Fuzzy Particle Swarm Optimization Algorithm for a Supplier Clustering Problem

E. Mehdizadeh,

Islamic Azad University, Science and Research Branch Branch
Corresponding author, Mehdizadeh@Qazviniau.ac.ir

R. Tavakkoli Moghaddam

Department of Industrial Engineering, Faculty of Engineering, University of Tehran
Tavakoli@Ut.ac.ir

Abstract:

This paper presents a fuzzy decision-making approach to deal with a clustering supplier problem in a supply chain system. During recent years, determining suitable suppliers in the supply chain has become a key strategic consideration. However, the nature of these decisions is usually complex and unstructured. In general, many quantitative and qualitative factors, such as quality, price, and flexibility and delivery performance, must be considered to determine suitable suppliers. The aim of this study is to present a new approach using particle swarm optimization (PSO) algorithm for clustering suppliers under fuzzy environments and classifying smaller groups with similar characteristics. Our numerical analysis indicates that the proposed PSO improves the performance of the fuzzy c-means (FCM) algorithm.

Keywords: Fuzzy clustering, supplier clustering problem and particle swarm optimization

1. Introduction

Supply chain is a commonly known term in academia as well as industry. It refers to a network of integrated and dependent processes which taw matenals are transformed to finished product. In addition to managing the material flows, it involves in managing the integrated information about the product flow, from suppliers to end users, in order to improve customer satisfaction, reduce time to market, and reduce the costs related to inventories. Supply chain management (SCM) is concerned with managing this complexity. In today's highly competitive, global operating environment, it is impossible to produce low cost with high quality products successfully without satisfactory suppliers (Vokurka & et al, 1996, 106-127). Over the past several years, with the recent trend on just-in-time (JIT) manufacturing philosophy, there is an emphasis on strategic sourcing that establishes long-term mutually beneficial relationship with fewer, but better suppliers (Vokurka & et al, 1996, 106-127) and (Talluri & Narasimhan, 2004, 236–250). Due to the globalization of trade and the Internet, enlarging a purchaser's options set, supplier base management has become a challenging problem in business and academic research. One solution is to divide all suppliers into smaller sets, in which each set has a greater degree of similar characteristics. Potential benefits of working with a smaller subset of suppliers include speeding development and fulfillment operations, reducing the number of variables impacting the operation, allowing design engineers to work closely with supplier engineers, and the like (Ericson, 2003). In addition, benchmarking suppliers within the same group is more effective for supplier development processes. The cluster characteristics can also be used to gain insight about the suppliers'

performances falling in the same group. Therefore, there is a need to develop a suitable methodology to clustering suppliers. However, most data associated with suppliers are categorized, and they lack geometric properties upon which the majority of clustering techniques are employed. There are some recent successes in developing clustering algorithms that, however, disregard the uncertainty during the clustering process. The current market trends characterized by globalization, uncertainty, customer behavior, continuously changing business environment, and need for flexibility and security have increased complexities as well as interdependencies among the various entities of the supply chain.

Today, organizations focus on core competencies and outsource the non-core activities. This has increased the dependence of companies on their suppliers and increased the emphasis on supplier base management. Supplier base management practices are classified into three categories: supplier evaluation, supply base rationalization, and supplier development. Supplier evaluation includes all efforts expended by companies in evaluating their suppliers using various supplier selection models and techniques to support supplier selection decisions. Some of the popular supplier selection techniques that companies use are data envelopment analysis (Seiford, & Thrall, 1990, 7-38), analytical hierarchical process (Satty, 1980), linear weighting models, outranking (Boer et al, 1998, 109-118), expert systems ((Vokurka & et al, 1996, 106-127), and portfolio analysis (Martinez-de-Albeniz, & Simchi-Levi, 2005, 90-114). Supply base rationalization includes identification and elimination of suppliers that are not capable of meeting the company's needs. This is called supply base optimization and the outcome of this strategy is a pool of suppliers that are potentially capable of meeting

the purchasing organization's need for products and services. To be a source of competitive advantage, suppliers performance must be managed and developed to meet the needs of the buying firm (Krause et al, 1998, 39-58). Supplier development refers to any activity that a buyer undertakes to improve a supplier's performance and/or capacities to meet the buyer's short-term or long-term supply needs (Krause, & Handfield, 1999). Supplier development is a bilateral effort by both the buying and supplying organizations to jointly improve the supplier's performance and/or capabilities in one or more of the following areas: cost, quality, delivery, time-to-market, technology, environmental responsibility, managerial capability and financial variability (Krause, & Handfield, 1999).

