utility}_{\text{LT}}$	Average machine utility of the machines located in the lithography area

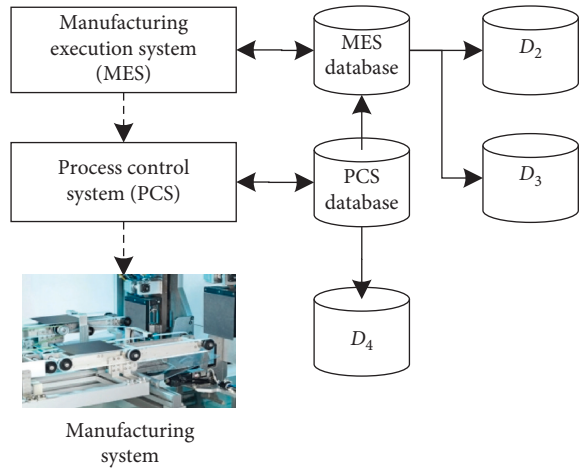

Figure 12: Relationship between D_2, D_3, and D_4.

Figure 13: Upload dataset.

Then some statistics of each variable can be found in Figure 14. The user can also choose some thresholds to smooth the outliers for each variable before variable clustering.

The result of MCLPSO-KMEANSVAR is presented in Figure 15. We can reduce the number of optimization objectives by variable clustering.

7. Conclusion

In this work, MCLPSO, a novel memetic algorithm is presented in our previous research, is introduced as a metaheuristic approach to improve k-means-based variable clustering. The experiment results show that MCLPSO-KMEANSVAR out-

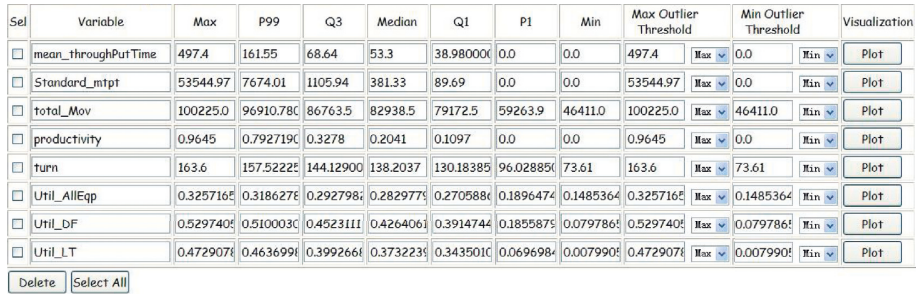

Figure 14: Statistics of variables.

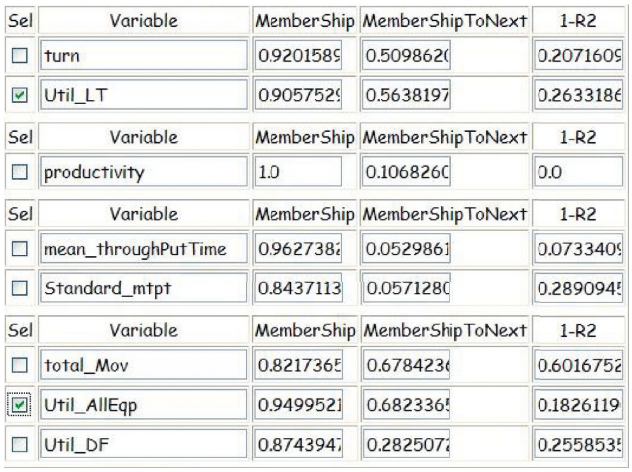

Figure 15: Result of variable clustering by MCLPSO-KMEANSVAR.

performs KMEANSVAR significantly. We also develop a web-based interactive software platform to implement MCLPSO-KMEANSVAR and give a case study of performance analysis for semiconductor manufacturing system. In the future research, we will further study the practical use of MCLPSO-KMEANSVAR in other problems and develop a distributed MCLPSO-KMEANSVAR for analyze the big data.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

In this work, the design, implementation, experiment, and part of the case study are finished by JiaCheng Ni and Li Li at Tongji University. The paper revisions and part of case study are finished by JiaCheng Ni at DELL EMC.

Acknowledgments

The authors would like to thank Prof. P. N. Suganthan for providing the codes of his research group. The authors would also like to thank Zhen Jia, Qiang Chen, and Jinpeng Liu for discussing the usage of machine learning in Internet of things (Iot) applications. This work was supported by the Key Research and Development Project of National Ministry of Science and Technology under grant no. 2018YFB1305304 and the National Natural Science Foundation of China under grant no. 61873191.

References

[1] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Magazine*, vol. 17, no. 3, pp. 37–54, 1996.

[2] A. Fahad, N. Alshatri, Z. Tari et al., "A survey of clustering algorithms for big data: taxonomy and empirical analysis," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 267–279, 2014.

