A Novel Clustering Technique Based on Improved Noising Method

Yongguo Liu[1,2,4], Wei Zhang[3], Dong Zheng[4], and Kefei Chen[4]

[1] College of Computer Science and Engineering,
University of Electronic Science and Technology of China,
Chengdu 610054, P. R. China
[2] State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210093, P. R. China
[3] Department of Computer and Modern Education Technology,
Chongqing Education College, Chongqing 400067, P. R. China
[4] Department of Computer Science and Engineering,
Shanghai Jiaotong University, Shanghai 200030, P. R. China

Abstract. In this article, the clustering problem under the criterion of minimum sum of squares clustering is considered. It is known that this problem is a nonconvex program which possesses many locally optimal values, resulting that its solution often falls into these traps. To explore the proper result, a novel clustering technique based on improved noising method called INMC is developed, in which one-step DHB algorithm as the local improvement operation is integrated into the algorithm framework to fine-tune the clustering solution obtained in the process of iterations. Moreover, a new method for creating the neighboring solution of the noising method called mergence and partition operation is designed and analyzed in detail. Compared with two noising method based clustering algorithms recently reported, the proposed algorithm greatly improves the performance without the increase of the time complexity, which is extensively demonstrated for experimental data sets.

1 Introduction

The clustering problem is a fundamental problem that frequently arises in a great variety of application fields such as pattern recognition, machine learning, and statistics. In this article, we focus on the minimum sum of squares clustering problem stated as follows: Given N objects in R^m, allocate each object to one of K clusters such that the sum of squared Euclidean distances between each object and the center of its belonging cluster for every such allocated object is minimized. This problem can be mathematically described as follows:

$$\min_{W,C} J(W,C) = \sum_{i=1}^{N}\sum_{j=1}^{K} w_{ij}\|\mathbf{x}_i - \mathbf{c}_j\|^2 \qquad (1)$$

where $\sum_{j=1}^{K} w_{ij} = 1$, $i = 1, \ldots, N$. If object $\mathbf{x}_i$ is allocated to cluster C_j, then w_{ij} is equal to 1; otherwise w_{ij} is equal to 0. Here, N denotes the number of objects,

M. Lazo and A. Sanfeliu (Eds.): CIARP 2005, LNCS 3773, pp. 81–92, 2005.

m denotes the number of object attributes, K denotes the number of clusters, $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ denotes the set of N objects, $C = \{C_1, \ldots, C_K\}$ denotes the set of K clusters, and $W = [w_{ij}]$ denotes the $N \times K$ $0-1$ matrix. Cluster center $\mathbf{c}_j$ is calculated as follows:

$$\mathbf{c}_j = \frac{1}{n_j} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i \tag{2}$$

where n_j denotes the number of objects belonging to cluster C_j. This clustering problem is a nonconvex program which possesses many locally optimal values, resulting that its solution often falls into these traps. It is known that this problem is NP-hard [1]. If exhaustive enumeration is used to solve this problem, then one requires to evaluate

$$\frac{1}{K!} \sum_{j=1}^{K} (-1)^{K-j} \binom{K}{j} j^N \tag{3}$$

partitions. It is seen that exhaustive enumeration cannot lead to the required solution for most problems in reasonable computation time [2].

Many methods have been reported to deal with this problem [2,3]. Among them, K-means algorithm is a very popular one but it converges to local minima in many cases [4]. Moreover, many researchers attempt to solve this problem by stochastic optimization methods including evolutionary computation [5,6,7], tabu search [8], and simulated annealing [9]. By adopting these techniques, researchers obtain better performance than by using local iteration methods such as K-means algorithm. In [10], the noising method, a recent metaheuristic technique firstly reported in [11], is introduced to deal with the clustering problem under consideration. In the field of metaheuristic algorithms, to efficiently use them in various kinds of applications, researchers often combine them with local descent approaches [12,13]. To efficiently use the noising method in the clustering problem, in [10], the authors introduced K-means algorithm as the local improvement operation to improve the performance of the clustering algorithm. As a result, two methods called NMC and KNMC, respectively, are developed. NMC does not own K-means operation but KNMC does. The choice of the algorithm parameters is extensively discussed, and performance comparisons between these two methods and K-means algorithm, GAC [5], TSC [8], and SAC [9] are conducted on experimental data sets. It is concluded that, with much less computational cost than GAC, TSC, and SAC, KNMC can get much better clustering results sooner than NMC, GAC, and TSC, and obtain results close to those of SAC. Meanwhile, it is found that the results of KNMC are still inferior to those of SAC in most cases.