The objective of clustering problem is to group a set of objects into a number of clusters. Different clustering algorithms have been used for this propose. These algorithms can be classified into three main categories: 1) heuristic; 2) hierarchical; and 3) partition clustering methods (Zahid et al, 1999, 1089-1097). Fuzzy clustering algorithms are partitioning methods that can be used to assign objects of the data set to their clusters. These algorithms optimize a subjective function that evaluates a given fuzzy assignment of objects to clusters. Various fuzzy clustering algorithms have been developed. For instance, (Bezdek, 1981) developed a fuzzy c-means (FCM) algorithm that is the most widely used in many applications. (Zimmermann, 1996) showed methods for the fuzzy data analysis and underlined that three types of these methods can be distinguished in modern fuzzy data analysis. In general, the first class consists of algorithmic approaches, which are fuzzified versions of classical methods, such as fuzzy clustering approaches. The second and third classes consist of knowledge-based approaches and neural net approaches. Evolutionary algorithms are increasingly combined with these approaches. (Klir & Yuan, 2003) described fuzzy clustering by the use of pattern recognition. They also introduced two basic methods for fuzzy clustering as follows: 1) a FCM clustering method based on fuzzy c-partitions; and 2) fuzzy equivalence relation based on a hierarchical clustering method. It is worthy noting that the clustering problem is an optimization problem.

Since large instances can be so difficult to optimally solve, evolutionary methods are used for clustering problems. (Bezdek & Hathaway, 1994589-594) optimized the hard c–means model with a genetic algorithm. (Klawonn & Keller,1998), 975-991) extended and applied this scheme to the FCM model. In addition, ant colony optimization (ACO) has been successfully applied to clustering problems. (Runkler, 2005, 1233-1261) introduced an ACO algorithm that explicitly minimizes the FCM cluster models. Particle swarm optimization (PSO) was first introduced by (Eberhart & Kennedy, 1995, 1942-1948) in order to optimize various continuous nonlinear functions. PSO has been applied successfully to a wide range of applications. Recently, particle swarm optimization has been applied to image clustering (Omran,

et al, 2002, 370-374), network clustering (Zhang et al, 2004, 372-375) and (Tillett et al, 2003, 73-83), clustering analysis (Chen & Ye, 2004, 789-794), and data clustering (Van der Merwe, & Engelbrecht, 2003, 215-220). In particular, Van der Merwe & Engelbrecht proposed new approaches for using PSO in clustering data; however, clustering is not a fuzzy clustering. Runkler and Katz (Runkler & Katz, 2006, 601-608) applied PSO to cluster data by considering fuzzy clustering. They introduced two new methods to minimize the two reformulated versions of the FCM objective function by PSO. (Mehdizadeh & Tavakkoli-Moghaddam, 2007, 1466-1470) proposed a particle swarm optimization (PSO) algorithm to a clustering supplier problem in a supply chain system. They showed that the proposed PSO would improve the performance of the fuzzy c-means (FCM) algorithm for the given problem.

The remaining of this paper is organized as follows: Particle swarm optimization (PSO) is overviewed in Section 2. The proposed fuzzy PSO algorithm is illustrated in Section 3. Experimental results are summarized in Section 4. Finally, conclusions are presented in Section 5.

