Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

A clustering algorithm using the tabu search approach with simulated annealing

S.-C. Chu[1,2] & J.F. Roddick[1]
[1]*School of Informatics and Engineering, Flinders University of South Australia, Adelaide, Australia.*
[2]*Department of Industrial Engineering and Management, National Kaohsiung Institute of Technology, Taiwan.*

Abstract

In this paper, an algorithm for cluster generation using tabu search approach with simulated annealing is proposed. The main idea of this algorithm is to use the tabu search approach to generate non-local moves for the clusters and apply the simulated annealing technique to select suitable current best solution so that speed the cluster generation. Experimental results demonstrate the proposed tabu search approach with simulated annealing algorithm for cluster generation is superior to the tabu search approach with Generalised Lloyd algorithm.

1 Clustering

Clustering is the process of grouping patterns into a number of clusters, each of which contains the patterns that are similar to each other in some way. The existing clustering algorithms can be simply classified into the following two categories: hierarchical clustering and partitional clustering [1]. The hierarchical clustering operates by partitioning the patterns into successively fewer structures. This method gives rise to a dendogram in which the patterns are formed a nested sequence of partitions. Hierarchical procedures can be either agglomerative or divisive. An agglomerative clustering approach is a process in which each pattern is placed in its own cluster and these atomic clusters are gradually merged into larger and larger clusters until the desired objective is attained. A divisive clustering approach reverses the process of agglomerative clustering approach by starting with all patterns in one cluster and subdividing into several smaller clusters.

Partitional clustering procedures typically start with the patterns partitioning into a number of clusters and divide the patterns by increasing the number of partitions. The most popular class of partitional clustering methods are the prototype-based clustering algorithms. In the prototype-based clustering algorithms, each cluster is represented by a prototype, and the sum of distance from the pattern to the prototype is usually used as the objective function. Normally, the prototype is the centre of the cluster.

The K-means (or GLA, or LBG) algorithm [2] is one of the prototype-based cluster algorithms. It is a descent algorithm in the sense that at each iteration, the average distortion is reduced. For this reason, the K-means algorithm can get trapped in local optima. The performance of the K-means algorithm depends on the number of optima and the choice of the initial condition. The K-means (or GLA, or LBG) algorithm can be described as follows:

Step 1: Select N clustering patterns as the initial centroids of the clusters randomly. Set $n = 0$, where n is the iteration count.

Step 2: Find the nearest centroid to each clustering pattern. Put X_j in the partitioned set (or cluster) P_i if C_i is the nearest centroid to X_j.

Step 3: After obtaining the partitioned sets $P = (P_i; 1 \geq I \geq N)$ increment n and calculate the overall average distortion

$$D_n = \frac{1}{T} \sum_{i=1}^{N} \sum_{j=1}^{T_i} D(X_j^{(i)}, C_i)$$

where $P_i = \{X_1^{(i)}, X_2^{(i)}, \ldots X_{T_i}^{(i)}\}$. T_i is the number of clustering patterns belonging to the partitioned set P_i.

Step 4: Find centroids of all disjoint partitioned sets P_i by

$$C_i = \frac{1}{T} \sum_{j=1}^{T_i} X_j^{(i)}.$$

Step 5: If $(D_{n-1} - D_n)/D_n > \epsilon$, go to step 2; otherwise terminate the program. Here ϵ is a small distortion threshold.

The tabu search approach was proposed by Glover [3]. The idea of tabu searching is to forbid some search directions at a present iteration in order to avoid cycling, and bthus enable the process to jump off local optima. Tabu list memory is used to partially or completely record the movement of elements from the current best solution to its selected neighbour. In this paper, the tabu search approach is applied to generate the non-local moves for the clusters and the simulated annealing technique [4, 5] is utilized to decide the current best solution for generating the test clusters for the next iteration in order to reduce the computation time for cluster generation. In

Section 2, the basic concept of the tabu search approach will be introduced. In Section 3, the theory of simulated annealing will be described. The tabu search approach combining with the simulated annealing algorithm for cluster generation will be proposed in Section 4. Test results are shown in Section 5.

2 Tabu Search Approach

Tabu search approach [6–9] is a higher-level method for solving optimization problems. It is designed to optimize the problem by performing a sequence of moves that lead the procedure from one test solution to another. Each move is selected randomly from a set of currently available alternatives. The new test solutions are generated by performing the moves from the current best solution. The current best solution is the test solution which is not a tabu solution or is a tabu solution but satisfies the aspiration criterion. A tabu solution is a solution in which the elements of the solution are partially or completely recorded in the tabu list memory. It is called aspiration if the test solution is in the tabu condition but is the best solution for all iterations up to that point.

Al-Sultan [10] applied the tabu search approach to cluster the patterns. A set of test solutions is generated from the current solution randomly. For each pattern, a random number, $0 \leq R \leq 1$, is generated. If $R \geq P_t$ then this pattern is assigned to cluster i, where i is randomly generated but not the same cluster as in the current solution, $0 \leq i \leq N$, and P_t is the predefined probability threshold; otherwise it is partitioned to the same cluster as in the current best solution. From the best solution to the worst one, if the aspiration level is satisfied or the tabu conditions are avoided, then this test solution is chosen as the current best solution and each pattern assigned to the ith cluster in the current best solution is recorded in the tabu list memory. If all test solutions are tabu, then the test solutions are generated from the current best solution again. The program is terminated if the predefined distance or the maximum number of iterations is reached. The tabu search approach for clustering patterns in paper [10] is illustrated in Figure 1.

The tabu search approach is also combined with the K-means (or GLA) algorithm for cluster generation [11] and the centres of the clusters are taken as the codevectors for vector quantization. The performance comparison of this algorithm is better than both GLA algorithm and the tabu search approach for cluster generation. In [11], there are two types of test solutions: partition based test solution and codebook based test solution. In the partition based test solution, all the training patterns form the test solutions. In the codebook based test solution, the elements of the test solution are the centres of the clusters. The basic prosedure of the tabu search approach with GLA algorithm for clustering patterns is illustrated in Figure 2.

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

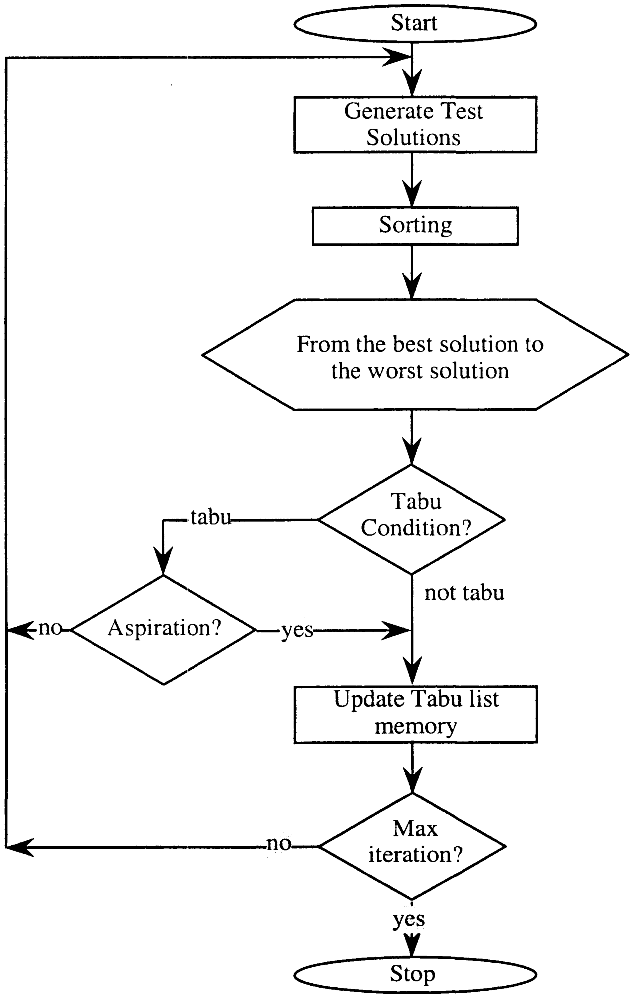

Fig. 1. Flowchart of the tabu search approach for clustering patterns

3 Simulated Annealing

Simulated annealing [4] is a random search method which has been presented for combinatorial optimization problems. Vecchi and Kirkpatrick applied a simulated annealing method to the optimization of a wiring problem [5]. Gamal *et al.* also used the method of simulated annealing to construct good source codes, error-correcting codes and spherical codes [12]. The simulated annealing was also applied to cluster generation for vector quantization [13, 14]. The simulated annealing for clustering patterns can be described as follows:

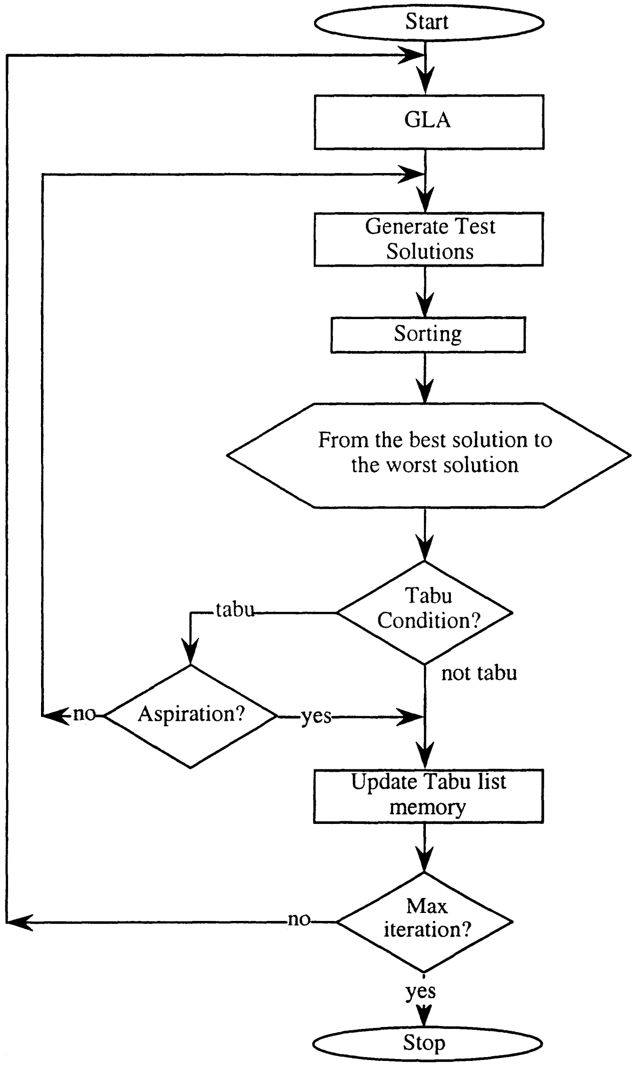

Fig. 2. Flowchart of the tabu search approach with GLA algorithm

Step 1: The training pattern X_j, $j = 1, 2, \ldots T$, is partitioned into the cluster, S_i, $i = 1, 2, \ldots N$, randomly. Set $n = 0$ and calculate the centre of the cluster.

$$C_i = \frac{1}{|S_i|} \sum_{X_j \in S_i} X_j$$

where $|S_i|$ denotes the number of training patterns in the cluster S_i.

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

Step 2: The clusters are perturbed by randomly selecting a pattern and moving this pattern from its current cluster to the different randomly selected cluster. Calculate the new centroids.

Step 3: The change in distortion ΔD is defined as the distortion of current clusters minus the distortion of previous clusters. The perturbation is accepted if

$$e^{-\frac{\Delta D}{\hat{T}_n}} > \gamma,$$

where γ is a random value generated uniformly on the interval [0,1].

Step 4: If the distortion of the current clusters reaches the desired value or the iteration count n reaches the predetermined value, then terminate the program; otherwise, increment n and go to step 2.

The simulated annealing algorithm starts with an initial temperature $\hat{T}_0$. The temperature sequence $\hat{T}_0, \hat{T}_1, \hat{T}_2, \ldots$ are positive numbers which is called an annealing schedule where

$$\hat{T}_0 \geq \hat{T}_1 \geq \hat{T}_2 \ldots$$

and

$$\lim_{n \to \infty} \hat{T}_n = 0$$

In this paper, simulated annealing is applied to select the suitable current best solution to improve performance as compared with the combination of the tabu search approach with the GLA algorithm.