The motivation of this article is how to design a new noising method based clustering algorithm. On one hand, the low complexity should be kept, and on the other hand, the quality of outputs should be further improved. Here, we find there are still some problems in KNMC. Firstly, methods better than K-means algorithm are not considered, and secondly, the probability threshold employed in [10] need be determined in advance. But it is very difficult for the designer to choose the proper value in different cases. In this paper, two novel operations are introduced, DHB operation and mergence and partition operation. The role of DHB operation is similar to that of K-means operation in [10], but the former can further improve the current solution. Mergence and partition operation similar to the probability threshold is used to establish the neighboring solution, but it does not need any parameter and can attain much better results than the latter. By introducing these two modules, we develop a new clustering technique based on improved noising method called INMC. By extensive computer simulations, its superiority over NMC, KNMC, and even SAC is demonstrated.

The remaining part of this paper is organized as follows: In Section 2, INMC algorithm and its components are described in detail. In Section 3, how to determine proper modules is extensively discussed. Performance comparisons between the proposed algorithm and other techniques are conducted on experimental data sets. Finally, some conclusions are drawn in Section 4.

2 INMC Algorithm

As stated in [10,14], instead of taking the genuine data into account directly, the noising method considers the optimal result as the outcome of a series of fluctuating data converging towards the genuine ones. Figure 1 gives the general description of INMC. The architecture of INMC is similar to that of KMNC and their most procedures observe the main architecture of the noising method. The difference between KNMC and INMC lies that two new operations are introduced in INMC. The detail discussion about KNMC and the noising method can be found in [10] and [14], respectively. Here, DHB operation consisting of one-step DHB algorithm is used to fine-tune solution X_c and accelerate the convergence speed of the clustering algorithm. Moreover, mergence and partition operation is designed to establish the neighboring solution.

```
Begin
  set parameters and the current solution $X_c$ at random
  while $N_i \leq N_t$ do
    $N_i \leftarrow N_i + 1$
    perform DHB operation to fine-tune solution $X_c$
    perform mergence and partition operation to create the neighbor $X'$
    if $f(X') - f(X_c) + noise < 0$, then $X_c \leftarrow X'$
    if $f(X_c) < f(X_b)$, then update the best solution $X_b \leftarrow X_c$
    if $N_i = 0(\mathrm{mod} N_f)$, then decrease the noise rate $r_n$
  end do
  output solution $X_b$
end
```

Fig. 1. General description of INMC

2.1 DHB Operation

In [15], an iterative method called DHB algorithm, a breadth-first search technique, for the clustering problem is reported. According to this algorithm, another alternative approach called DHF algorithm, a depth-first search technique, is described in [16]. In [17], two algorithms called AFB algorithm and ABF algorithm, respectively, based on hybrid alternating searching strategies, are presented to overcome the drawbacks of either a breadth-first search or a depth-first search in the clustering problem. In [18], five iteration methods (DHB, DHF, ABF, AFB, and K-means) are compared. First four methods have the similar performance and own stronger convergence states than K-means algorithm. Their time complexities are the same as that of K-means algorithm. In [18], the conclusion is drawn that first four algorithms can get much better clustering results sooner than K-means algorithm and DHB algorithm is recommended to perform the clustering task. The detail descriptions of five methods can be found in the corresponding references. In this paper, we choose DHB algorithm as the local improvement operation to fine-tune solution X_c. Firstly, we define several variables so as to describe DHB operation. For cluster C_j, its objective function value is defined as:

$$J_j = \sum_{\mathbf{x}_i \in C_j} \|\mathbf{x}_i - \mathbf{c}_j\|^2 \tag{4}$$

If object $\mathbf{x}_i$ belonging to cluster C_j is reassigned to C_k, then cluster centers are moved accordingly, J_j decreases by ΔJ_{ij}, J_k increases by ΔJ_{ik}, and the objective function value J is updated as follows:

$$\begin{cases} \Delta J_{ij} = n_j \|\mathbf{x}_i - \mathbf{c}_j\|^2/(n_j - 1) \\ \Delta J_{ik} = n_k \|\mathbf{x}_i - \mathbf{c}_k\|^2/(n_k + 1) \\ J' = J - \Delta J_{ij} + \Delta J_{ik} \end{cases} \tag{5}$$

and C_j and C_k are modulated as follows:

$$\begin{cases} \mathbf{c}'_j = (n_j \mathbf{c}_j - \mathbf{x}_i)/(n_j - 1) \\ \mathbf{c}'_k = (n_k \mathbf{c}_k + \mathbf{x}_i)/(n_k + 1) \end{cases} \tag{6}$$

Then, DHB operation is described as follows: Object $\mathbf{x}_i$ belonging to cluster C_j is reassigned to cluster C_k, iff

$$\min(\Delta J_{ik}) < \Delta J_{ij} \tag{7}$$

where $i = 1, \ldots, N$, $j, k = 1, \ldots, K$, and $j \neq k$. According to Equations 5 and 6, the corresponding parameters are updated. After all objects are considered, the modified solution is obtained.

2.2 Mergence and Partition Operation

In [10], the probability threshold popularly used to create the neighborhood of tabu search is adopted to establish the neighboring solution of the current solution X_c. But the designer has to determine the value of this parameter in advance by computer simulations. In this paper, mergence and partition operation is designed to create the neighboring solution and no parameter is needed any longer. In [19], three clustering concepts, under-partitioned state, optimal-partitioned state, and over-partitioned state, are given to describe the variation of two partition functions so as to establish the cluster validity index. In this article, we introduce these concepts to explain why and how we establish the neighboring solution by mergence and partition operation. In general, for a cluster, there are only three partition states, under-partitioned state, optimal-partitioned state, and over-partitioned state. In over-partitioned case, an original cluster is improperly divided into several parts. In under-partitioned case, more than two original clusters or parts of them are improperly grouped together. Only in optimal-partitioned one, all original clusters are correctly partitioned. For a suboptimal clustering solution, there must be the under-partitioned cluster and the over-partitioned cluster. Therefore, it is seen that further partitioning the under-partitioned cluster and merging the over-partitioned cluster are natural and suitable for establishing the neighboring solution and exploring the correct clustering result. By improving all improperly partitioned clusters, we can expect to achieve the proper result at last. Here, we randomly perform one partition and one mergence on solution X_c, keep the number of clusters constant, and form the neighbor. As the increase of the number of iterations, this operation are repeatedly performed on suboptimal solutions and the proper solution will be finally achieved. Mergence and partition operation includes four sub-operations: mergence cluster selection, partition cluster selection, cluster mergence, and cluster partition. Here, the cluster to be merged C_m and the cluster to be partitioned C_p are randomly determined. For cluster C_m, its belonging objects will be reassigned to their respective nearest clusters. That is, object $\mathbf{x}_i \in C_m$ is reassigned to cluster C_j, iff

$$\|\mathbf{x}_i - \mathbf{c}_j\|^2 < \|\mathbf{x}_i - \mathbf{c}_k\|^2 \tag{8}$$

where $k, j = 1, \ldots, K$, $C_j, C_k \neq C_m$, and $C_j \neq C_k$. After this sub-operation, cluster C_m disappears and the number of clusters decreases by one. Meanwhile, For cluster C_p, we view objects belonging to cluster C_p as a new data set, and adopt iteration methods such as K-means algorithm to divide its belonging objects into two new clusters. Here, K-means algorithm is chosen to perform this task by computer simulations. After this sub-operation, cluster C_p is divided into two new clusters and the number of clusters increases by one. Above four steps are performed on solution X_c and the neighboring solution X' is established.

3 Experimental Results

In order to analyze the performance of the proposed algorithm, we firstly evaluate the individual contributions made by different operations. Then the proposed algorithm is applied to seven data sets and compared with SAC, NMC, and KNMC. These experimental data sets are chosen because they represent different situations and provide the extensive tests of the adaptability of the proposed algorithm. Simulation experiments are conducted in Matlab on an Intel Pentium III processor running at 800MHz with 128MB real memory. Each experiment includes 20 independent trials.

3.1 Performance Evaluation

In this section, the experiments are performed to compare performance of different modules. Due to space limitations, here, the well-known data set, German Towns with eight clusters, is chosen to show the comparison results. For other experimental data sets, the similar results are obtained.