2. Particle Swarm Optimization

As described in evolutionary techniques, PSO also uses a population of potential solutions to search the search space. However, PSO differs from other evolutionary algorithms such that there are no DNA inspired operators in order to manipulate the population. Instead, in PSO, the population dynamics resembles the movement of a "birds' flock" while searching for food, where social sharing of information takes place and individuals can gain from the discoveries and previous experience of all other companions. Thus, each companion (called particle) in the population (called swarm) is assumed to "fly" over the search space in order to find promising regions of the landscape. In the case of minimizing a function, such regions possess lower function values than other visited previously. In this context, each particle is treated as a point in a D-dimensional space, which adjusts its own "flying" according to its flying experience as well as the flying experience of other particles (companions). There are many variants of the PSO proposed in the literature so far, when Eberhart and Kennedy first introduced this technique. In our experiments, a version of this algorithm is used adding an inertia weight to the original PSO dynamics (Eberhart & Shi, 1998, PL5-PL13). This version is described bellow:

First, let us define the notations used in this paper. Particle i of the swarm is represented by the D-dimensional vector $X_i = (X_{i1}, X_{i2}, …, X_{iD})$ and the best particle of the swarm, i.e., the particle with the smallest function value, is denoted by the index g. The best previous position, representing the best function value, of particle i is recorded and represented as $p_i = (\rho_{i1}, \rho_{i2}, …, \rho_{iD})$. The position change (velocity) of particle i is $V_i = (V_{i1}, V_{i2}, …, V_{iD})$. The particles are then manipulated according to the following equations:

$$V_{id}(t+1) = \chi \{ w V_{id}(t) + c_1 \phi_1 [\rho_{id}(t) - X_{id}(t)]$$
$$+ c_2 \phi_2 [(\rho_{gd}(t) - X_{id}(t)] \} \quad (1)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (2)$$

where $d = 1, 2,…, D$; $i = 1, 2 ,…, n$; n is the size of the swarm; w is the inertia weight and its role is discussed below; c_1 and c_2 are two positive acceleration constants; ϕ_1 and ϕ_2 are two random values in range [0, 1]; and χ is a constriction factor used in constrained optimization problems in order to control the magnitude of the velocity (in unconstrained optimization problems, which this factor is usually set to 1.0). Eq. (1) computes the i-th particle's new velocity by taking into consideration three terms: (I) the particle's previous velocity; (II) the distance between the particle's best previous and its current position; and (III) the distance between the swarm's best experience (the position of the best particle in the swarm) and the ith particle's current position. In Eq. (2), particle i flies toward a new position. In general, the performance of each particle is measured according to a predefined fitness function, which is problem-dependent. The role of the inertia weight w is considered very important in the PSO convergence behavior. The inertia weight is employed to control the impact of the previous history of velocities on the current velocity, regulating the trade-off between the global and local exploration abilities of the swarm. A large inertia weight facilitates global exploration (searching new areas) while a small one tends to facilitate local exploration (i.e., fine tuning the current search area). A general rule of thumb suggests that it is better to initially set the inertia to a large value in order to make better global exploration of the search space, and gradually decrease it to get more refined solutions. Thus, a linearly decreasing inertia weight value is used. The initial population can be generated randomly.

3. Proposed Fuzzy PSO algorithm (FPSO)

Fuzzy clustering problem is an optimization problem (Klir & Yuan, 2003) and is a combinatorial optimization problem that is hard to solve, even for rather small values of c and n. In fact, the number of distinct ways to partition x into nonempty subsets is as follows:

$$|M_c| = (1/c!)[\sum_{j=1}^{c} (c!/ j!(c-j)!) \, ^{c-j} j^n] \quad (3)$$

Which for $c=10$ and $n=25$ is already roughly 10^{18} distinct 10-partitions of 25 points (Bezdek, 1981). We can PSO for solving fuzzy clustering in large scale.

