ISSN: 0067-2904

Finding Best Clustering For Big Networks with Minimum Objective Function by Using Probabilistic Tabu Search

Ali Falah Yaqoob*, Basad Al-Sarray
Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq

Abstract

Fuzzy C-means (FCM) is a clustering method used for collecting similar data elements within the group according to specific measurements. Tabu is a heuristic algorithm. In this paper, Probabilistic Tabu Search for FCM implemented to find a global clustering based on the minimum value of the Fuzzy objective function. The experiments designed for different networks, and cluster's number the results show the best performance based on the comparison that is done between the values of the objective function in the case of using standard FCM and Tabu-FCM, for the average of ten runs.

Keywords: Fuzzy; C-Means, Tabu, Clustering, Network, Facebook.

ايجاد افضل تعنقد للشبكات الكبيرة باستخدام طريقة تابو الاحتمالية البحثية

علي فلاح يعقوب*، بسعاد علي السراي
قسم علوم الحاسبات، كلية العلوم، جامعة بغداد، بغداد، العراق

الخلاصة

هي طريقة تجميع شائعة تستخدم لجمع عناصر البيانات المتشابهة داخل مجموعة Fuzzy C–mean(FCM)ال هي لها عدة حلول محلية الصغرى. خوارزمية التابو هي خوارزمية ارشادية. FCMوفقاً لبعض القياسات. مشكلة ال في هذا البحث ، نحاول استخدام خوارزميه التابو لايجاد الحلول الكليه المثلى الغير محلية التي تستند الى القيمه الدنيا لدالة الهدف (ايجاد افضل تعنقد باقل قيمه لداله الهدف)، تم تصميم عدد من التجارب وتنفيذها على شبكات حقيقيه مختلفه وتمت المقارنه بين الطريقه التقليديه لحساب دالة الهدف والطريقه المقدمة بالاعتماد على خوارزميه التابو الاحتمالية.

1. Introduction

Networks appear in different topics, for example, social media, electrical power networks, communication networks, Politic, biology, etc. In general, the strictures of the networks are finding by applying the mathematical techniques to give a description of the suitable patterns. Clustering is an unsupervised important technique used to search for structures in data. Clustering methods used to partition a set of elements into clusters such that objects in the same cluster are more similar to each other than objects in different clusters according to some defined criteria. Fuzzy c-means (FCM) is a method of clustering based on the minimization of the objective function. Al-Sultan, and Chawki, [1], studied the problem of possession of many local minima of Fuzzy clustering mathematical program, they proposed fuzzy C-means heuristic approach to this problem based on Tabu search technique. Ng et al, [2] presented a Tabu search based clustering algorithm, to extend the k-means paradigm to categorical domains, and domains with both numeric and categorical values. This technique gives the solution space beyond local optimality by finding the global solution of the fuzzy clustering problem. Zhang et al [3], compared three techniques that implemented to extend fuzzy c-means (FCM)

*Email: ali.f.yaqoob94@gmail.com

clustering to very large data, where both loadable and very large datasets to conduct the numerical experiments that facilitate comparisons based on time and space complexity, speed, quality of approximations. Zhu et. al, [4] generalized an algorithm called GIFP-FCM to get an effective clustering, they introduced a membership constraint function based on norm distance measure and competitive learning, they showed the robustness and convergence of their proposed algorithm. Shang et. al, [5] proposed a self-adaptive method to determine the optimal number of clusters, the algorithm designed to automatically determined the possible maximum number of clusters instead of using the empirical rule and obtained the optimal initial cluster centroids. Kochenberger et al, [6] presented the problem of max-cut problems via Tabu search, they applied TS algorithm on large scale Max-cut test problems, and unconstrained quadratic binary program. The rest of the paper, in section.3 we give the mean idea of Fuzzy C-means clustering via minimizing the objective function here we call it Fuzzy objective function, section.4 we explain the algorithm of Tabu Search for FCM, Section.5 presented the results and discussions.

2. Clustering via Fuzzy C-means

Fuzzy c-means (FCM) is a data clustering technique wherein each data point belongs to a cluster to some degree that is specified measures. This technique was originally introduced by Bezdek in 1981. FCM algorithm attempts to partition a finite collection of elements into a collection of c fuzzy clusters with respect to some given criterion. Fuzzy c-means aims to minimize the objective function total weighted mean-square error, v_{ij} the degree to in case an observation σ_i belongs to a partition ρ_j, c_j is the center of the cluster j, [7],[8]

$$\Upsilon = \sum_{j=1}^{k} \sum_{\sigma_i \in \Sigma_j} |\sigma_i - c_j|^2 . \nu_{ij}^m \quad (1)$$

Equation (1) is similar to fuzzy c-means algorithm, where the parameter c is the fuzziness of the clustering. The centroid of each cluster

$$\nu_{ij}^m = \frac{1}{\sum_{l=1}^{k} \left(\frac{|\sigma_i - \rho_j|}{|\sigma_i - \rho_k|} \right)^{\frac{2}{m-1}}} \quad (2)$$

For Fuzzy clustering the centroid is the mean of all points, weighted by their degree of belonging to the cluster:

$$\rho_j = \frac{\sum_{\sigma \in C\Sigma_j} \nu_{ij}^m \sigma}{\sum_{\sigma \in \Sigma_j} \nu_{ij}^m} \quad (3)$$

ρ_j is the centroid of the cluster j,ν_{ij} is the degree to which an observation σ_i belongs to a cluster ρ_j

Algorithm .1 Fuzzy clustering procedure

1. *Determine K number of clusters*
2. *Assign randomly to each point coefficients for being in the clusters.*
3. *Repeat until the maximum No.Iter is reached, or when the getting the condition of convergences*
   - *4. Find centroid for each cluster, using Equation(3)*
   - *5. Find the coefficients of each point of being in the clusters, using Equation(2).*

The algorithm minimizes intra-cluster variance as well, but has the same problems as k-means; the minimum is a local minimum, and the results depend on the initial choice of weights. Hence, different initializations may lead to different results.

