Chapter 4

FUZZY BASED COMMUNITY DETECTION

4.1. Introduction

In real-world scenarios, nodes exhibit degree of belongingness to different communities rather than having membership of single community. Identification of disjoint communities is not sufficient to meet the realities involving partial membership of nodes. Therefore, community detection algorithms not only have to sense network structures but also quantitative affiliations to multiple communities. Fuzzy community detection has been introduced to measure the belongingness of nodes in different communities i.e. membership degrees. In contrast to disjoint community detection, fuzzy community detection not only senses qualitative affiliations to communities but also network structure (Davis and Carley, 2008). Some fuzzy community detection methods have been proposed in recent years. However, many of them require prior information about communities (e.g. number of communities), which may degrade the accuracy of communities.

In this chapter, we used the genetic algorithm with fuzzy concept and compared to other existing methods like as crisp genetic algorithm and vertex similarity based genetic algorithm. We investigated the combination of roulette wheel selection and square quadratic knapsack concept. The usefulness and efficiency of proposed algorithm are verified through the accuracy and quality metrics and provide a rank of proposed algorithm using multiple criteria decision-making method.

We have used the combination of roulette wheel selection and square quadratic knapsack problem on the genetic algorithm. An experimental result shows the improvement on convergence rate of proposed algorithm and discovered communities are highly inclined towards quality.

After both experiments, we employed the new idea of finding the fuzzy community detection in social network with the help of permanence concept with node similarity based genetic algorithm. In this experiment we found the both disjoint community and the overlapping community detection then we compares it as quality wise and accuracy wise with the help of some functions. We utilized the disjoint community structure as an input for our base algorithm. We employed the artificial datasets and the real world datasets for our experiment. It fulfills the role of both disjoint community detection and fuzzy community detection without adding any extra step of genetic algorithm.

4.2. Proposed FGA(Genetic Algorithm with Fuzzy Concept) Approach

4.2.1. Valuation Functions

4.2.1.1. Modularity

Objective Function Modularity is employed to calculate however separated the various vertex sorts from one another and it will be calculated as follows.

$$Q= \frac{1}{2m}\sum_{i,j} A_{ij} - \frac{k_i k_j}{2m} \quad \text{..... (4.1)}$$

Here, Q represents quality function modularity, m represents some edges, Aij represents an entry of contiguousness matrix, ki, kj represents the degrees of vertex i, j severally, ci, cj represents the part of vertex i, j and (x,y)=1 if x = y, zero otherwise.

4.2.1.2. Normalized Mutual Information

It has extensively utilized measures to analysis the network community detection algorithms (Leskovec, et al., 2010) . It equals to one if the detected communities and their ground truth square measure identical whereas the worth is zero if the communities' square measure strictly distinct with the ground truth. It will be defined as

$$I_{norm}(X,Y)=\frac{2I(X,Y)}{H(X)+H(Y)} \quad \text{..... (4.2)}$$

$$H(X)=-\sum_x P(x)\log P(x) \quad \text{..... (4.3)}$$

$$H(X|Y)=-\sum_{x,y} P(x,y)\log P(x|y) \quad \text{..... (4.4)}$$

4.2.1.3. Omega-Index

It is employed to calculate overlapped communities and utilized to estimate the quantity of clusters during which the vertex involve (Wang, et al., 2010).

$$\frac{1}{|v|^2}\sum_{u,v} v\{|C_d| = |C_g|\} \quad \text{..... (4.5)}$$

Where C_d, Cg, represent the set of ground truth communities, identified communities correspondingly, that the pair of vertices u and v shares.

4.2.1.4. Simple Modularity And Fuzzy Modularity

Two concepts of modularity are different in the aspect that in simple GA, the vertex either belongs to a particular community or not. However, in Fuzzy GA we a node can belong to a certain community as a percentage. Simple GA cannot be applied to Fuzzy Algorithm as it simple GA treats the relation between nodes and community as binary and not as a fraction. Hence we devised a different qualitative measure for fuzzy (Liu, 2010)

4.2.1.5. Zhang Fuzzy Modularity

In this objective function (Zhang, et al., 2014) the community detection procedure in begins by partitioning V with spectral bunch applied to G using FCM, once the eigenvector illustration of G is chosen. When a fuzzy c-partition U $\in$Mf c_n is found this manner, the partition is reborn to a possibility c-partition U_λ $\in$Mf c_n of V as follows:

A threshold λ is chosen (presumably $0 < \lambda < 1$), so accustomed extract from the k_{th} row of U $\in$ MF c_n the vertex set $V_k = \{i|u_{ki} > \lambda, 1 \le i \le n\}$. For every vertex i in V_k , the corresponding entry of (U_λ) k (the k_{th} row of U_λ) is set to one. When a omit all k rows of U is completed, the remaining (non−1) entries in U_λ are set to zero.

4.2.1.6. Liu Fuzzy Modularity

$$Q_l = \frac{1}{\|W\|}\sum_{k=1}^{c}\sum_{i,j\in V_k}\left[\left(\frac{u_{ki}+u_{kj}}{2}\right)w_{ij} - p_e(i,j)\right] \quad \text{..... (4.6)}$$

$$d_f(i) = \sum_{j\in V_k}\left(\frac{u_{ki}+u_{kj}}{2}\right)w_{ij} + \sum_{r\notin V_k}\left(\frac{u_{ki}+(1-u_{kr})}{2}\right)w_{ir}. \quad \text{..... (4.7)}$$

4.3. Experimental Work

In Proposed work, we have used the individual encoding of the chromosomes of the Genetic Algorithm. Chromosomes are encoded into binary matrix (Cantú-Paz and Kamath, 2005) .

$$M = \begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1t} \\ m_{21} & m_{22} & \cdots & m_{2t} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nt} \end{pmatrix} \quad \text{..... (4.8)}$$

The Binary Matrix where M is an n×t matrix, t (1<t<n) is the number of communities after partitioning G. n is the number of nodes. Each $m_{i,j}$ indicates how much of the node belongs to a particular community. Without Overlapping these values are binary. However, when overlapping Communities are taken into account, then these values are between 1 and 0. Now this will follow two rules.

$$\sum_{j=1}^{t} m_{ij} = 1$$
$$\sum_{i=1}^{n} m_{ij} > 0 \quad \text{..... (4.9)}$$

Population initialization takes place in which we have used node similarity matrix. It was done because similar nodes that have a high probability of being in the same node are initialized in one community. Then Crossover and Mutation operation are done.

In crossover operation, crossover the 80% of the top solution from the total solution. Modulation operator is applied to the least 20% of the solution. Then with initial population and the resultant population, the best half is chosen. The Quality of the chromosomes is measured through a suitable operator Now to measure the quality of the solution we have used the Modularity operator as it gave better and more consistent result than it`s alternative. Then many iterations of the above are run until the modularity does not increase.

4.4. Experimental Analysis

In this experiment, we employed three different datasets to analyze accuracy and quality of proposed algorithm. We compared proposed algorithm to various version of genetic algorithms. One is simple genetic algorithm (Pizzuti, 2008) , and another is a genetic algorithm with node similarity (Li, et al., 2013). The results are given below.

Parameter	Value	Description
P_n	100	Number of individuals
P_c	0.8	Crossover individuals
P_m	0.2	Mutation individuals
N_{max}	100	Number of iterations
alpha	1/(2*no of communities)	Threshold values

Table 4.1: Value Description of Parameters

4.4.1. Strike Dataset

In a wood-processing facility (Michael, 1997) , workers started strike; the communication structure among the employees in the form of a graph is given below. In Figure 4.1, represent the graphical view of strike dataset in a single color but after applied the SGA we have to find the communities denoted the different colors. Similarly applied the VGA resultant is available in (III) graph. In last graph (IV), we have defined the three colors, but the red color defines the fuzzy communities and overlapping communities. This process is followed to the other datasets also in given below Figure 4.2 and Figure 4.3.

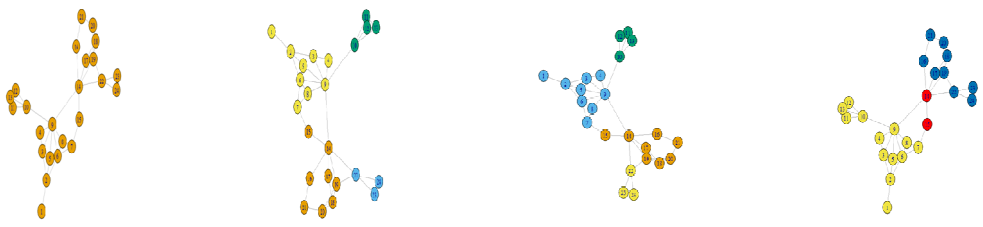

Figure 4.1 Community-based graphical representations for Strike dataset

4.4.2. Karate club dataset

Zachary's Karate Club (Zachary, 1977) is a renowned social network of a university karate club explained in "An Information Flow Model for Collision and Fission in small set" paper by Wayne W. Zachary.

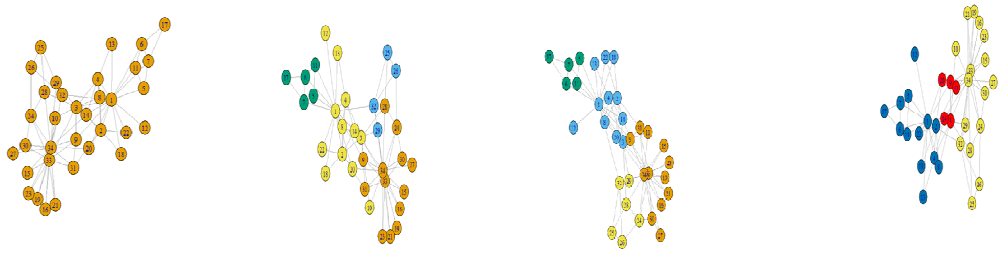

Figure 4.2 Community-based graphical representations for Karate club dataset

4.4.3. Dolphin dataset

The description of the data set is as follows - A directionless social network of frequent relations among 62 dolphins in a society living off Doubtful Sound, New Zealand, as compiled by (Lusseau, et al., 2003).

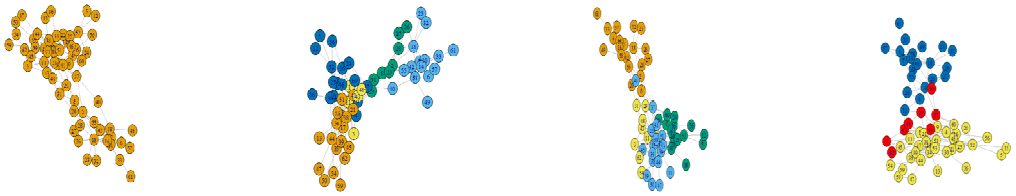

Figure 4.3 Community-based graphical representations for Dolphin dataset

In this experiment, we have found the results in different format means graphical and tabular for used datasets. In Table 4.2, represent the whole result with accuracy and quality functions for all datasets. We have used NMI, entropy, and omega as accuracy parameters while using modularity, conductance, and coverage as quality parameters. First of all, we talk about the Dolphin dataset; FGA has got the higher values for NMI=0.7864, Omega=0.9264, and Coverage=0.94968. Similarly, Entropy=0.2135 and Conductance=0.36653 have got the lower value compare to the other algorithm. Entropy and conductance have contained the inverse

property mean lower value get the good community structure and vice versa. So it means Fuzzy based Genetic Algorithm good performance for this dataset.

Datasets	Algorithms	NMI	Entropy	Modularity	Omega	Conductance	Coverage
Dolphin	SGA	0.470468	0.529532	0.5122226	0.707033	0.935054	0.7924528
	VGA	0.370501	0.629499	0.4816265	0.636171	1.741664	0.7169811
	FGA	0.786448	0.213552	0.1462689	0.926494	0.3665375	0.9496855
Karate club	SGA	0.56209	0.437911	0.4003123	0.84492	1.416041	0.7692308
	VGA	0.491055	0.508945	0.4151052	0.802139	1.15	0.7564103
	FGA	0.661445	0.338555	0.2948628	0.855615	0.301607	0.9871795
Strike	SGA	0.68407	0.315933	0.5557479	0.85507	0.5996503	0.86842
	VGA	0.78437	0.21563	0.5619806	0.91304	0.59033639	0.8684f2
	FGA	0.59047	0.40953	0.3771963	0.82971	0.1311688	0.97368

Table 4.2: Accuracy and quality metric values for various datasets

In karate club dataset, similarly FGA have got the higher values NMI=0.66144, Omega=0.85561 and Coverage=0.987179 compare to SGA and VGA algorithm. FGA have also had good performance for this dataset because it has contained the lower Entropy=0.33855 and Conductance=0.301607. Modularity only has gained the lower value 0.29486 compare to other available algorithms.

Datasets	Algorithms	MCDM Rank
Dolphin	SGA	0.398
	VGA	0.165
	FGA	**0.945**
Karate club	SGA	0.255
	VGA	0.246
	FGA	**0.821**
Strike	SGA	0.444
	VGA	**0.656**
	FGA	0.425

Table 4.3: MCDM ranking score obtained for different datasets

In Strike dataset, VGA has got the higher NMI=0.78437, Omega=0.91304, Modularity =0.56198 and Entropy =0.21563 have contained the lower value. Only Conductance=0.1311 and Coverage=0.97368 have supported the FGA algorithm.

Finally, we analyzed that FGA have performed good for large datasets but average performance for the small datasets. It means fuzzify genetic algorithm find the overlapping communities and maintain the accuracy and quality also. Another fact is that VGA is good for the small datasets. FGA have good performance for the whole datasets (Strike, Karate, and Dolphin) but main focus on the accuracy of the proposed algorithm. In Table 3, we will show the final results with proposed algorithm compare to other algorithms called MCDM rank table. It contains the MCDM (Multiple criteria decision making) ranks (Pizzuti, 2008) In this rank system, we will summarize the both quality and accuracy metrics as a single score value. Figure 4.4 to Figure 4.6 represents the MCDM graphs for individual datasets for all the algorithms. According to the graphs, we will easily show that which algorithm is better to score comparing to other. FGA have got the highest MCDM score For Dolphin and Karate club datasets. It means we have proved that proposed FGA have better performed compare to existing Genetic algorithms. The new fuzzy algorithm (FGA) is also hopeful for further work with new ideas.

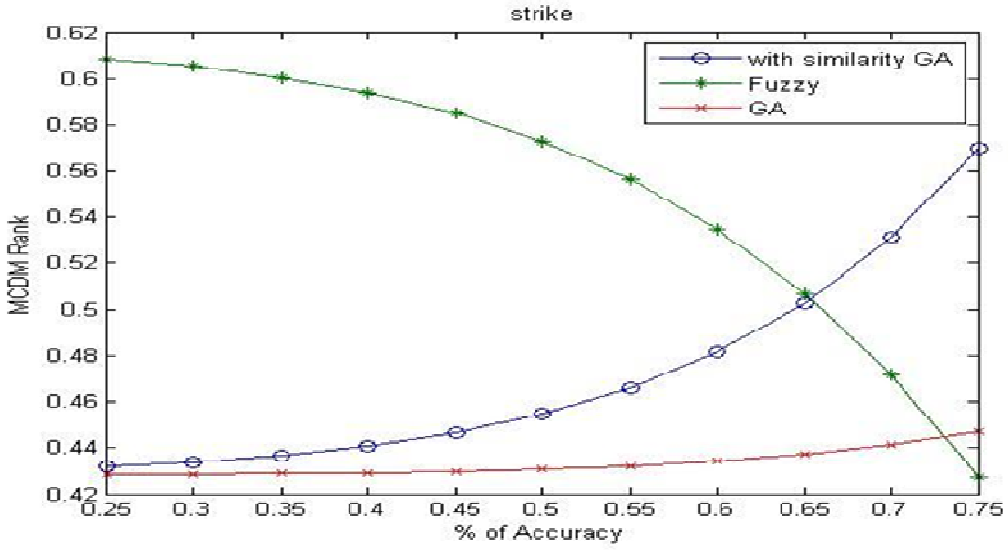

Figure 4.4 MCDM ranking graphs for Strike datasets with variation of accuracy and quality

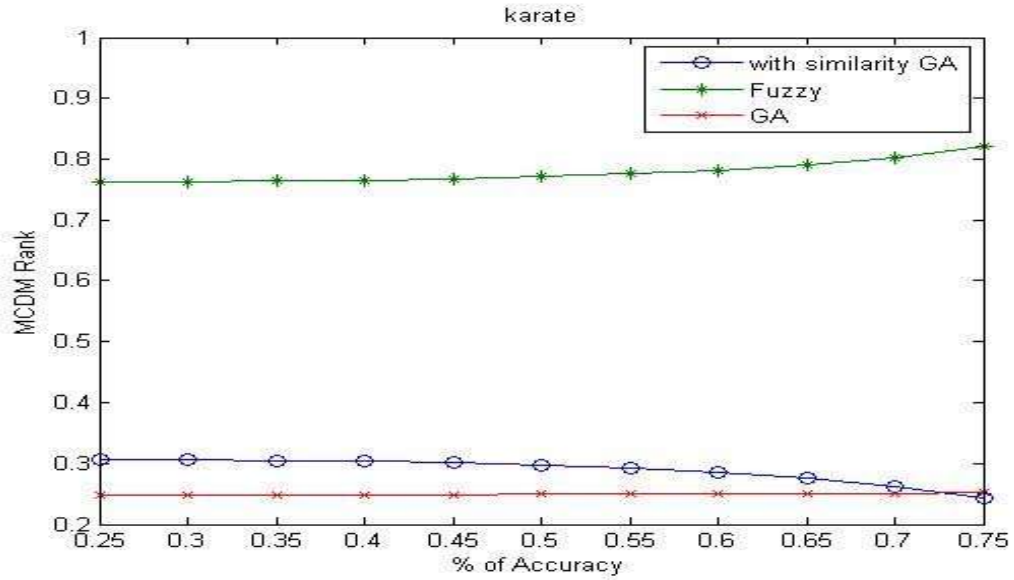

Figure 4.5 MCDM ranking graphs for Karate club datasets with variation of accuracy and quality

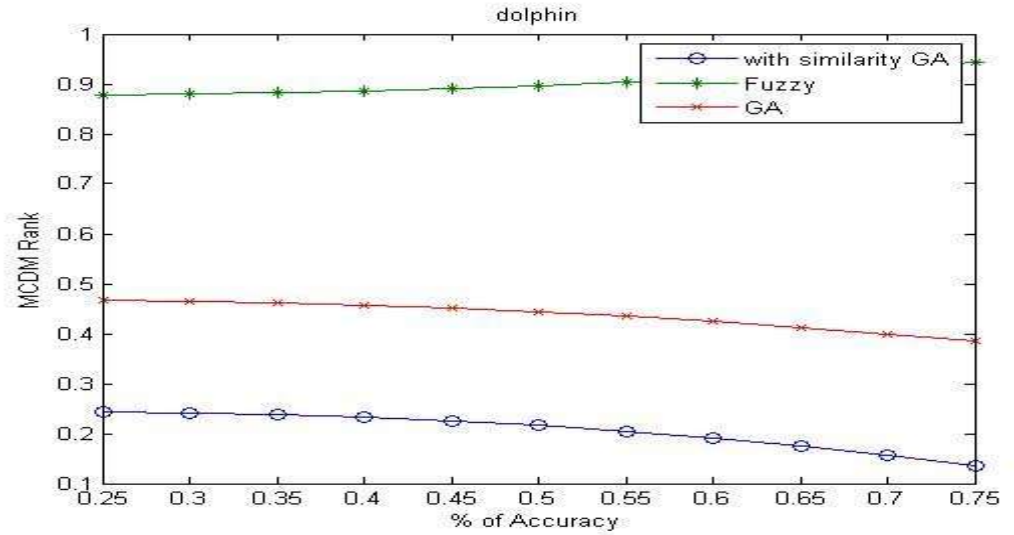

Figure 4.6 MCDM ranking graphs for Dolphin datasets with variation of accuracy and quality

In this experiment, genetic algorithm for community detection in social networks, matrix encoding empowers traditional individual crossover and no additional requirement of decoding. Initial individuals generated by measuring distance are diverse yet retain an acceptable level of accuracy. As evident from MCDM graphs, the fuzzy based algorithm gives a better result than other crisp community algorithms.

After this experiment, we employed the genetic algorithm for fuzzy community detection in social networks but we want some more revised GA so that we choose the GAFCD (Zhang, et al., 2016) . It is able to find the both fuzzy partition and crisp partition while MSFCM (Lin, 2014) can only find fuzzy partition and GALS (Ishibuchi and Yamamoto, 2004) can only find crisp partition. Unique feature of GAFCD the first genetic algorithm for finding truly fuzzy (i.e. inclusive of both fuzzy and crisp communities) with max modularity community structure in a network.

4.5. Modified GAFCD

In this proposed work, we have done different type of experiments on genetic algorithm and check the performance of modified GA. I have done the modification in the algorithm but not change in internal architecture. In this Genetic algorithm, input datasets in the form of adjacency matrix and some other input parameters given below in Table 4.4 and Table 4.5. Output is the form of Partition and the cover matrix (U).

Parameter	Value	Description
m	1.7	Used in determining the membership of each node
c_p	0.1	Percentage of individual selected directly
n_{pc}	10	Number of individuals with given number of partition
p_m	1.0	Mutation percentage
p_c	0.9	Cross-over percentage
Occ_{max}	10	Number of occurrences of generation with termination condition
ε	10^{-5}	Termination condition
t_{max}	100	Number of iterations
c_{min}	2	Minimum number of partitions of social network
c_{max}	10	Maximum number of partitions of social network

Table 4.4: Values of different parameters

We proposed the following two changes in GAFCD to improve the final modularity value and NMI value for the fuzzy community detected.

- While calculating the modularity value (Q-value), we calculated the contribution of each community separately, while also maintaining the combined Q-value of

each individual of the population. Q value was given by the trace of a c∗c matrix, where c=number of communities. The matrix was given by U∗B∗U'. So for all of the c communities present in this matrix, we stored the diagonal values in a vector called Q per community.

- In the fuzzy crossover function, after applying Roulette wheel selection for calculating the optimal number of communities in the crossover child, random selection of individual communities was done from the union of the communities of the two parents. Instead of doing a random selection in this step, we used the Q per community vector calculated above to select the individual communities from the union. We applied Roulette wheel selection in this step.

Dataset	Symbol	Vertices	Edges
Karate	K	34	78
Dolphin	D	62	159
PolBooks	P	105	441
Football	F	115	613
Jazz	J	198	2742
Sawmill	S	36	62
LesMis	L	77	254
Words	W	112	425
Metabolic	M	453	2025

Table 4.5: Description of the used Datasets

4.5.1. Experimental Result & Analysis

We compare MGAFCD with GAFCD, MSFCM and GALS on 10 real-world data sets that are described in Table II. Metabolic Network is an undirected, weighted graph, but it has 15 loops or self-connections (none of the algorithms here can handle these loops). Here, we simply remove these loops to make Metabolic Network a simple graph. Karate and LesMis datasets are weighted and undirected, while all the other data sets are undirected and unweighted. . The different steps involved in SGA are:

- Initialization: Before evolution, populations of individuals are randomly initialized.
- Fitness Evaluation: In every iteration, the competitiveness of individuals is first evaluated on the basis of a quality function and a fitness score is assigned to each individual by this quality function. m= 1.7, n_{pc}=10. 10 partitions with each community size from c_{min} to c_{max} are generated and taken as single individuals. Population size=n_{pc}*(cmax-cmin+1).
- Survival of the Fittest: Individuals are selected for crossover and mutation with pre-define probabilities pc and pm respectively. p_c= 0.9, pm=1.0, c_p=0.1
- Evolution: The selection process guarantees that an individual with a higher fitness score will be chosen with a higher probability.
- Iteration: After a new generation is produced, SGA terminates and returns the best individual of the current generation if some stopping conditions are satisfied. Number of iterations=100.

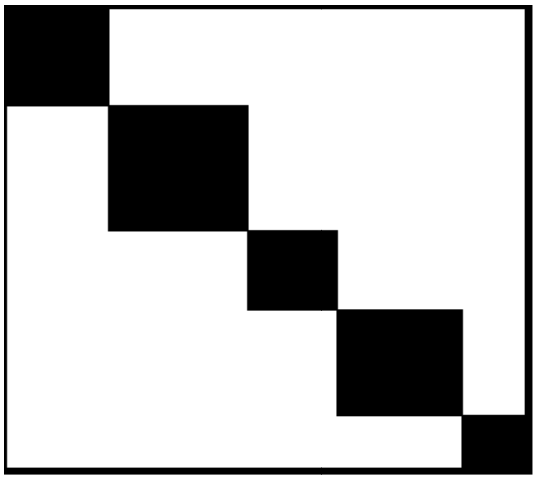

Figure 4.7 Dolphin, Q=0.5285, c=5

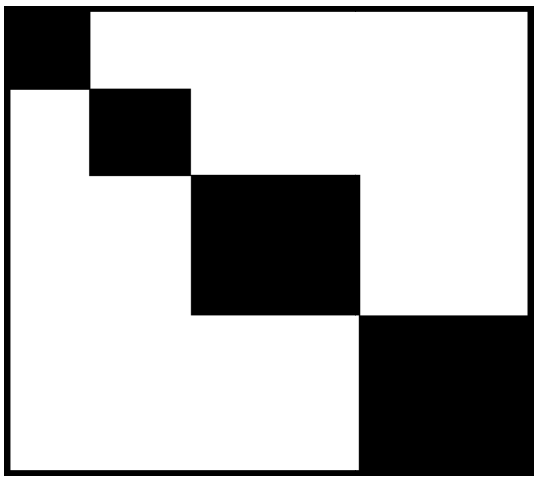

Figure 4.8 Karate, Q=0.4449, c=4

In this experiment, Fig 4.7- 4.10 represents fuzzy community partitions of our given dataset or social network. In these figures, relative sizes of each of the communities are shown.

- Figure 4.7 represents partition of Dolphin dataset. It forms 5 communities with 12, 20, 9, 16 and 5 nodes respectively. This partition gives Q value as 0.5285.

- Figure 4.8 represents partition of Karate dataset. It forms 4 communities with 5, 6, 11 and 12 nodes respectively. This partition gave a Q value of 0.4449.
- Figure 4.9 represents partition of Jazz dataset. It forms 4 communities with 53.4, 62, 21.6 and 61 nodes respectively. This partition gave a Q value of 0.4452.
- Figure 4.10 represents partition of metabolic dataset. It forms 9 partitions with 36.75, 60, 44, 11, 74, 107.25, 7, 93 and 20 nodes respectively. This partition shows a Q value of 0.4447.

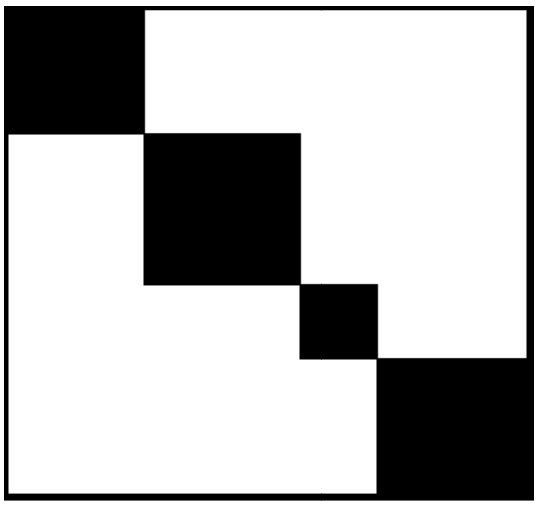

Figure 4.9 Jazz, Q=0.4452, c=4

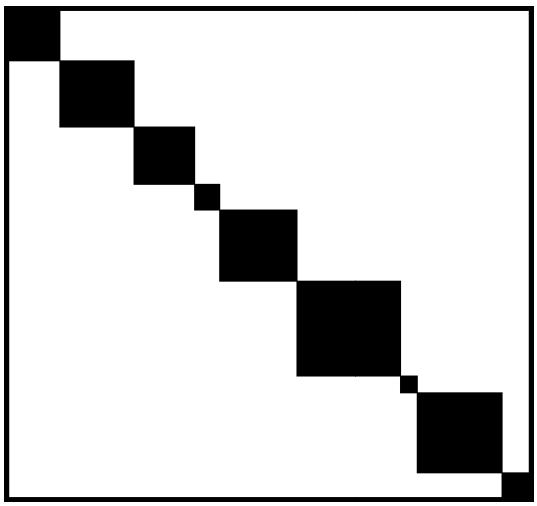

Figure 4.10 Metabolic, Q=0.4447, c=9

Table 4.6, shows the values that we compared between the MSFCM, GAFCD, GALS and our algorithm, GAFCD. It involves modularity values, i.e. Q_{best}, Q_{std} and Q_{mean}. In that experiment, MGAFCD modularity values increased by an approximate factor of 0.02 in the metabolic dataset and others. We also received improved values of $_{Qstd}$ in comparison to MSFCM, GALS algorithm for all datasets. For some of the smaller datasets like Dolphin, Qstd decreased (and thus improved) in comparison to GAFCD. But, for some of the bigger datasets like Metabolic, this value increased, making the communities found a bit inconsistent, though with better modularity. In datasets like Karate, Dolphin and Football, partition found is crisp as was in the implementation of GAFCD. But for datasets like Jazz and Metabolic, fuzzy communities are observed. GAFCD and MGAFCD have same number of communities for Metabolic dataset but different Q values. This is in consistence with the fact that we have selected the optimal number of communities in the way similar to the way GAFCD did. But, we

have improved the algorithm in selection of communities that form the next generation individual. Thus, it shows same number of communities but different modularity value.

Algorithm	Modularity	K	D	P	F	J	S	L	W	M
MSFCM	mean Q	0.4129	0.3963	0.4596	0.5266	0.398	0.3279	0.4897	0.0052	0.2588
	std Q	0.0001	0.0043	0.0009	0.0008	0.02	0.0001	0.0108	0.0013	0.0118
GAFCD	mean Q	0.4449	0.5285	0.5272	0.6046	0.4452	0.5501	0.5667	0.3107	0.4261
	std Q	0	0.0001	0	0	0	0	0	0.0009	0.0014
GALS	mean Q	0.4449	0.5282	0.5272	0.6045	0.4448	0.5501	0.5313	0.3094	0.4153
	std Q	0	0.0004	0	0.0003	0.0001	0	0.0013	0.002	0.0068
MGAFCD	^{Q}best	0.4449	**0.5285**	**0.5275**	0.6046	0.4452	**0.5503**	0.5667	**0.3107**	**0.4415**
	c	0	0	0	0	0	0	0	0.0005	0.005
MSFCM	^{Q}best	0.4132	0.3991	0.4601	0.5268	0.4078	0.328	0.4971	0.0083	0.2876
	c	3	4	3	10	4	5	5	9	7
GAFCD	^{Q}best	0.4449	0.5285	0.5272	0.6046	0.4452	0.5501	0.5667	0.3126	0.4287
	c	4	5	5	10	4	4	6	7	9
GALS	^{Q}best	0.4449	0.5285	0.5272	0.6046	0.4449	0.5501	0.5439	0.3121	0.428
	c	4	5	5	10	4	4	6	7	18
MGAFCD	^{Q}best	0.4449	0.5285	**0.5275**	0.6046	0.4452	**0.5503**	0.5667	0.3126	**0.4447**
	c	4	5	5	10	4	4	6	7	9

Table 4.6: Compared Performance of Community Detection Algorithms

We have a successfully modified the existing GAFCD algorithm. The existing GFCD algorithm did the following: It made a fuzzy partition of the network using one step FCM initialization. It treated each partition as an individual. The modularity value for the partition was used as the objective function to evaluate each partition. These partitions were then sorted according to these modularity values. Then certain percentage of individuals was directly selected for the next generation. Next, crossover was done. In this we combined the two parents. Suppose parent 1 has c_1 communities and parent 2 has c_2 communities. Then the child made can have the number of communities ranging from 2 to c_1+c_2. We calculated average fitness for all the communities with a given number of partitions. Then Roulette Wheel selection was done to obtain the optimized number of communities in the child. Then mutation was done which involved modifying each column of partition using q_{pip} solver assuming that the other columns remain the same. The modification we did proved effective for large dataset like metabolic as it used informed selection. It was not quite effective for smaller datasets as random selection and informed selection will select almost the same set of communities. Also, mutation operator will

modify the partition of smaller datasets easily thus eliminating the need for informed selection. Whereas, in case of large datasets, the modification increased the modularity values and made a difference.

In further research work we have employed the genetic algorithm for more optimized out in the form of community structure in social networks. We want best utilization of the genetic algorithm & found the crisp and fuzzy community structure in a single step. We employed the vertex similarity and permanence concept with genetic algorithm.

4.6. Node Similarity Based Genetic Algorithm with Permanence Concept (NSGAP)

In the experiment, we propose a two-step algorithm which makes use of traditional community detection through Genetic algorithm upon which, Node similarity based Permanence concept algorithm is applying for obtaining overlapped communities.

In Figure 4.11, The first step involves a genetic random walk on the population initialized using the principle of vertex similarity. The three operations that comprise this random walk are Selection, Crossover, and Mutation. After a certain number of generations, the features of the population converge to give a disjoint community structure which is of high resemblance to the actual overlapped structure of community.

In further step, instead of applying a separate overlapping community finding algorithm, we suggest a technique which calculates the membership of each vertex in every cluster, using the concept of permanence (Chakraborty, et al., 2014).

Permanence of a vertex is the combination of the external distribution of links of the node and the internal connectedness with its community, i.e. the clustering coefficient.

$$\mathrm{Perm}(v) = \frac{I(v)}{E_{\max}(v)} \times \frac{1}{D(v)} - (1 - c_{in}(v)) \qquad \text{..... (4.10)}$$

Where I (v) is the quantity of inner associations of v, D (v) is the level of v, E_{max} (v) is the greatest associations of v to a particular cluster and c_{in} (v) is the clustering coefficient among the interior neighbors of v.

4.6.1. Permanence Based Vertex Replication

We start with the disjoint community structure and take a set of all the vertices which have at least one external connection. For each vertex in this set,

- The aggregate permanence of v and its neighbors in their assigned clusters is computed using equation 4.10.
- The vertex v is expelled from its own particular cluster and put in each of its outer clusters independently. