Negative Association Rule Mining through Particle Swarm Optimization

Akhilesh Chauhan
Department of Information Technology
SVITS,Indore,India

Ashish Bansal, Ph.D.
Department of Information Technology
SVITS,Indore,India

ABSTRACT
Mining hidden pattern from existing databases is an important topic in field of data mining. The knowledge obtained from these databases is used in different applications like in market basket analysis. Association Rules are important to discover the relationships among the attributes in a database. In general the rules generated by Association Rule Mining technique do not consider the negative occurrences of attributes in them, but by focusing on infrequent items generated in system we can predict the rules which contains negative attributes. This paper proposes an improved algorithm NAPSO based on Particle Swarm Optimization. The algorithm improves result provided by apriori algorithm.

Keywords
Association Rule Mining (ARM), Data Mining (DM), Negative Association Rule (NAR), Particle Swarm Optimization (PSO).

1. INTRODUCTION
In today's internet world, the amount of data stored in database applications continues to grow. This large amount of data contains knowledge, which can be utilized to improve decision making process of an organization. This knowledge discovery can be done in various ways available today, like through Association Rule Mining. The amount of knowledge also varies to a large extent from different kind of rules to predict values. In this paper we have considered Association Rule Mining and tried to improve this technique by applying Particle Swarm optimization (PSO) on the rules generated by it. An introduction about Association Rule Mining and PSO is given in the following sub-sections, followed by algorithm in section 3 which will describe the details of Negative Association Rule Mining, Apriori and PSO algorithm.Section 4 proposed algorithm and results in section 5 is discussed, followed by conclusion in the last section.

2. LITERATURE SURVEY
Introduced in 1993 [1], association rule mining has gained great deal of attention. In computer science and data mining, Apriori is a classic algorithm for learning association rules. Apriori is designed to operate on databases containing transactions. Apriori algorithm is an influential algorithm for mining frequent item sets for Boolean association rules. This algorithm contains a number of passes over the database. During pass k, the algorithm finds the set of frequent item sets L_k of length k that satisfies the minimum support requirement. The algorithm terminates when L_k is empty. A pruning step eliminates any candidate, which has a smaller subset.The Apriori Algorithm is an influential algorithm for mining frequent item sets for boolean association rules. Even today people use it for mining in KDD.

In brief, an association rule is an expression X=>Y, where X and Y are item sets. The meaning of this kind of rule is, given a database D containing say N transactions, where say T belongs to D is a transaction, then X=>Y expresses that whenever a transaction T contains X than T probably also contains Y. The confidence of an association rule, given as support is defined as the percentage of transactions containing Y in addition to X with regard to overall number of transactions containing X, i.e. the conditional probability (support(x and y)) / (support x).The introduction of these rules was similar to market-based data where rules like "A customer buys milk and Bread will also buy butter with a probability, say x %" is a famous example.

Mining Association rules has some limitations too. First of all the number of rules grows exponentially with the number of items. But this complexity is tackled with some latest algorithms which can efficiently prune the search space. Secondly, the problem of finding rules from rules, i.e. picking interesting rules from set of rules. The work tackling the second problem mainly support the user when browsing the rule set, e.g. [2] and the development of further useful quality measures on the rules, e.g. [3][4][5]. Thirdly, the problem that is being discussed in this paper is that, association rules do not utter the rules in which the negation of attributes is there. Like, say there are three attributes in the database X1, X2, X3, than rules like "If a customer takes X1 and not X2 than he will take X3 with a confidence of say c %" will not be provided by normal association rule mining. In order to generate these kinds of rules and also to tackle the second problem discussed above, i.e. to evolve quality rules, this paper is using PSO.

The PSO algorithm conducts search using a population of particles which correspond to individuals. A population of particles is initially randomly generated. Each particle represents a potential solution and has a position represented by a position vector. A swarm of particles moves through the problem space, with the moving velocity of each particle represented by a velocity vector. At each time step, a function representing a quality measure is calculated by using as input. Each particle keeps track of its own best position, which is associated with the best fitness it has achieved so far in a vector. Furthermore, the best position among all the particles obtained so far in the population is kept track as output. In addition to this global version, another local version of PSO keeps track of the best position among all the topological neighbours of a particle. At each time step, by using the individual best position, and global best position, a new velocity for particle is updated. The computation for PSO is easy. Furthermore, the flexibility of PSO to control the balance between local and global exploration of the problem space helps to overcome premature convergence of elite strategy in GA, and also enhances search ability.