In the fuzzy clustering, a single particle represents a cluster center vector. That is, each particle P_i is constructed as follows:

$$p_l = (V_1, V_2, …, V_i, …., V_c) \quad (4)$$

Where l represent the number of clusters and $l=1, 2,…, n$ and Vi is the vector of $c-th$ cluster center.

$$V_i = (V_{i1}, V_{i2}, …, V_{iD}) \quad (5)$$

$1,2,…,D$ are dimensions of cluster center vectors.
Therefore, a swarm represents a number of candidates clustering for the current data vector. Here; each point or data vector belongs to every various cluster by different membership function, thus; assign a fuzzy membership to each point or data vector. Each cluster has a cluster center. And per iteration present a solution that gives a vector of cluster centers. We determine the position of vector P_i for every particle and update it, then change the position of cluster centers based of particles.
For the purpose of this algorithm, define the following notations:

n : The number of data vector
C : The number of cluster center
$V_l^{(t)}$: The position of the l-th particle in stage t
$Vel_l^{(t)}$: The velocity of the l-th particle in stage t
X_k : The vector of data and $k=1,2,…,n$
$\rho_l^{(t)}$: The best position funded by lth particle in stage t
$\rho_g^{(t)}$: The best position funded by all particles in stage t
$p^{(t)}$: The fuzzy pseudo partition in stage t
$A_i^{(t)}(X_k)$: The membership function of the k-th data vector in stage t into the i-th cluster

The fitness of particles is easily measured as the following:

$$J(p) = \sum_{k=1}^{n} \sum_{i=1}^{c} [A_i(X_k)]^m \| x_k - V_i \|^2 \quad ;$$
$$l = 1,2,…,n\_part \quad (6)$$

The fitness of particles is easily measured by Eq. (6). The c-means algorithm tends to converge faster than the proposed FPSO algorithm with a less accurate clustering. In this section, the performance of the PSO clustering algorithm is improved by seeding the initial swarm with the result of the c-means algorithm. The FPSO algorithm first

executes the *c*-means algorithm once. In this case, the *c*-means clustering algorithm can be terminated by one of two stopping criteria: (1) The maximum number of iterations; or (2) $\left|p^{(t+1)} - p^{(t)}\right| \le \varepsilon$. The result of *c*-means algorithm is then used as one of the particles, while the rest of the swarms are initialized randomly. The following algorithm can use to find cluster for each data vector, as follows:

Step 1) Let *t*=0, select initial parameters, such as number of cluster center *c*, initial position of particle by the FCM, initial velocity of particles, c_1, c_2, w, χ, and a real number $m \in (1, \infty)$, and a small positive number ε for stopping criterion.

Step 2) Calculate $A_i^{(t)}(X_k)$ for all particles (*i*=1, 2, …, *c* and *k*=1, 2, …, *n*) by Equation (7) and update $p^{(t+1)}$.

$$A_i^{(t+1)}(X_k) = \left[\sum_{j=1}^{c} \left(\frac{\left\| X_k - V_i^{(t)} \right\|^2}{\left\| X_k - V_j^{(t)} \right\|^2} \right)^{\frac{1}{m-1}} \right]^{-1} \tag{7}$$

Step 3) for each particle, calculate the fitness by the use of Eq. (6)

Step 4) Update the global and local best position

Step 5) Update $Vel_l^{(t)}$ and $V_l^{(t)}$ (*l*=1, 2, …, *n_particle*) as given by Eqs. (8) and (9).

$$Vel_{id}(t+1) = \chi\{wVel_{id}(t) + c_1\phi_1[\rho_{id}(t) \tag{8}$$
$$- V_{id}(t)] + c_2\phi_2[(\rho_{gd}(t) - V_{id}(t)]\}$$

$$V_{id}(t+1) = V_{id}(t) + Vel_{id}(t+1) \tag{9}$$

Step 6) Update $p^{(t+1)}$ by Step 2

Step 7) Compare $p^{(t)}$ and $p^{(t+1)}$ If $\left|p^{(t+1)} - p^{(t)}\right| \le \varepsilon$, then stop;

otherwise, increase *t* by one and continue form Step 3.

In the above algorithm, parameter *m*>1 is selected according to the given problem. The partition becomes fuzzier with increasing *m* and there is currently no theoretical basis for an optimal choice for the value of *m* (Klir & Yuan, 2003).