3.1. Tabu Search

Tabu Search is a Global Optimization algorithm and a Metaheuristic or Meta-strategy for controlling an embedded heuristic technique. The basic concept of Tabu search as described by Glover (1986), he presented it as a meta-heuristic superimposed on another heuristic. The overall approach is to avoid entrainment in cycles by forbidding or penalizing moves which take the solution, in the next iteration, to points in the solution space previously visited (hence "Tabu") [9]. The idea of Tabu method is a simulation to the human behavior appears to operate with a random element that leads to inconsistent behavior given similar circumstances.

Tabu method estimates the resulting tendency to deviate from a path, might be regretted as a source of error but can also prove to be a source of gain with the exception that new path is not chosen

randomly. Instead, the Tabu search proceeds according to the supposition that there is no point in accepting a new (weak) solution unless it is to avoid a path already investigated.
The new regions of the search space avoiding local minima and ultimately finding the optimal solution. The Tabu search begins its searching to a local minimum with avoiding of retracing the steps used; the recent moves are keeping in one or more Tabu lists. This list will not prevent a previous move from being repeated, but rather to ensure it will not reverse. Tabu lists used for recorded the history and build the Tabu search memory. The role of the memory can change as the algorithm proceeds, for more details see [9, 10, 11]. The differences between the implementations of the Tabu method are done with the size, variability, and adaptability of the Tabu memory to a problem search space.

Algorithm 2. Tabu search for FCM

Input: Network, No.of Partition , Tabu.list length
Output: Optimal solution Sol.opt, Best Cost
Step.1 Find centers of fuzzy c-mean.
Step.2 Generate Initial search value
Step.3 while $|F_{best} - F_{LastBest} > E_{(F(X))}$ and $Iters \leq Max.Ite$
Step.4 Starting Tabu Search for minimam Fuzzy obj.fun
Step.5 Generate Initial Solution ($S.Init$) , Tabu.List = []
Step.6 While (Stop Condition doesn't get)
Step.7 Get neighbors (best candidate)
Candidate solution List =[]
Step.8 For ($Sol.best$ in $Sol.Optim$ region (neighborhood))
Step.9 If (there isn't any features in (Sol.new, TabuList))
Step.11 Define $Sol.new$ constraint $Sol.new$ list

$$Sol.new \leftarrow Locate\ Best\ Candidate\ (Candidate\ List)$$

Step.12 $If\ (Cost(S.new) <= Cost(Sol.Optim)) \rightarrow Sol.Optim = Sol.new$
Step.13 $Tabu.List\ \ Feature\ Differences(Sol.new, Sol.opt)$
While $(Tabu\ List\ > Tabu\ List\ Size) \rightarrow Delete\ Feature\ (Tabu\ List)$
Step.15 Return ($Sol.Opt$)
Step.16 Iters = Iters + 1 ;

3.2. Probabilistic Tabu search

Tabu probability version used for reducing the dependence on memory. The probabilities governing the acceptance of moves from a specified candidate set derive from three sources, move attractiveness, related to changes induced in c(x), Tabu status, related to tenure on a Tabu list, and aspiration level, related to the value of c(x) achieved in relation to a historical standard.
Let $X(\sigma)$ be a neighborhood of the point σ and assume that it contains all neighboring points $o \in \Omega^n$ with Hamming distance $d(\sigma, o) \leq 2$. A neighborhood $X_p(\sigma) \subseteq X(\sigma)$which is collected randomly based on probabilistic threshold $p \in (0,1)$. For each $o \in X(\sigma), o \in X_p(\sigma)$ randomly with probability p and independently from other points. $X_p(x)$ may be transformed from empty for arbitrary threshold p to including all points from $X(\sigma)$. For a finite sequence $\{\sigma_\tau\}, 1 \leq \tau \leq k$ and $\sigma_{\tau+1} \in X(\sigma_\tau)$, ordered set $Tabu.List = \{(\sigma_\tau, y_\tau), (\sigma_{\tau-1}, y_{k-1}), \dots, (\sigma_{\tau-l+}, y_{\tau-l+1})\}$ is called a Tabu list if vectors σ_τ and $\sigma_{\tau+1}$ differ by coordinates (σ_τ, y_τ). The constant l is called the length of the Tabu list. Note that $x_\tau, and\ y_\tau$ are defined as equal if the vectors σ_τ and $\sigma_{\tau+1}$ are differed by exactly one coordinate. By definition, $\sigma_\tau, y_\tau = 0$, if $\sigma_{\tau+1} = \sigma_\tau$. Let $X(\sigma_\tau, T.List)$ be a set of points $y \in X_p(\sigma_t)$, which are not forbidden by the Tabu list. $X_p(\sigma_\tau, T.List)$ may be empty for nonempty set $X_p(\sigma_\tau)$.