Three local improvement operations (No operation, K-means operation, and DHB operation) adopted by NMC, KNMC, and INMC, respectively, are compared. In NMC, there is no local improvement operation. Here, the probability threshold is used to create the neighboring solution. The best results obtained by the methods equipped with different operations in the process of iterations are compared as shown in Figure 2. It is seen that No operation is the worst. For other two operations, it seems that their results are almost equal to each other. But after No operation is removed, the real results are shown as Figure 3. It is clear that K-means operation is obviously inferior to DHB operation. As a result, the algorithm equipped with DHB operation can attain the best results more quickly and stably than ones with other two operations.

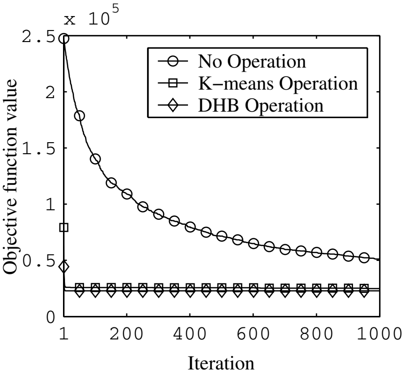

Fig. 2. Comparison of three operations for improving solution X_c

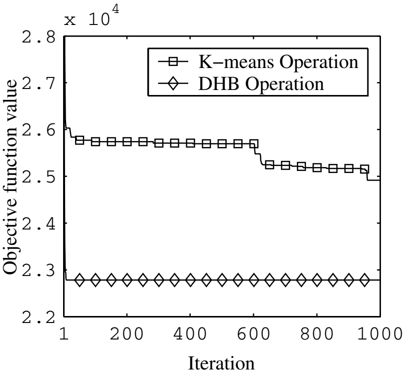

Fig. 3. Comparison of K-means operation and DHB operation

We now discuss the issue of creating the neighboring solution. Here, to compare performance of the probability threshold and mergence and partition operation, we do not adopt the local improvement operation to improve the current solution. Figure 4 shows that mergence and partition operation is far superior to the probability threshold and greatly improve the performance of the clustering algorithm. Without the cooperation of the local improvement operation, the neighboring solution provided by mergence and partition operation still accelerates the clustering algorithm to attain the best result stably and quickly. Therefore, it can be expected the combination of DHB operation and mergence and partition operation can further improve the performance of the clustering method.

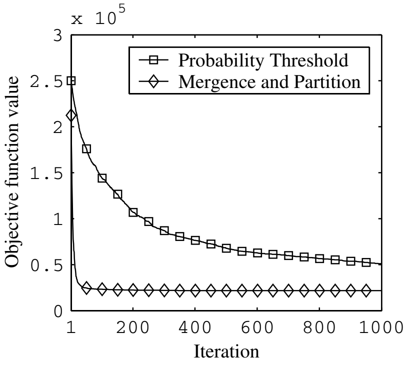

Fig. 4. Comparison of two modes for creating the neighboring solution

3.2 Performance Comparison

In this paper, seven data sets are chosen to perform computer simulations besides the ones adopted in [10]. Two well-known data sets are added: German Towns [2] and British Towns [20]. Here, we consider two cases: one is that the number of clusters is variable; the other is that this parameter is fixed. Among data sets, the number of clusters in German Towns varies in the range $[4, 10]$. We label them as GT4C, GT5C, GT6C, GT7C, GT8C, GT9C, and GT10C, respectively. This data set consists of Cartesian coordinates of 59 towns in Germany. The case of British Towns is the same as that of German Towns. This data set is composed of 50 samples each of four variables corresponding to the first four principal components of the original data. In other data sets, the number of clusters is fixed. The detail descriptions of these five data sets (Data52, Data62, Iris, Crude Oil, and Vowel) can be found in [10].

In this paper, our aim is to improve the noising method for the clustering problem under consideration and to further explore better results than those of KNMC and even SAC. In [10], it is shown that KNMC is better than K-means algorithm, GAC, and TSC. Therefore, we here focus on SAC, NMC, KNMC, and INMC. For SAC, according to the recommendation of the reference, the number of iterations at a temperature is set to be 20, the initial annealing temperature is set to be 100, α is set to be 0.05, and the terminal annealing temperature is set to be 0.01. In [10], The choice of the algorithm parameters is determined by computer simulations as follows: the noise range is equal to 10, the terminal noise rate is equal to 0, the original noise rate is equal to 10, the number of iterations at the fixed noise rate is equal to 20, and the total number of iterations is equal to 1000. For INMC, its parameter settings are the same as those of NMC and KNMC.