4. Experimental Results

We compare the results of the FCM and FPSO algorithms on various problems from the literature. The performances are measured by objective function value by Eq. (6) and CPU time. A general rule of thumb is that a clustering result with lower *J (p)* and lower CPU time is preferable. For a comparable assessment, we code these methods by

using the fuzzy tools available in MATLAB 7 and the FPSO respectively, 10 particles, with *w*=0.72, $c_1 = c_2 = 1.49$. For our experimental tests, we use a PC Pentium III, CPU 1133 MHz and 256 MB of RAM for the same parameters for all algorithms implementations: *m*=2, the maximum number of iterations is 100 and ε = 0.00001.

Example 1 (Bezdek, 1981): Let us consider a data set, *x*, consisting of 15 points in $\mathfrak{R}^2$ as given in Table 1. Assume that we want to determine a fuzzy pseudo partition with two clusters (i.e., *c*=2).

Table 1
Data set *x* of 15 points

k	1	2	3	4	5	6	7	8
x_{k1}	0	0	0	1	1	1	2	3
x_{k2}	0	2	4	1	2	3	2	2

9	10	11	12	13	14	15
4	5	5	5	6	6	6
2	1	2	3	0	2	4

Example 2(Klir & Yuan, 2003): Let us consider a small data set, *x*, consisting of five points in $\mathfrak{R}^2$, as given in Table 2. We employ the algorithms proposed in this paper to this set of data with *c*=2 and *c*=3.

Table 2
Data set *x* of five points

k	1	2	3	4	5
x_{k1}	0	1	2	3	4
x_{k2}	0	1	3	1	0

Example 3 [Artificial problem]: Let us consider a data set, *x*, generated at random consisting of five points in $\mathfrak{R}^2$, as given in Table 3. We apply the proposed algorithms to this set of data with *c*=3.

Table 3
Data set *x* of five points

	1	2	3	4	5
x_{k1}	0	0	0	1	1
x_{k2}	0	2	4	1	2

Example 4[Artificial problem]: A total of 100 data vectors are generated at random with U~ (0, 20), and apply the algorithms to this set of data with *c*=2.

Example 5 (Newman et al, 1998): The Zoo data set is obtained from the UCI Machine Learning Repository [27]. This Zoo data set has 101 data points, whose point represents information of an animal in terms of 18 categorical attributes. Each animal data point is classified to 7 classes.

Example 6(Newman et al, 1998): The Iris plants data set is obtained from the UCI Machine Learning Repository. This is perhaps the best known database to be found in the

pattern recognition literature. The data set has 150 data points containing three classes of 50 instances each, in which each class refers to a type of Iris plant.

Example 7(Newman et al, 1998): The Wine data set is obtained from the UCI Machine Learning Repository. This data set has 178 data points, 13 attributes. Each data point is classified to three classes.

Example 8 (Newman et al, 1998): The training Image Segmentation data set is obtained from the UCI Machine Learning Repository. This data set has 210 data points, whose point represents information of an image in terms of 19 categorical attributes. Each data point is classified to seven classes (i.e., brick face, sky, foliage, cement, window, path, grass).

4.1. Effects of parameters

We present the impact of parameters on solution quality by using the FPSO. The values given in the table report the CPU time for Example 5 taken from Newman et al,. As mentioned before, a combination of c_1=1.49, c_2=1.49 and w=0.72 are the best combination in terms of the solution quality.

4.2. Comparison of methods

Table 4 shows the comparison of two clustering algorithms (i.e., the FCM and FPSO algorithms) for eight different examples, in which there are 764 data points for the experiments.

Table 5
Discritized data.