Algorithm.3 Probabilistic Tabu Search scheme

1. Initialize $\sigma_0 \in \Omega^n, \Upsilon^* := \Upsilon(\sigma_0), T.List := [], \tau := 0.$

2. While a stopping criteria not get do
2.1. Generate neighborhood $X_p(\sigma_\tau, T.List)$.
2.2. If $X(\sigma_\tau, T.List) = [\,]$, then $\sigma_{\tau+1} := \sigma_t$,
Else find $\sigma_{\tau+1}$ such that $\Upsilon(\sigma_{\tau+1}) = min\{\Upsilon(o),\ o \in X(\sigma_\tau, T.List)\}$.
2.3. If $\Upsilon(\sigma_{\tau+1}) < \Upsilon *$ then $\Upsilon^* := \Upsilon(\sigma_{\tau+1})$.
2.4. Update the Tabu list T. List and the counter $\tau := \tau + 1$.
If $l > |X(\sigma)|$, then all points may be forbidden and $X_p(\sigma, T.List) = []$.

4.1. Setting of the experiments

This section deals with experimental part of this paper, the results show the ability of the proposed algorithm to find optimal solution, the best clusters, based on the values of the Fuzzy objective function. The experiments designed for the real network with different topics and complicity, the details of the networks given in Table-1.

Table 1-Details of the networks that used in this work

Networks	No. of Nodes	No. of Edges
Zackary Karate	34	78
Dolphin	62	158
American Football Collage (AFC)	115	613
Facebook	3958	84241
Protein	2284	6644
Political blogs	1107	9537
Internet Level AS Network(ILAN)	6444	11284
Chesapeake	39	170
Delaunay	1024	3056
Twitter	2623	21000

The experiments are designed to find the clusters for different types of large networks see details in Table-1, [12].

Here, the maximum number of iterations are 10000, minimum amount of improvement 1e-20. The process of computing clustering stopped when the maximum number of iterations is reached, or when the objective function improvement between two iterations is less than the minimum amount of specific tolerance.

The comparison for the values of the Fuzzy objective function is given in Table-2., in this table the values of the objective function that computed for the case of using standard FCM, and in the case of using Tabu method to compute the objective function. Different setting is adopted to implement the experiments, the experiments designed for the case of the known number of clusters by assuming K=2, the second implementation when the number of clustersis auto selected, results are given in Table 2 and Table-3. Give the average best values of the objective function for 10 runs with number of clusters=2 or auto select. The affected parameters are *P* probability threshold, the values of P are on the range (0, 1), the small value of P gives the minimum Fuzzy objective function, the results are given in Figure-(1-11).

Table 2- Values of the obj.fun computing by FCM and Tabu Fuzzy-FCM

Data-Networks	F.obj Fun with No. of Clust = 2	Best J-FCM by TS-FCM in No. of Clus = 2	F.obj Fun in Auto No. of Clust	Best objFun by TS-FCM
Zackary Karate	$1792 \times 10^{(4)}$	**$3800 \times 10^{(4)}$**	$3079 \times 10^{(5)}$	**$9552 \times 10^{(-24)}$**
Dolphin	$1232 \times 10^{(5)}$	**$2387 \times 10^{(5)}$**	$9411 \times 10^{(6)}$	**$5039 \times 10^{(-24)}$**
AF C	$1856 \times 10^{(2)}$	**$3541 \times 10^{(4)}$**	$2491 \times 10^{(6)}$	**$30945 \times 10^{(-22)}$**
Facebook	$1723 \times 10^{(4)}$	**$3672 \times 10^{(10)}$**	$1503 \times 10^{(11)}$	**0.24470**
Protein	$8329 \times 10^{(5)}$	**$1613 \times 10^{(10)}$**	$6164 \times 10^{(8)}$	**33.0192**
Political	$3920 \times 10^{(8)}$	**$6294 \times 10^{(9)}$**	$4701 \times 10^{(8)}$	**60.2409**
Internet	$1021 \times 10^{(9)}$	**$2070 \times 10^{(11)}$**	$6730 \times 10^{(9)}$	**528.5416**
Chesapeake	$1131 \times 10^{(7)}$	**$1131 \times 10^{(4)}$**	$1951 \times 10^{(5)}$	**$30938 \times 10^{(-19)}$**
Delaunay	$1539 \times 10^{(10)}$	**$1539 \times 10^{(9)}$**	$1061 \times 10^{(9)}$	**31.6721**
Twitter	$9875 \times 10^{(11)}$	**$9653 \times 10^{(11)}$**	$3505 \times 10^{(11)}$	**$3504 \times 10^{(-12)}$**

Table 3-Comparison of the average of 10 runs among the values of F.obj.Fun computing by FCM and TS-FCM

Data-Networks	Average obj.fun, 10 run, clus =2	Average obj.fun auto selecting clusters
Zackary Karate	$2.3870 \times 10^{(4)}$	710.1709
Dolphin	$3.8006 \times 10^{(3)}$	114.4181
American Football College	$3.5413 \times 10^{(5)}$	$4.3935 \times 10^{(4)}$
Facebook	$3.6726 \times 10^{(10)}$	$1.9695 \times 10^{(8)}$
Protein	$1.6131 \times 10^{(9)}$	$1.8369 \times 10^{(7)}$
Political	$6.2950 \times 10^{(8)}$	$9.6541 \times 10^{(6)}$
Internet	$2.0704 \times 10^{(10)}$	$6.6961 \times 10^{(8)}$
Chesapeake	$1.1315 \times 10^{(4)}$	410.4503
Delaunay	$1.5391 \times 10^{(8)}$	$8.5734 \times 10^{(5)}$
Twitter	$9865 \times 10^{(6)}$	$3510 \times 10^{(4)}$