3. BASIC THEORY

3.1 Negative Association Rule

Classic association rules consider only items enumerated in transactions. Such rules are referred to as positive association rules. Negative association rules also consider the same items, but in addition consider negated items (i.e. absent from transactions). Negative association rules are useful in many applications, for example in market-basket analysis to identify products that conflict with each other or products that complement each other [6]. Recently, mining negative association rules has received some attention and they have been proved to be useful in the real world. Many studies have shown that negative associations are as important as the traditional positive ones in practice [7]. Negative correlation in association rules was first mentioned in [8]. The first effort for finding negative rules was done by Savasere A. et al. [9].

Logically speaking, a negative association rule is a rule with the connective "¬", which corresponds to the absence or negative attribute of an item (item set). A rule of the form A→B is called a positive association rule and rules of the other forms (A→ ¬B, ¬A→ B and ¬A→ ¬B) are negative association rules [10]. For convenience rules are called by A→ ¬B by first negative form, ¬A→ B by second negative from and ¬A→ ¬B by third negative form. After finding the positive association rules, the discovered rules were used to generate possible negative association rules and pruned the redundant rules at last. The support and confidence of the negative association rules can make use of those of the positive association rules.

The support is given by the following formulas:

$$\text{supp}(\neg A) = 1 - \text{supp}(A) \quad (1)$$

$$\text{supp}(A \rightarrow \neg B) = \text{supp}(A) - \text{supp}(A \cup B) \quad (2)$$

$$\text{supp}(\neg A \rightarrow B) = \text{supp}(B) - \text{supp}(A \cup B) \quad (3)$$

$$\text{supp}(\neg A \rightarrow \neg B) = 1 - \text{supp}(A) - \text{supp}(B) + \text{supp}(A \cup B) \quad (4)$$

The confidence is given by the following formulas:

$$\text{conf}(A \rightarrow \neg B) = \frac{supp\ (A) - supp\ (A \cup B)}{supp\ (A)} \quad (5)$$

$$\text{conf}(\neg A \rightarrow B) = \frac{supp\ (B) - supp\ (A \cup B)}{1 - supp\ (A)} \quad (6)$$

$$\text{conf}(\neg A \rightarrow \neg B) = \frac{1 - supp\ (A) - supp\ (B) + supp\ (A \cup B)}{1 - supp\ (A)} \quad (7)$$

3.2 Apriori

The Apriori algorithm developed by [11] is a great achievement in the history of mining association rules. It is by far the most well-known association rule algorithm. This technique uses the property that any subset of a large item set must be a large item set. Also, it is assumed that items within an item set are kept in lexicographic order. These common item sets are extended with other individual items in the transaction to generate candidate item sets. However, those individual items may not be large. As we know that a superset of one large item set and a small item set will result in a small item set, these techniques generate too many candidate item sets which turn out to be small. The Apriori algorithm addresses this important issue. The Apriori generates the candidate item sets by joining the large item sets of the previous pass and deleting those subsets which are small in the previous pass without considering the transactions in the database. By only considering large item sets of the previous pass, the number of candidate large item sets is significantly reduced.

In the first pass, the item sets with only one item are counted. The discovered large item sets of the first pass are used to generate the candidate sets of the second pass using the apriori_gen() function. Once the candidate item sets are found, their supports are counted to discover the large item sets of size two by scanning the database. In the third pass, the large item sets of the second pass are considered as the candidate sets to discover large item sets of this pass. This iterative process terminates when no new large item sets are found. Each pass i of the algorithm scans the database once and determines large item sets of size i. L_i denotes large item sets of size i, while C_i is candidates of size i.

The apriori_gen() function as described in [Agrawal1994] has two steps. During the first step, L_{k-1} is joined with itself to obtain C_k. In the second step, apriori_gen() deletes all item sets from the join result, which have some (k-1)–subset that is not in L_{k-1}. Then, it returns the remaining large k-item sets.

Therefore, association rule mining has received a lot of attention. The traditional algorithms discover valid rules by exploiting support and confidence requirements, and use a minimum support threshold to prune its combinatorial search space.