Obj	QMP	SA	PMC	MGT	DD	C	Q	P	D	CRP	Other	Effi
1	3	2	3	3	4	2	1	1	1	1	1	I
2	2	2	1	1	1	2	2	1	1	1	1	E
3	1	1	1	1	2	1	3	1	4	1	3	E
4	5	2	2	2	3	4	5	2	4	1	4	E
5	5	2	3	3	4	4	3	3	3	1	2	I
6	3	2	2	3	3	3	3	3	4	2	3	E
7	2	1	3	2	4	3	4	2	2	2	3	E
8	5	2	3	3	3	3	2	2	2	1	2	I
9	5	2	3	3	4	4	1	1	1	1	1	I
10	1	2	1	1	1	1	5	1	4	1	3	E
11	2	1	1	2	3	2	1	1	3	1	2	I
12	1	1	3	3	3	3	3	3	2	1	2	E
13	5	2	3	3	4	4	1	2	3	1	3	I
14	4	2	3	3	4	4	2	1	2	1	2	I
15	4	2	2	3	1	1	4	3	4	2	3	E
16	3	2	3	3	2	2	3	1	1	1	1	I
17	5	2	3	3	4	3	3	1	2	1	1	I
18	5	2	3	3	4	4	4	1	1	1	1	I
19	4	2	3	1	4	3	4	1	4	1	3	I
20	4	2	3	3	1	4	2	2	4	1	4	E
21	5	2	3	2	4	4	2	1	3	1	2	I
22	5	2	2	3	3	4	5	3	4	2	4	E
23	4	2	3	3	2	3	4	3	3	2	4	E

The values reported are: the average, best, and worst OFVs as well as their CPU time over 10 simulations. Our tests show that the FPSO computation times (i.e., CPU time) for the all examples are significantly lower than the FCM method with high solution quality in terms of the OFV. When the problem becomes large, different between two algorithms be larger. It is worthy noting that the pure PSO algorithm works in high speed; however, finding the good solution quality is poor. In contrast, the FCM has low speed with high solution quality, especially for large-sized problems. In this paper, we propose the FPSO algorithm holding both advantages of high speed (i.e., low CPU time) and high solution quality (i.e., OFV). The associated results show a general improvement of the performance when the PSO is seeded with the outcome of the FCM algorithm. The results indicate that:

- The FPSO algorithm has better performance than the FCM algorithm in terms of CPU time and the solution quality for large sizes.

- There are a few differences between two algorithms, when the size of problems is small.

Fig. 1 summarizes the effect of varying the number of clusters for the different algorithms for Example 4. It is expected that the OFV should go down when the number of clusters increases. This figure also shows that the FPSO algorithm consistently performs better than another approach when the number of clusters increases.

We also apply the hybrid fuzzy clustering PSO, namely FCPSO, to clustering suppliers for supplier base management. A supplier data set adopted from (Talluri & Narasimhan, 2004, 236–250) is used to validate this proposed algorithm in the domain of supplier base management. This given data set contains supplier capability and supplier performance information on 23 suppliers with identifying ten attributes (shown in Table 5) including quality management practices and systems (QMP), documentation and self-audit (SA), process/manufacturing capability (PMC), management of firm (MGT), design and development capabilities

(DD), cost (C), quality (Q), price (P), delivery (D), cost reduction performance (CRP) and others. (Talluri & Narasimhan, 2004, 236–250) applied data envelopment analysis (DEA) to determine the efficiency of each supplier. Their conclusion on each supplier is shown in the last column of Table 5. All suppliers with efficiency (shown as Effi column)

equal to one are considered efficient and all suppliers with efficiency less than one are considered inefficient. Some categorical data collected by (Talluri & Narasimhan, 2004, 236–250) were normalized and calculated based on weighted average.

They first discritize the data set and apply max-max-roughness (MMR) to cluster suppliers. The results are summarized in Table 5. In Table 6, we summarize the results of the FCPSO with three clusters. Some industry best practices on supplier performance classify the suppliers into golden, silver and bronze.

Clearly, FCPSO has some ability to separate suppliers which are efficient from those that are inefficient.
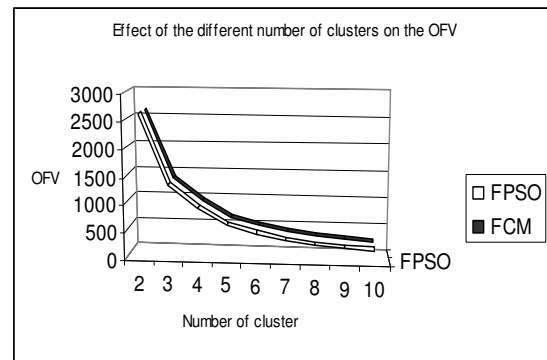