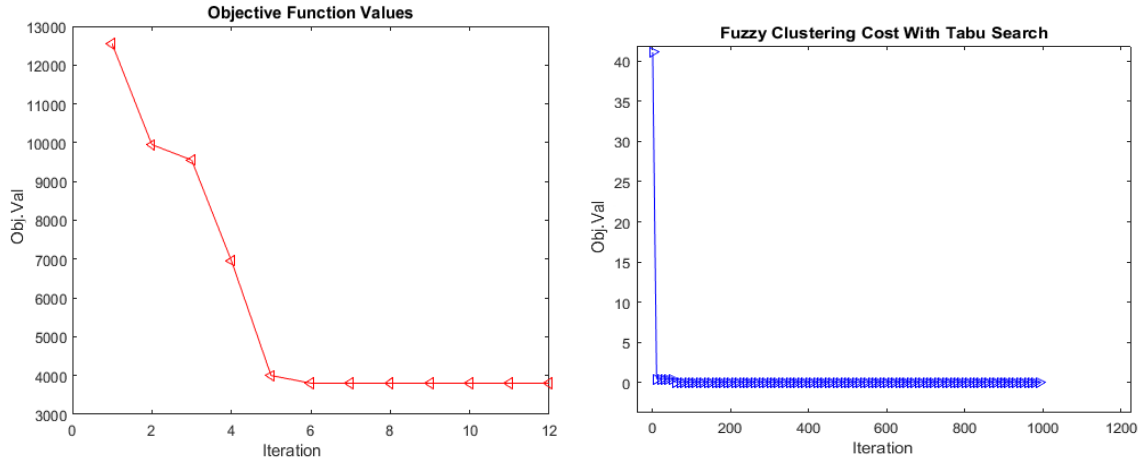

Figure 1-Values of the best Fuzzy obj.fun (red- FCM classic via blue -TS-FCM) for Zachary

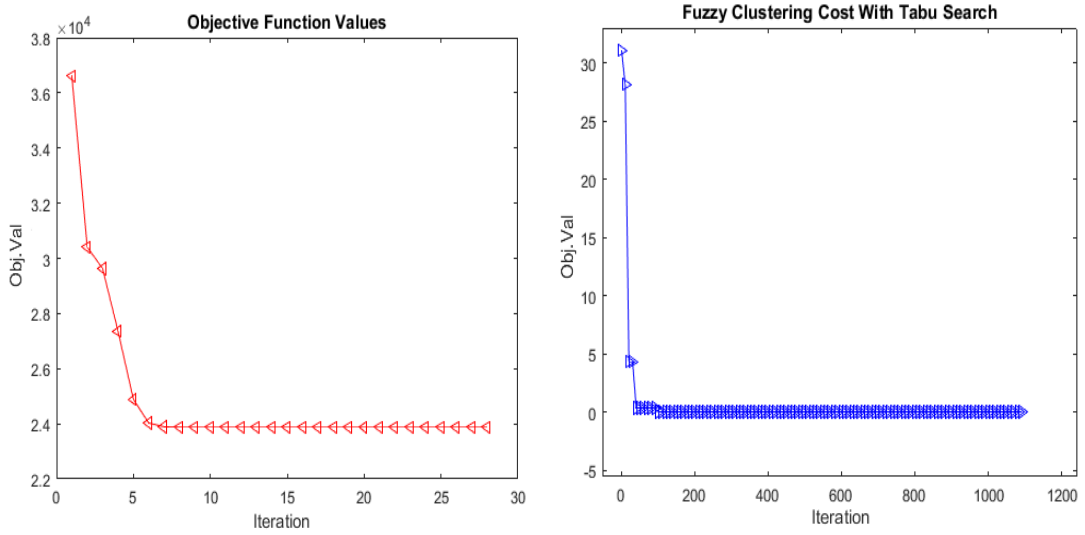

Figure 2-Values of the best Fuzzy obj.fun (red- FCM classic via blue -TS-FCM) for Dolphin