3.3 Particle Swarm Optimization

The concept of PSO was first suggested by Kennedy and Eberhart in 1995[12].Particle swarm optimization (PSO) is inspired by the social behaviour observed in flocks of birds and schools of fish. In nature, there is a leader who leads the bird or fish group to move, as illustrated in Fig. 1.Most members of the group follow the leader. In PSO, a potential solution to the considered problem is represented by a particle, similar to the individuals in the bird and fish group. Each particle travels in the solution space and attempts to move toward a better solution by changing its direction and speed based on its own past experience and the information from the current best particle of the swarm. [12]

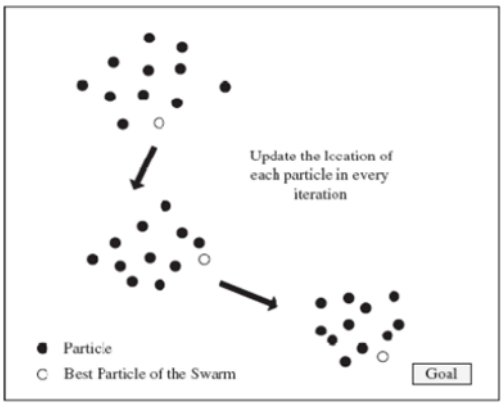

Fig 1: Concept of PSO

The procedure of PSO is described as follows:

3.3.1 Particle initialization:

An initial swarm of particles is generated in search space. Usually, the population size is decided by the dimension of problems.

3.3.2 Velocity and position update:

In each iteration, a new velocity value for each particle is calculated based on its current velocity, the distance from its previous best position, and the distance from the global best position. The new velocity value is then used to calculate the next position of the particle in the search space. The particle's velocity and position are dynamically updated as follows:

$$V_{id}^{new} = w \times V_{id}^{old} + c_1 \times rand \times \left(P_{id} - x_{id}^{old}\right) + c_2 \times rand \times \left(P_{gd} - x_{id}^{old}\right),$$

$$x_{id}^{new} = x_{id}^{old} + V_{id}^{new}.$$

The new velocity of a particle, V_{id}^{new} is updated by taking into consideration of the particle's previous velocity V_{id}^{old},and previous position, x_{id}^{old} . w = [0.5+ rand/2] is an inertia weight and rand is a uniformly generated random number between 0 and 1. The cognition parameter, c1, and social parameter, c2, are acceleration coefficients that are conventionally set to a fixed value 0–2.P_{id} is the previous individual best position of this particle and P_{gd} is the current global best position then calculates the new position of the particle, x_{id}^{new} [13]

a) Both particle's position and the global best position are far from the optimum and the particle velocity is low compared to its distance to the optimum [14].

b) Global best position is close to the optimum and the particle position is far from them resulting in a small improvement region and a large next position region.

3.3.3 Evaluation and update of best locations:

The fitness value of each particle is calculated by the objective function. The values of P_{id} and P_{gd} are then evaluated and replaced if better particle best position or global best position is obtained.

3.3.4 Termination:

Steps (b) and (c) are repeated iteratively until the termination condition is met.

4. PROPOSED WORK AND METHODOLOGY

The proposed algorithm NAPSO comprises two parts, mining and optimization. The first part provides procedures to obtain positive and negative items. In the second part of the algorithm, which is the main contribution of this study, the PSO algorithm is employed to obtain and optimize the negative association rules.First part proceed with particle swarm encoding, this step is similar to chromosome encoding of genetic algorithms. The next step is to generate a population of particle swarms according to the calculated fitness value. Finally, the PSO searching procedure proceeds until the stop condition is reached, which means the best particle is found. The support and confidence of the best particle can represent the minimal support and minimal confidence.

For the purpose of this study sample data is taken from market basket. It contains nine transaction and five items shown below. After reading data is transformed i.e. transactional data into binary type data, each recorded and stored as either 0 or 1. This approach can accelerate the database scanning operation, and it calculates support and confidence more easily and quickly. For instance, there are a total of only five different products in the database, so five cells exist for each transaction. According to the definition of association rule mining, the intersection of the association rule of item set X to item set Y (X→Y) must be empty. Items which appear in item set X do not appear in item set Y, and vice versa. Hence, both the front and back partition points must be given for the purpose of chromosome encoding. The item set before the front partition point is called "item set X," while that between the front partition and back partition points is called "item set Y." The fitness value in this study is utilized to evaluate the importance of each particle. The fitness value of each particle comes from the fitness function. Here, we employ the target function to determine the fitness function value as shown in

Fitness (k) = confidence(k) X support(k)

Fitness (k) is the fitness value of association rule type k.