Before conducting comparison experiments, we analyze the time complexities of methods adopted in this article. The time complexities of SAC, NMC, and KNMC are $O(GN_sKmN)$, $O(N_tmN)$, and $O(N_tKmN)$, respectively, where G denotes the number of iterations during the process that the annealing temperature drops, N_s denotes the number of iterations at the fixed temperature, and N_t denotes the total number of iterations in the noising method. It is known that the cost of NMC is lower than that of KNMC, but the performance of NMC is far inferior to that of KNMC. For INMC, the complexity of DHB operation is $O(KmN)$. The complexity of mergence and partition operation is $O(KmN)$. Therefore, the time complexity of INMC is $O(N_tKmN)$ that is equal to that of KNMC. Under this condition, the complexity of SAC is over thrice as much as those of INMC and KNMC.

Table 1. Comparison of the clustering results of four methods for German Towns

		SAC	NMC	KNMC	INMC
GT4C	Avg	49600.59	75063.93	51610.14	49600.59
	SD	0.00	7917.87	6652.60	0.00
	Min	49600.59	63245.97	49600.59	49600.59
GT5C	Avg	39496.39	67157.79	40075.44	39091.02
	SD	783.34	6897.88	1094.02	376.04
	Min	38716.02	58374.91	38716.02	38716.02
GT6C	Avg	32220.44	62077.15	33837.61	31502.50
	SD	1548.73	7178.90	1369.15	975.98
	Min	30535.39	49445.41	30535.39	30535.39
GT7C	Avg	26964.11	54509.46	29009.23	24511.56
	SD	1707.07	6077.62	2146.84	136.82
	Min	24432.57	40164.41	25743.20	24432.57
GT8C	Avg	22603.12	52753.38	24496.94	21573.29
	SD	1458.92	5527.67	1591.55	153.88
	Min	21499.99	45283.29	22114.03	21483.02
GT9C	Avg	19790.99	47585.86	21746.58	18791.13
	SD	420.20	4602.09	1925.25	175.83
	Min	19130.63	35490.28	19521.02	18550.44
GT10C	Avg	18028.90	42796.75	20451.51	16515.07
	SD	633.67	4078.29	1643.82	125.47
	Min	16864.78	35015.27	18462.07	16307.96

The average (Avg), standard deviation (SD), and minimum (Min) values of the clustering results of four methods for German Towns are compared as shown in Table 1. In face of German Towns in which the number of clusters is variable, NMC is the worst and fails to attain the best values even once within specified iterations and its best values obtained are far worse than the best ones. KNMC equipped with K-means operation can attain much better results than NMC. KNMC can attain the optimal results of German Towns when the number of clusters is small. But when this number is greater than and equal to seven, KNMC cannot obtain the ideal results any longer. SAC spending much more computational resource than KNMC obtains better performance than KNMC as stated in [10]. SAC can attain the optimal results of German Towns when the number of clusters is up to seven. As the increase of this number, it does not attain the best results but its results are still superior to those of KNMC.

With the cooperation of DHB operation and mergence and partition operation, INMC can achieve the best value in each case. Its stability and solution quality are far superior to those of NMC, KNMC, and even SAC. Meanwhile, its time complexity the same as that of KNMC does not increase.

The average (Avg), standard deviation (SD), and minimum (Min) values of the clustering results of four methods for British Towns are compared as shown in Table 2. In face of British Towns, NMC is still the worst and fails to attain the best value in each case. At this time, KNMC can attain the optimal results of British Towns when the number of clusters is up to five. As the increase of the number of clusters, KNMC cannot obtain the best results any longer. In face of British Towns, the performance of SAC also becomes bad. It only attains the best values of British Towns with four and six clusters. But SAC still obtains better performance than KNMC in most case. In face of British Towns with different clusters, INMC can still attain the best value in each case. It is shown that the stability and solution quality of INMC are far superior to those of NMC, KNMC, and SAC.