4 The Proposed Algorithm

Although the tabu search approach can avoid the cycling condition so that remaining in local optimum can be avoided, it can be further improved by introducing counters to calculate the frequency of the move for each element, i.e., the non-local moves from the current best solution is limited by counting the frequency of the move. In this modified tabu search approach, if the distortion of the best solution of all iterations keep the same for some fixed number of iterations, we reset the current best solution using the best solution of all iterations. Simulated annealing is used to decide which test solution is suitable to be the current best solution for generating the test solutions for next iteration. The proposed tabu search approach with simulated annealing algorithm for cluster generation is as follows:

Step 1: Generate an initial solution C_{init} using GLA algorithm. Set $C_{curr} = C_{best} = C_{init}$. Set a counter $Count_j$ for each element in the solution, $j = 1, 2, \ldots T$. T is the total number of training patterns.

Step 2: Generate S test solutions C_i by changing the values of elements from the current best solution C_{curr} with probability P_t. For each test solution, if the qth element of the test solution is changed and the value of $Count_p$ is smaller than v, then increment $Count_p$. If all the values of the counters are equal to v, then reset all counters to zero.

Step 3: Calculate the mean squared error $D(C_i)$ for each test solution and sort these test solutions using mean squared error in increasing order.

Step 4: From the best test solution to the worst test solution, if $D(C_{best}) > D(C_i)$, set $C_{curr} = C_i$ and go to step 6; otherwise go to next step.

Step 5: Calculate $\Delta D = D(C_{curr}) - D(C_i)$ and set $\hat{T}_n = \hat{T}_0 \alpha^n$. Generate a random number γ $(0 \leq \gamma \leq 1)$, if $\gamma < e^{\frac{\Delta D}{\hat{T}_n}}$ then set $C_{curr} = C_i$ and go to step 6. Otherwise if $i = S$, then go to step 2; otherwise increment i and go to step 4.

Step 6: If $D(C_{best})$ remains unchanged for M iterations, then set $C_{curr} = C_{best}$, clear the counter $Count_j, j = 1, 2, \ldots T$. If $D(C_{best}) > D(C_{curr})$ then set $C_{best} = C_{curr}$. If the number of iterations has reached, then terminate the program; otherwise go to step 2.

5 Experiments

Experiments were carried out to test the clustering distortion using GLA algorithm, tabu search approach with GLA algorithm, and tabu search approach with simulated annealing. The initialization of the tabu search approach with simulated annealing can be randomly generated or obtained from GLA algorithm. Since the codebook based test solution for tabu search approach with GLA algorithm is better than the partitioned based test solution in paper [11], we only adopt the codebook based test solution for the tabu search approach with GLA algorithm for cluster generation. The evaluation function of the clustering distortion is the mean squared error (MSE) as following:

$$MSE = \frac{1}{kT} \sum_{i=1}^{N} \sum_{j=1}^{T_i} D(X_j^{(i)}, C_i)$$

where T_i is the number of clustering patterns belonging to the ith cluster. T is the total number of training patterns and k is the number of dimension for each pattern. $X_j^{(i)}$ is the jth pattern belonging to the ith cluster.

Three 512×512 images, i.e., “Lena” image, “Baboo” image, and “Pepper” image are used as the test material. 4×4 pixel blocks are taken from these gray images as the training patterns, i.e., the number of dimensions is 16. The parameters used for the test solutions S, number of training patterns T, probability threshold P_i, the limit value of counter v, the limit number of the same current best iterations, the initial temperature $\hat{T}_0$ and the value of α are 20, 16384, $\frac{100}{16384}$, 3, 30, 500 and 0.99, respectively.

Experimental results demonstrate the proposed algorithm can reduce the average distortion by 0.3% $\sim$ 9.6% and 10% $\sim$ 13% comparing with the GLA algorithm and tabu search approach with GLA algorithm for the clustering at the lst hour, respectivity. At the clustering time of 8 hours, the proposed algorithm will reduce the average distortion by 1.67% $\sim$ 9.99% and 1.65% $\sim$ 5.15% comparing with the GLA algorithm and tabu search approach with GLA algorithm, respectivity.