Figure 1. Effect of the different number of clusters on the OFV

Table 4
comparison of FCM and FPSO clustering algorithms

Problem	Method	Average OFV	Average CPU time
Example 1 with c =2	FCM	26.328158	.31727
	FPSO	26.328158	.12454
Example 2 with c =2	FCM	5.566497	.62192
	FPSO	5.566492	.12654
Example 2 with c =3	FCM	1.758945	.44425
	FPSO	1.758944	.12451
Example 3 with c =2	FCM	3.256326	.79469
	FPSO	3.256323	.50545
Example 4 with c =2	FCM	2633.734765	1.13997
	FPSO	2633.734765	.55227
Example 5 with c =7	FCM	89.15341	1.345331
	FPSO	88.268703	.82019
Example 6- IRIS with c =3	FCM	60.575960	.80155
	FPSO	60.575956	.65138
Example 7- WINE with c =3	FCM	1796125.937087	2.28148
	FPSO	1796125.937076	.78182
Example 8 – Image with c =7	FCM	678309.4305	3.94473
	FPSO	678163.185	1.20124

Table 6
Results obtained by the proposed FVPSO.

Cluster Number	Cluster Number by FCPSO	Efficiency
1	4	E
1	15	E
1	19	I
1	20	E
1	22	E
1	23	E
2	1	I
2	2	E
2	3	E
2	6	E
2	7	E
2	10	E
2	11	I
2	12	E
2	16	I
3	5	I
3	8	I
3	9	I
3	13	I
3	14	I
3	17	I
3	18	I
3	21	I

Thus, in this study, we summarize the results of FCPSO with the number of clusters set to be three in Table 6. Thus, the purity of Clusters 1, 2, and 3 is 83% (5/6), 67 %(6/9), and 12.5% (1/8), respectively. The majority of suppliers in Clusters 1 and 2 are efficient whereas majority of suppliers in Cluster 3 are inefficient.

5. Conclusion

Managing a massive supplier base is a challenging task. There is a need to cluster suppliers into manageable smaller subsets. However, little supplier clustering work has been done, especially, when categorical data are involved and uncertainty of belonging to a cluster is

considered. Fuzzy clustering has been adapted successfully to solve this problem, which is a hard combinatorial problem. However, when the problem becomes larger, fuzzy clustering algorithms may result uneven distribution of suppliers. This paper proposes a hybrid algorithm, namely FPSO, combining the fuzzy *c-means* (FCM) and the particle swarm optimization (PSO) algorithms in order to cluster suppliers in fuzzy environments. We have examined this algorithm on a number of data sets. We have also utilized real industry data from literature to test this proposed algorithm. Our experimental tests showed that the FPSO computation times (CPU time) for the all examples were significantly lower than the FCM method with high solution quality in terms of the objective function value (OFV).

6. References

[1] R.J. Vokurka, J. Choobineh, and L. Vadi, *"A prototype expert system for the evaluation and selection of potential suppliers"*, International Journal of Operations and Production Management. 12 (1996), pp. 106-127.

[2] S. Talluri, and R. Narasimhan, *"A methodology for strategic sourcing"*, European Journal of Operational Research. 154 (2004), pp. 236–250.

[3] J. Ericson, *"Supply base best practices"*, E-Business News (2003) - available at http://www.line56.com/default.asp?NewsID=4473.

[4] L. Seiford, and R.Thrall, *"Recent developments in DEA: the mathematical programming approach to frontier analysis"*, Journal of Econometrics. 46 (1990), pp. 7-38.
T. Satty, *"The analytical hierarchical process"*, McGraw Hill, 1980.

[5] L. Boer, L. Wegen, and J. Telgen, *"Outranking methods on support of supplier selection"*, European Journal of Purchasing & Supply Management. 4 (1998), pp.109-118.

[6] V. Martinez-de-Albeniz, and D. Simchi-Levi, *"A portfolio approach to procurement contracts"*, Production and Operations Management. 14 (2005), pp. 90-114.