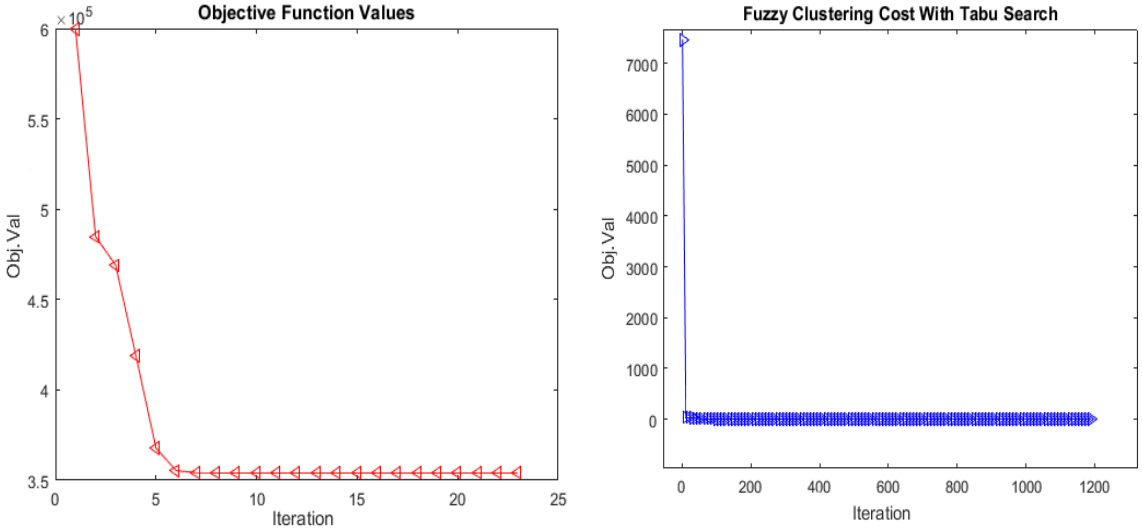

Figure 3.-Values of the best Fuzzy obj.fun (red- FCM classic via blue -TS-FCM) for AF

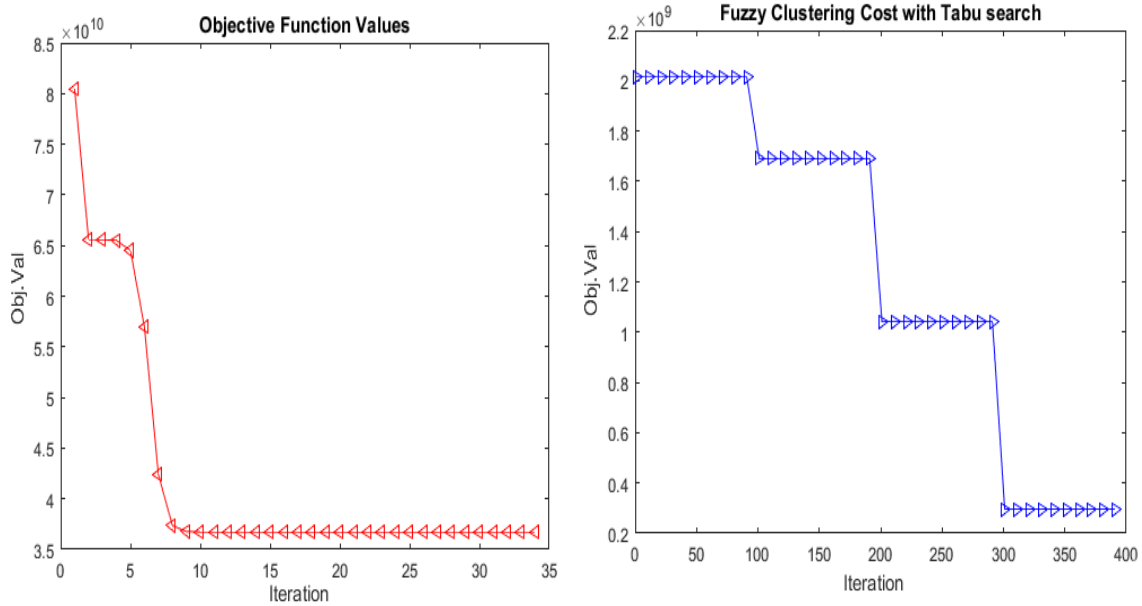

Figure 4-Values of the best Fuzzy obj.fun (red- FCM classic via blue -TS-FCM) for Facebook

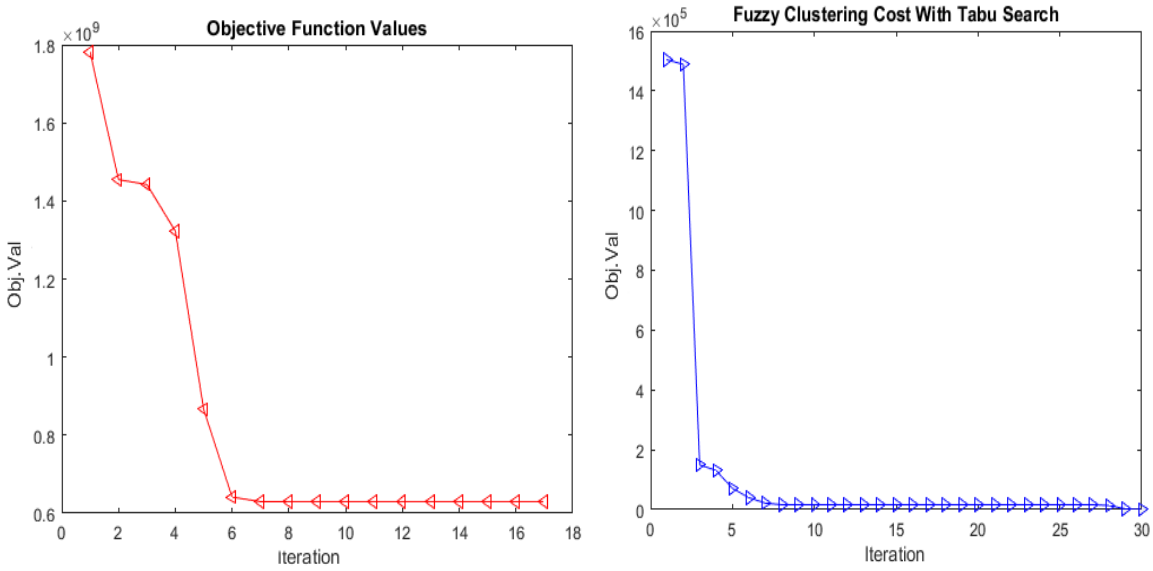

Figure 5- Values of the best Fuzzy obj.fun (red- FCM classic via blue -TS-FCM) for Political