Table 2. Comparison of the clustering results of four methods for British Towns

		SAC	NMC	KNMC	INMC
BT4C	Avg	180.91	213.74	182.05	180.91
	SD	0.00	13.05	1.96	0.00
	Min	180.91	186.25	180.91	180.91
BT5C	Avg	160.56	189.45	162.76	160.23
	SD	0.00	9.82	3.12	0.00
	Min	160.56	172.64	160.23	160.23
BT6C	Avg	145.37	178.18	147.29	141.46
	SD	3.30	10.17	2.97	0.00
	Min	141.46	167.61	142.30	141.46
BT7C	Avg	130.26	175.20	132.69	126.60
	SD	2.45	12.01	3.86	0.29
	Min	128.68	156.40	128.28	126.28
BT8C	Avg	120.07	163.48	121.18	113.82
	SD	3.01	8.93	3.96	0.57
	Min	114.07	141.67	116.65	113.50
BT9C	Avg	111.18	155.78	111.30	103.24
	SD	2.47	9.83	3.25	0.22
	Min	103.75	142.75	104.31	102.74
BT10C	Avg	100.71	148.21	103.14	92.81
	SD	3.36	10.69	4.14	0.17
	Min	93.19	131.07	98.47	92.68

After considering the case in which the number of clusters is variable, we focus on the other case. The average (Avg), standard deviation (SD), and minimum (Min) values of the clustering results of four methods for other five data sets are compared as shown in Table 3. In these experiments, the number of clusters is constant. As stated in [10], NMC is the worst, KNMC is the second, and SAC is the best. SAC can attain the best values of Data52, Iris, and Crude Oil in all trials. But after INMC is considered, more promising results are expected. INMC can stably obtain the best values of Data52, Data62, Iris, and Crude Oil in all trials. For Vowel, its solution quality and stability are much better than other three methods.

Table 3. Comparison of the clustering results of four methods for Data52, Data62, Iris, Crude Oil, and Vowel

		SAC	NMC	KNMC	INMC
Data52	Avg	488.02	2654.52	488.69	488.02
	SD	0.00	55.52	0.58	0.00
	Min	488.02	2557.31	488.09	488.02
Data62	Avg	1103.11	19303.58	1230.02	543.17
	SD	366.63	422.77	1382.50	0.00
	Min	543.17	18005.98	543.17	543.17
Iris	Avg	78.94	302.99	85.37	78.94
	SD	0.00	37.43	19.26	0.00
	Min	78.94	242.15	78.94	78.94
Crude Oil	Avg	1647.19	1995.44	1647.27	1647.19
	SD	0.00	124.27	0.12	0.00
	Min	1647.19	1787.43	1647.19	1647.19
Vowel	Avg	31941263.99	250796549.46	31554139.24	31389900.02
	SD	1205116.61	2866658.66	1209301.09	412724.00
	Min	30720909.84	245737316.31	30718120.60	30690583.33

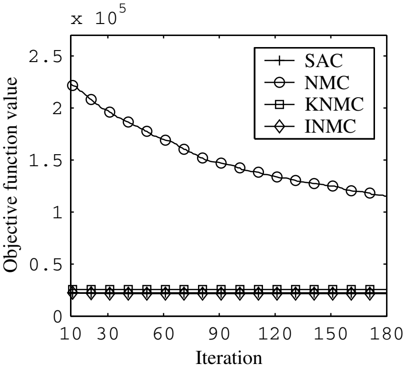

Fig. 5. Comparison of four methods for German Towns

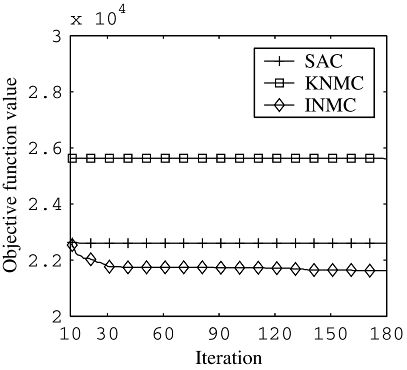

Fig. 6. Comparison of SAC, KNMC, and INMC for German Towns

In order to understand the performance of four methods better, we use German Towns with eight clusters to show the iteration process. In Figure 5, it is seen that NMC is obviously much inferior to other three methods. For other three algorithms, it seems that their results are almost equal to each other. But after NMC is removed, the real results are shown as Figure 6. It is seen that INMC is superior to SAC and KNMC, which shows that without the increase of the time complexity, the performance of the noising method based clustering algorithm can be greatly improved by introducing proper components into the algorithm framework.