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

Algorithm	Tabu-SA with GLA initialization	Tabu-SA with random initialization	GLA	Tabu-GLA
Initialization	92.476	99.403	72.360	68.634
1 hour	55.912	57.354	59.889	64.309
2 hour	55.431	56.310	59.889	60.234
3 hour	55.431	56.129	59.889	59.445
4 hour	55.431	56.129	59.889	59.280
5 hour	55.431	56.129	59.889	59.216
6 hour	55.431	56.129	59.889	59.018
7 hour	55.431	56.129	59.889	58.501
8 hour	55.431	56.129	59.889	58.434

Fig. 3. Performance comparison using Lena image as the training patterns.

Algorithm	Tabu-SA with GLA initialization	Tabu-SA with random initialization	GLA	Tabu-GLA
Initialization	355.181	360.119	330.381	323.543
1 hour	308.624	305.591	309.636	312.276
2 hour	306.797	304.385	309.636	311.043
3 hour	305.628	304.141	309.636	309.958
4 hour	305.005	303.492	309.636	309.693
5 hour	304.575	303.218	309.636	309.298
6 hour	304.399	302.906	309.636	309.264
7 hour	304.304	302.772	309.636	309.264
8 hour	304.158	302.695	309.636	309.264

Fig. 4. Performance comparison using Baboo image as the training patterns.

Algorithm	Tabu-SA with GLA initialization	Tabu-SA with random initialization	GLA	Tabu-GLA
Initialization	127.286	140.189	82.332	75.351
1 hour	60.328	61.788	66.788	67.041
2 hour	60.117	60.891	66.788	66.005
3 hour	60.117	60.891	66.788	64.969
4 hour	60.117	60.891	66.788	64.409
5 hour	60.117	60.891	66.788	64.185
6 hour	60.117	60.891	66.788	63.804
7 hour	60.117	60.891	66.788	63.496
8 hour	60.117	60.891	66.788	63.381

Fig. 5. Performance comparison using Pepper image as the training pattern.

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

6 Conclusions

The main idea of this paper is to modify the tabu search approach by introducing the counter to limit the non-local moves from the current best solution and forbid the current best solution from remaining the same more than some fixed number of iterations. In particular, the simulated annealing method is applied to select the suitable current best solution so that the performance is improved compared with the tabu search approach with GLA algorithm for cluster generation.

References

1. A. Jain and R. Dubes, *Algorithms for clustering data.* Englewood Cliffs, New Jersey: Prentice Hall, 1988.
2. Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.
3. F. Glover, "Tabu search, part i," *ORSA Journal of Computing*, vol. 1, no. 3, pp. 190–206, 1989.
4. S. Kirkpatrick, C. Gelatt Jr., and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 200, no. 4598, pp. 671–680, 1983.
5. M. Vecchi and S. Kirkpatrick, "Global wiring by simulated annealing," *IEEE Transactions on Computer-Aided Design*, vol. 2, no. 4, pp. 215–222, 1983.
6. J. Skorin-Kapov, "Tabu search applied to the quadratic assignment problem," *ORSA Journal of Computing*, vol. 2, no. 1, pp. 33–45, 1990.
7. J. Pan and S. Chu, "Non-redundant vq channel coding using tabu search strategy," *Electron. Lett.*, vol. 32, no. 17, pp. 1545–1546, 1996.
8. F. Glover and M. Laguna, *Tabu Search.* Kluwer Academic Publishers, 1997.
9. S. Chu and H. Fang, "Genetic algorithms vs. tabu search in timetable scheduling," in *Third International Conference on Knowledge-Based Intelligent Information Engineering Systems*, pp. 492–495, 1999.
10. K. Al-Sultan, "A tabu search approach to the clustering problem," *Pattern Recognition*, vol. 28, no. 9, pp. 1443–1451, 1995.
11. P. Frnti, J. Kivijrvi, and O. Nevalainen, "Tabu search algorithm for codebook generation in vector quantization," *Pattern Recognition*, vol. 31, no. 8, pp. 1139–1148, 1998.
12. A. Gamal, L. Hemachandra, I. Shperling, and V. Wei, "Using simulated annealing to design good codes," *IEEE Transactions on Information Theory*, vol. 33, no. 1, pp. 116–123, 1987.
13. A. Cetin and V. Weerackody, "Design of vector quantizers using simulated annealing," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 12, pp. 1550–1555, 1988.
14. J. Vaisey and A. Gersho, "Simulated annealing and codebook design," in *IEEE International Conference on Acoustics Speech and Signal Processing*, pp. 1176–1179, 1988.