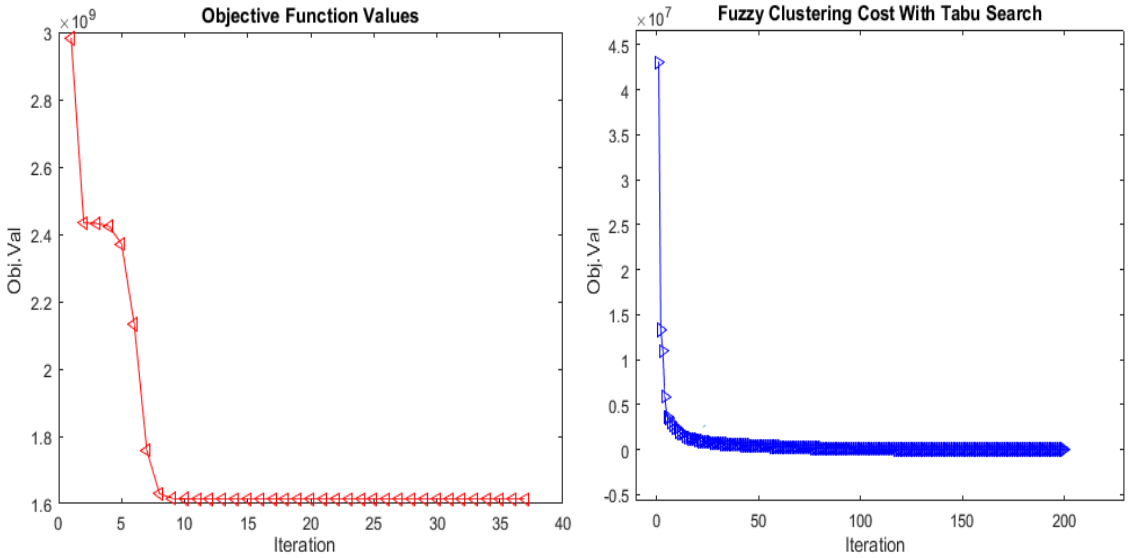

Figure 6-Values of the best Fuzzy obj.fun (red- FCM classic via blue -TS-FCM) for Protein

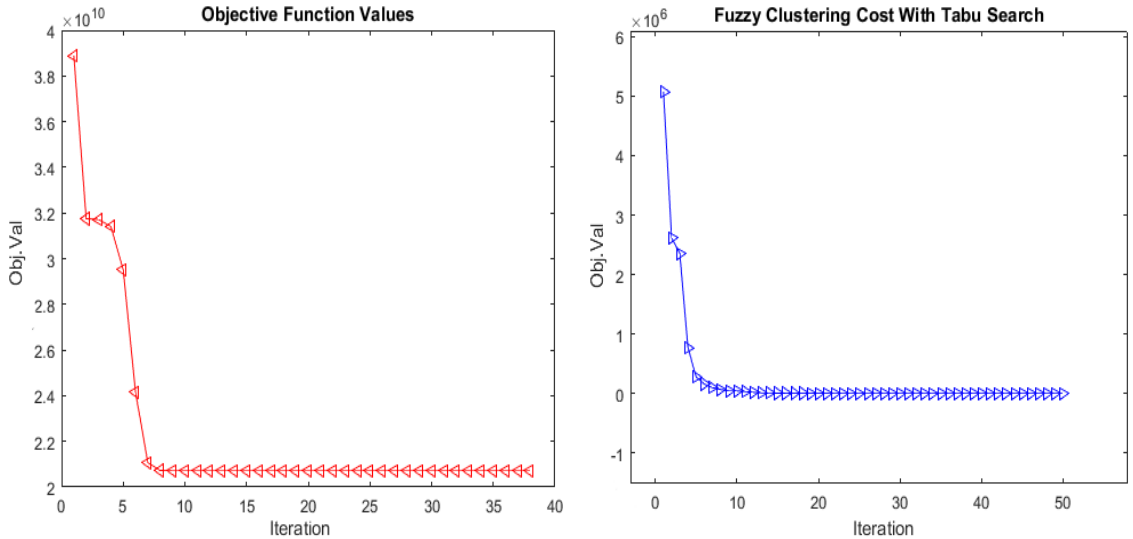

Figure 7- Values of the best Fuzzy obj.fun (red- FCM classic via blue -TS-FCM) for ALIN.