Confidence (k) is the confidence of association rule type k.

Support (k) is the actual support of association rule type k.

The objective of this fitness function is maximization. The larger the particle support and confidence, the greater the strength of the association, meaning that it is an important association rule.

In order to apply the evolution process of the PSO algorithm, it is necessary to first generate the initial population. In this study, particles selected have larger fitness values in the population. The particles in this population are called initial particles.Applying PSO to association rule mining is the main part of this study. PSO is used as a module to mine best fitness value. The algorithmic process is quite similar to that of genetic algorithms, but the proposed procedures include only encoding, fitness value calculation, population generation, best particle search, and termination condition. Each of the steps in the PSO algorithm and the process of generating association rules are explained as follows:

First, the particle with the maximum fitness value in the population is selected as the "gbest." a method is used to constrain the search. The constrained method calculates the distance between the particle's new position and all the possible particles inside the constrained range before the particle's position is updated. Definitely, the particle with the smallest distance will be selected and treated as the particle's new position. The nearest possible particle is selected to be the target particle's new position. This method can prevent a particle from falling beyond the search space when its position is updated. Later on to speed up the execution NAPSO is applied. To complete particle evolution, the design of a termination condition is necessary. In this study, the evolution terminates when the fitness values of all particles are the same. In other words, the positions of all particles are fixed. Another termination condition occurs after 50(for study purpose) iterations and the evolution of the particle swarm is completed. Finally, after the best particle is found, its support and confidence are recommended as the value of minimal support and minimal confidence.

The algorithm structure is illustrated below.

Step 1: read data from database.

Step 2: transform data into binary.

Step 3: apply apriori algorithm. to find positive items and negative items by applying formulas explained above.

Step 4: separate generated result.

Step 5: Now take inputs are items, min supp, min confidence and fitness value

Step 6: apply PSO algorithm, generate population for pso.

Computing fitness of each individual

If (rules_sup >= min_sup) && (rules_conf >= min_conf)

Fitness = rules_sup*rules_conf;

Else

Fitness = rules_sup*rules_conf*fit;

End;

If fitness of rule >= fit than select that rule as efficient association rule

Else go forward.

Step 7: rules are obtained

Step 8: Finding the best particle in population, the nearest possible particle is selected to be the target particle's new position.

Step 9: Apply NAPSO for optimization.

new_pop() = abs(round((init_pop() - init_pop(ft_loc))/2))

new_pop(i) = abs(round((init_pop()- init_pop(ft_loc))/(2*iter)));

Step 10: Result Optimized rule generated.

5. EXPERIMENTAL RESULTS

As we mention for our algorithms table of sample data set of goods over which we performed our proposed approach.

Table 1. Sample item set

T.id.	Item set
1	Canned Soup, Seafood, Snack Foods, Pizza
2	Paper Products, Starchy Foods, Pizza, jam
3	Paper Products, Seafood, Snack Foods, Pizza
4	Starchy Foods, Paper Products, Pizza, Snack Foods
5	Canned Soup, Starchy Foods, Bread, jam
6	Canned Soup, Starchy Foods, Pizza, Bread
7	Paper Products, Starchy Foods, jam, Bread
8	Canned Soup, Paper Products, Bread, Pizza
9	Seafood, Paper Products, Snack Foods, Pizza

Above table clearly show that first transaction id contain canned soup, seafood, snack food, pizza items in an item set. Apriori algorithm for generating both association rules either positive or negative. As discuss before that Apriori algorithm generates frequent item set according to their support value and after that generate rules.After applying Apriori algorithms frequent item set is generated. Further graphical results of support and confidence values is presented.for non optimization rule which are evaluated by frequent item set after their presence and absence comparison to get both positive and negative rules. Figure 1 shows association rules and support values, figure second shows association rules and confidence values.