4 Conclusions

In this article, in order to further improve the performance of the noising method based clustering algorithm, a novel algorithm called INMC is proposed. Two new operations are described in detail, DHB operation and mergence and partition operation. In the algorithm framework, DHB operation is used to modulate the current solution obtained in the process of iterations and to accelerate the convergence speed of INMC, and mergence and partition operation is developed to establish the neighboring solution. With the same time complexity as KNMC, INMC can get much better results more quickly and stably than NMC and KNMC. Moreover, compared with SAC, INMC spends much less resource and obtains much better results, which is not solved in [10]. In future, the estimation of the number of clusters should be considered. Meanwhile, improving the stability of the proposed algorithm to the best results in complicate cases will be the subject of future publications.

Acknowledgements

This research was supported in part by the National Natural Science Foundation of China (Grants 60473020, 60273049, 90104005) and State Key Laboratory for Novel Software Technology at Nanjing University.

References

1. Brucker, P.: On the complexity of clustering problems. Lecture Notes in Economics and Mathematical Systems. **157** (1978) 45–54
2. Spath, H.: Cluster analysis algorithms. Wiley, Chichester (1980)
3. Jain, A.K., Dubes, R.: Algorithms for clustering data. Prentice-Hall, New Jersey (1988)
4. Selim, S.Z., Ismail, M.A.: K-Means-type algorithm: generalized convergence theorem and characterization of local optimality. IEEE Trans Pattern Anal Mach Intell. **6**(1984) 81–87
5. Murthy, C.A., Chowdhury, N.: In search of optimal clusters using genetic algorithms. Pattern Recognit Lett. **17** (1996) 825–832
6. Babu, G.P., Murthy, M.N.: Clustering with evolutionary strategies. Pattern Recognit. **27** (1994) 321–329
7. Babu, G.P.: Connectionist and evolutionary approaches for pattern clustering. PhD dissertation. Indian Institute of Science, India (1994)
8. Al-sultan, K.S.: A tabu search approach to the clustering problem. Pattern Recognit. **28** (1995) 1443–1451
9. Bandyopadhyay, S., Maulik, U., Pakhira, M.K.: Clustering using simulated annealing with probabilisitc redistribution. Int J Pattern Recognit Artif Intell. **15** (2001) 269–285
10. Liu, Y.G., Liu, Y., Chen, K.F.: Clustering with noising method. Lecture Notes in Artificial Intelligence. **3584** (2005) 209–216
11. Charon, I., Hudry, O.: The noising method: a new method for combinatorial optimization. Oper Res Lett. **14** (1993) 133–137
12. Chelouah, R., Siarry, P.: Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. Eur J Oper Res. **148** (2003) 335–348

13. Chelouah, R., Siarry, P.: A hybrid method combining continuous tabu search and Nelder-Mead simplex algorithms for the global optimization of multiminima functions. Eur J Oper Res. **161** (2005) 636–654
14. Charon, I., Hudry, O.: The noising method: a generalization of some metaheuristics. Eur J Oper Res. **135** (2001) 86–101
15. Duda, R.O., Hart, P.E.: Pattern classification and scene analysis. Wiley, New York (1972)
16. Ismail, M.A., Selim, S.Z., Arora, S.K.: Efficient clustering of multidimensional data. In: Proceedings of 1984 IEEE International Confference on System, Man, and Cybernetics. Halifax. (1984) 120–123
17. Ismail, M.A., Kamel, M.S.: Multidimensional data clustering utilizing hybrid search strategies. Pattern Recognit. **22** (1989) 75–89
18. Zhang, Q.W., Boyle, R.D.: A new clustering algorithm with multiple runs of iterative procedures. Pattern Recognit. **24** (1991) 835–848
19. Kim, D.J., Park, Y.W., Park, D.J.: A novel validity index for determination of the optimal number of clusters. IEICE Trans Inf Syst. **E84-D** (2001) 281–285
20. Chien, Y.T.: Interactive Pattern Recognition. Marcel-Dekker, New York (1978)