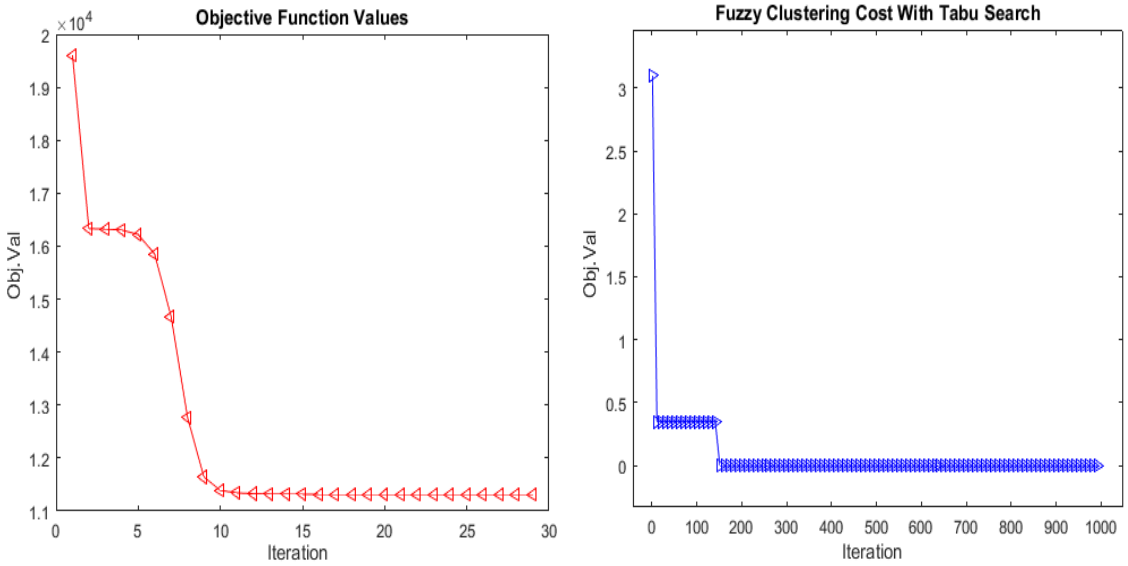

Figure 8-Values of the best Fuzzy obj.fun (red- FCM classic via blue -TS-FCM) for CSN**.**

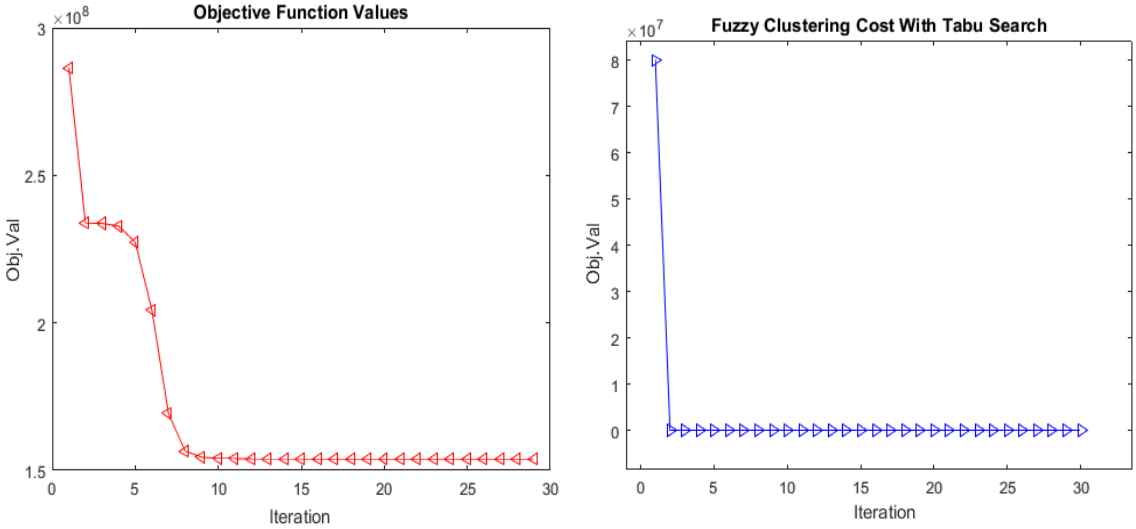

Figure 9-Values of the best Fuzzy obj.fun (red- FCM classic via blue -TS-FCM) for DSN

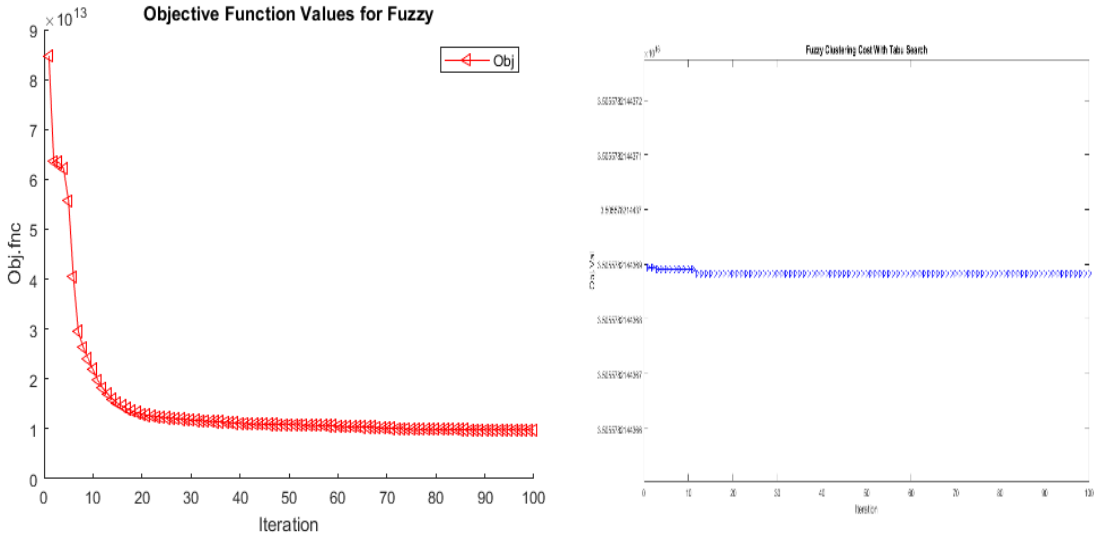

Figure 10-Values of the best Fuzzy obj.fun (red- FCM classic via blue -TS-FCM) for Twitter Network.

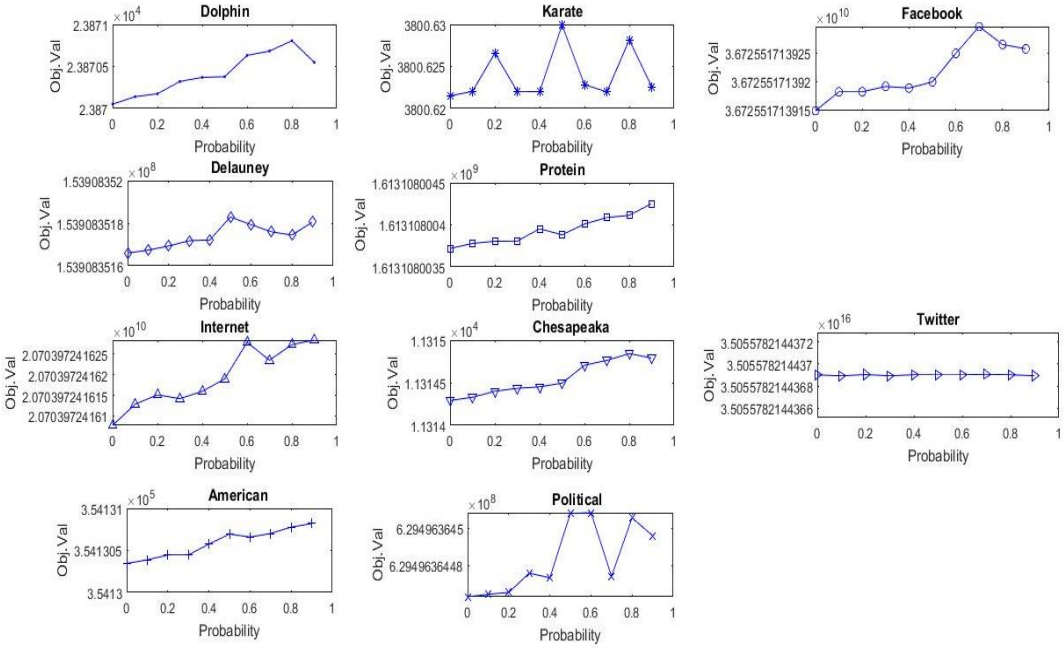

Figure 11-Values of the best Fuzzy obj.fun with respect to P-threshold for the networks.

5. Conclusions

In this paper we present Tabu Search for Fuzzy c-Means to estimate the best clustering by finding the best values of Fuzzy objective function and apply it on different types of real networks, the results show the ability of Tabu search to find the global solution and determine the centroids, this step is important to find the community detection of the big networks.

References

1. Al-Sultan, Khaled S. and Chawki A. **1997.** Fedjki. "A Tabu search-based algorithm for the fuzzy clustering problem." *Pattern Recognition*, **30**(12) (1997): 2023-2030.
2. Ng, Michael K., and Joyce C. Wong, C. **2002.** "Clustering categorical data sets using Tabu search techniques." *Pattern Recognition*, **35**(12) (2002): 2783-2790.
3. Shang, Jiayu, Shiren Li, and Junwei Huang. "A robust fuzzy local Information c-means clustering algorithm with noise detection." Ninth International Conference on Graphic and Image Processing (ICGIP 2017). Vol. 10615.
4. Zhu, Lin, Fu-Lai Chung, and Shitong Wang. "Generalized fuzzy c-means clustering algorithm with improved fuzzy partitions." *IEEE Transactions on Systems, Man, and Cybernetics, Part B* (Cybernetics) **39**(3) (2009): 578-591.
5. Zhang, Shihua, Rui-Sheng Wang, and Xiang-Sun Zhang. "Identification of overlapping community structure in complex networks using fuzzy c-means clustering." *Physica A: Statistical Mechanics and its Applications* **374**(1) (2007): 483-490.
6. Kochenberger, Gary A. **2013.** "Solving large scale max cut problems via Tabu search." *Journal of Heuristics* **19**(4) (2013): 565-571.
7. Delgado, Miguel, Antonio Gómez Skármeta, and Humberto Martínez Barberá. **1997.** "A tabu search approach to the fuzzy clustering problem." *Proceedings of the Sixth IEEE* International Conference on. Vol. 1. IEEE.
8. MiguelM D. **2015.** A Tabu Search Approach to the fuzzy Clustering problem.
9. Glover, F. **1989.** "Tabu search—part I." *ORSA Journal on computing*, **1**(3): 190-206.
10. Glover, F. **1989.** "Tabu search—part II." *ORSA Journal on computing*, **2**(1): 4-32.
11. Glover, F. and Manuel, L. **1998.** "Tabu search." Handbook of combinatorial optimization. Springer, Boston, MA, 2093-2229.
12. Stanford Large Network Dataset Collection, https://snap.stanford.edu/data/index.html