[3] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. C. P. L. F. de Carvalho, "A survey of evolutionary algorithms for clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 39, no. 2, pp. 133–155, 2009.

[4] J. Ni, L. Li, F. Qiao, and Q. Wu, "A novel memetic algorithm based on the comprehensive learning PSO," in *Proceedings of the 2012 IEEE Congress on Evolutionary Computation*, pp. 1–8, IEEE, Brisbane, Australia, June 2012.

[5] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[6] A. K. Jain, "Data clustering: 50 years beyond *K*-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[7] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," *Pattern Recognition*, vol. 33, no. 9, pp. 1455–1465, 2000.

[8] D. W. Van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," in *Proceedings of the 2003 Congress on Evolutionary Computation, 2003. CEC'03*, vol. 1, pp. 215–220, IEEE, Canberra, Australia, December 2003.

[9] A. A. A. Esmin, D. L. Pereira, and F. P. A. De Araujo, "Study of different approach to clustering data by using the particle swarm optimization algorithm," in *Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 1817–1822, IEEE, Hong Kong, China, June 2008.

[10] A. Ahmadyfard and H. Modares, "Combining PSO and *k*-means to enhance data clustering," in *Proceedings of the 2008 International Symposium on Telecommunications*, pp. 688–691, IEEE, Tehran, Iran, August 2008.

[11] N. Krasnogor, *Studies on the theory and design space of memetic algorithms*, Ph.D. dissertation, University of the West of England, Bristol, U.K, 2002.

[12] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 5, pp. 474–488, 2005.

[13] P. Moscato, "On evolution, search, optimization, GAs and martial arts: toward memetic algorithms," Caltech Concurrent Computation Program Report 826, California Institute of Technology, Pasadena, CA, USA, 1989.

[14] Y.-S. Ong, M. Lim, and X. Chen, "Memetic computation—past, present & future (research frontier)," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 24–31, 2010.

[15] W. Sheng, X. Liu, and M. Fairhurst, "A niching memetic algorithm for simultaneous clustering and feature selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 7, pp. 868–879, 2008.

[16] W. Sheng, S. Chen, M. Fairhurst, G. Xiao, and J. Mao, "Multilocal search and adaptive niching based memetic algorithm with a consensus criterion for data clustering," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 5, pp. 721–741, 2013.

[17] J. Ni, L. Li, F. Qiao, and Q. Wu, "A novel memetic algorithm and its application to data clustering," *Memetic Computing*, vol. 5, no. 1, pp. 65–78, 2013.

[18] E. Vigneau and E. M. Qannari, "Clustering of variables around latent components," *Communications in Statistics—Simulation and Computation*, vol. 32, no. 4, pp. 1131–1150, 2003.

[19] E. Vigneau, M. Chen, and E. M. Qannari, "ClustVarLV: an R package for the clustering of variables around latent variables," *The R Journal*, vol. 7, no. 2, p. 134, 2015.

[20] M. Chavent, V. K. Simonet, B. Liquet, and J. Saracco, "ClustOfVar: an R package for the clustering of variables," *Journal of Statistical Software*, vol. 50, no. 13, pp. 1–16, 2012.

[21] W. Hart, *Adaptive global optimization with local search*, Ph.D dissertations, University of California, San Diego, CA, USA, 1994.

[22] Q. H. Nguyen, Y.-S. Ong, and M. H. Lim, "A probabilistic memetic framework," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 604–623, 2009.

[23] G. Iacca, F. Neri, E. Mininno, Y.-S. Ong, and M.-H. Lim, "Ockham's Razor in memetic computing: three stage optimal memetic exploration," *Information Sciences*, vol. 188, pp. 17–43, 2012.

[24] I. Fister and J. B. Žumer, "Memetic artificial bee colony algorithm for large-scale global optimization," in *Proceedings of the 2012 IEEE Congress on Evolutionary Computation*, pp. 1–8, IEEE, Brisbane, Australia, June 2012.

[25] B. Liu, L. Wang, Y.-H. Jin, F. Tang, and D.-X. Huang, "Improved particle swarm optimization combined with chaos," *Chaos, Solitons & Fractals*, vol. 25, no. 5, pp. 1261–1271, 2005.

[26] J. Kennedy, "The particle swarm: social adaptation of knowledge," in *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (CEC'97)*, pp. 303–308, IEEE, Indianapolis, IN, USA, April 1997.

[27] D. Arthur and S. Vassilvitskii, "*k*-means++: the advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, January 2007.

[28] M. McCann, Y. Li, L. Maguire, and A. Johnston, "Causality challenge: benchmarking relevant signal components for effective monitoring and process control," in *Proceedings of the Causality: Objectives and Assessment*, pp. 277–288, Whistler, Canada, February 2010.

[29] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[30] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360)*, pp. 69–73, IEEE, Anchorage, AK, USA, May 1998.

[31] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.