[7] D. Krause, R. Handfield, and T. Scannel, *"An empirical investigation of supplier development: reactive and strategic processes"*, Journal of Operations Management. 17 (1998), pp. 39-58.

[8] D. Krause, and R. Handfield, *"Developing a world-class supply base"*, Center for Advanced Purchasing Studies available at http://www.capsresearch.org/publications/pdfspublic/krause1999es.pdf

[9] N. Zahid, M. Limoun, and A. Essaid, *"A new cluster-validity for fuzzy clustering"*, Pattern Recognition, 32(1999), pp. 1089-1097.

[10] J.C. Bezdek, "*Pattern recognition with fuzzy objective function algorithms*", New York, 1981.

[11] H.J. Zimmermann, "*Fuzzy set theory and its applications"*, Lower Academic Publishers, 1996.

[12] G. J. klir, and B. Yuan *"Fuzzy sets and Fuzzy logic, theory and Applications*"**,** prentice – Hall of India , 2003.

[13] J.C. Bezdek, and R.J. Hathaway, *"Optimization of fuzzy clustering criteria using genetic algorithms"*, Proc. of the IEEE Conf. on Evolutionary Computation, Orlando, Vol. 2, pp. 589-594, 1994.

[14] F. Klawonn, and A. Keller, *"Fuzzy clustering with evolutionary algorithms"*, International Journal of Intelligent Systems, 13(1998), pp. 975-991.

[15] T.A. Runkler, *"Ant colony optimization of clustering models"*, Int. J. of Intelligent Systems, 20 (2005), pp. 1233-1261.

[16] T.A. Runkler, and C.Katz, *"Fuzzy clustering by particle swarm optimization"*, IEEE Int. Conf. on Fuzzy Systems, Vancouver: Canada, pp. 601-608, 2006.

[17] J. Kennedy, and R.C. Eberhart, "*Particle swarm optimization*", Proceedings of the IEEE International Joint Conference on Neural Networks, Vol. 4, pp 1942-1948, 1995.

[18] M. Omran, , A. Salman, and A.P. Engelbrecht, *"Image Classification using Particle Swarm Optimization"*, Proc. of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, Singapore, pp. 370-374, 2002.

[19] C.J. Zhang, Y. Gao, S.P. Yuan, , and Z. Li, *"Particle swarm optimization for mobile and hoc networks clustering"*, Proceeding of the Int. Conf. on Networking, Sensing and Control, Taipei: Taiwan, pp. 372-375, 2004.

[20] J. Tillett, R. Rao, F. Sahin, and T. M. Rao, *"Particle swarms optimization for the clustering of wireless sensors"*, Proc. of SPIE: Digital Wireless Communications V, Vol. 5100, pp. 73-83, 2003.

[21] C.Y. Chen, and F. Ye, "*Particle swarm optimization and its application to clustering analysis*", Proc. of the Int. Conf. on Networking, Sensing and Control, pp. 789-794, 2004.

[22] D.W. Van der Merwe, and A.P. Engelbrecht, "*Data clustering using particle swarm optimization"* Pro. of the IEEE Congress on Evolutionary Computation, Canberra: Australia, pp. 215-220, 2003.

[23] E. Mehdizadeh, and R. Tavakkoli-Moghaddam, "*A hybrid fuzzy clustering PSO algorithm for a clustering supplier problem*", Proc. of the IEEE Int. Conf. on Industrial Engineering and Engineering Management (IEEM 2007), M. Helander, M. Xie, R. Jiao and K.C. Tan (Eds.), Singapore, 2-5 December, pp. 1466-1470, 2007.

[24] R.C. Eberhart, and Y.H. Shi, *"Evolving artificial neural networks"*, Proc. of the. Int. Conf. Neural Networks and Brain, China: Beijing, pp. PL5-PL13, 1998.

[25] D.J. Newman, S. Hettich, , C.L. Blake, and C.J. Merz, "*UCI Repository of machine learning databases"*, Department of Information and Computer Science, University of California, Irvine,CA, http://wwwics.uci.edu/~mlearn/MLRepository.html,1998.