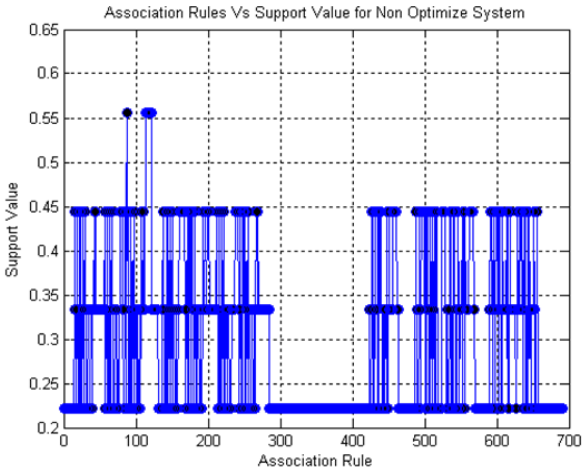

Fig 2: Support values for association rules

Figure 3, figure 4, and figure 5 shows optimize association rules with their support, confidence, and fitness values respectively. All results generated from modified approach. In figure 5 effective association rules with their desired fitness values is shown. After executing no of times approach provide more and more effective rules.

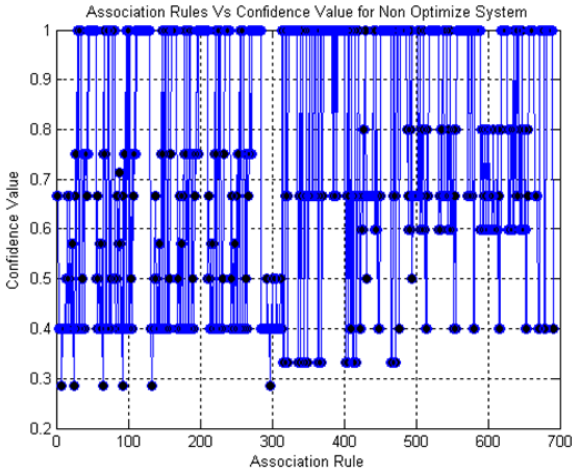

Fig 3: Confidence value for association rules

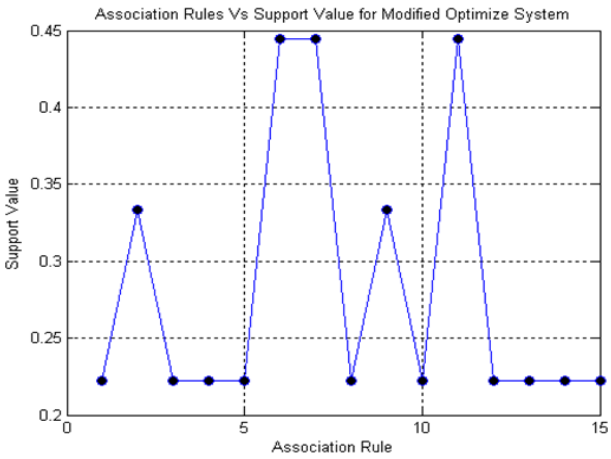

Fig 4: Optimize rules with support values.

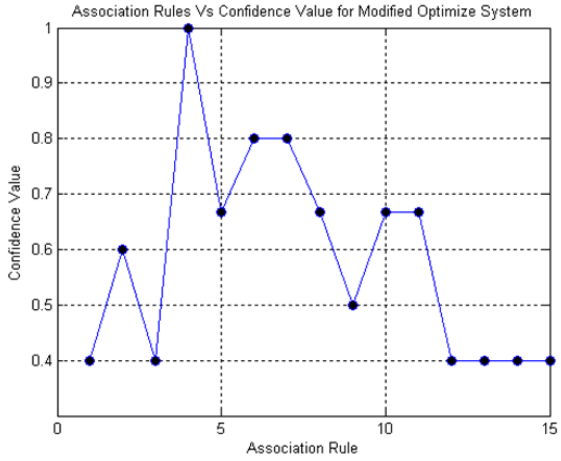

Fig 5: Optimize association rules with confidence values.

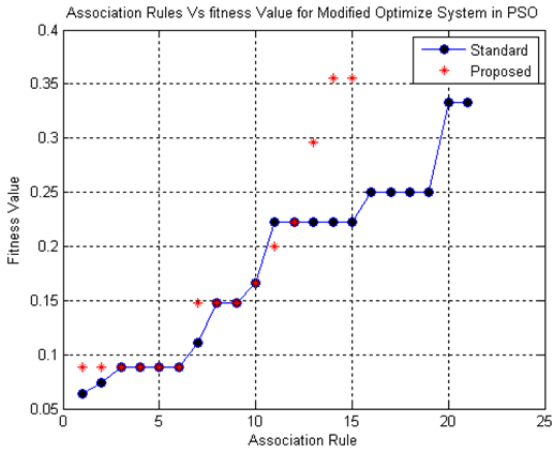

Fig 6: Optimize rules with their fitness values.

In Figure 6 the difference between simple algorithm and Modified Algorithm. Blue dot indicates fitness verses association rule for Simple algorithm and red stars indicate for modified Algorithm.

6. CONCLUSION AND FUTURE WORK

Due to proposed algorithm efficient and important negative association rules are obtained. Due to use of Apriori algorithm all kind of positive and negative items are obtained . PSO algorithm used for generation and optimization of rules also helps us to find efficient and useful rules. proposed approach helps in moving particles towards best solution rapidly due to which execution time of algorithm is improved. In future this algorithm can be applied on other large data bases in combination with another approach it will be hybrid approach which will overcome shortcomings of PSO.

7. REFERENCES

[1] R.Agrawal, T. Imielinski, and A.Swami. Mining association rules between sets of items in large databases. In the Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (ACM SIGMOD '93), Washington, USA, May 1993.

[2] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A.I. Verkamo. Finding interesting rules from large sets of discovered association rules. InProc. of the3rdInt'l Conf. on Information and Knowledge Management, Gaithersburg, Maryland, 29. Nov - 2. Dec 1994.

[3] C. Silverstein, S. Brin, R. Motwani and J.D. Ullan. Scalable techniques for mining causal structures. In the Proc. of 1998 ACM SIGMOD Int'l Conf. on Management of Data, Seattle, Washington, USA, June 1998.

[4] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalising association rules to correlations. In the Proc. of the ACM SIGMOD Int'l Conference on Management of Data (ACM SIGMOD '97).

[5] S. Brin, R. Motwani, J.D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In the Proc. of the ACM SIGMOD Int'l Conf. on Management of Data, 1997.

[6] M.L. Antonie, and O.R. Zaiane, "Mining Positive and Negative Association Rules: An Appraoch for Confined rules", Proc. Of the 8-th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD), Pisa, Italy, pp. 27-38, 2004.

[7] H. Zhu, and Zh. Xu, "An Effective Algorithm for Mining Positive and Negative Association Rules", Proc. Of the 2008 Intl. Conf. on Computer Science and Software Engineering (CSSE), Wuhan, China, pp. 455-458, 2008.

[8] S. Brin, R. Motwani, and C. Silverstein, "Beyond market basket: Generalizing association rules to correlations", Proc. of 1997 ACM SIGMOD Intl. Conf. Management of Data, ACM, Tucson, Arizona, USA, pp. 265-276, 1997.

[9] A. Savasere, E. Omiecinski. and S. Navathe, "Mining for Strong Negative Associations in a Large Database of Customer Transactions", Proc.vOf the 1998 Intl. Conf. on Data Engineering (ICDE), pp. 494-502, 1998.

[10] M. Gan, M.Y. Zhang, and Sh.W. Wang, "One Extended Form For Negative Association rules and the Corresponding Mining Algorithm", Proc. Of the 4-th Intl. Conf. on Machine Learning and Cybernetics, Guangzhou, pp. 1716-1721, 2005.

[11] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: Proceedings of the 20th International Conference on Very Large Databases, VLDB, September 1994, pp. 487–499.

[12] J. Kennedy, and R. Eberhart,"Particle Swarm Optimization,"Proceedings of the IEEE conference on neural networks – ICNN'95,vol. IV, Perth, Australia, 1995, pp. 1942-1948.

[13] Chi-Yang Tsai, I-Wei Kao "Particle swarm optimization with selective particle regeneration for data clustering". Expert Systems withApplications 2010.

[14] Ahmad Nickabadi, Mohammad Mehdi Ebadzadeh, Reza Safabakhsh "A novel particle swarm optimization algorithm with adaptive inertia weight" Applied Soft Computing 2011.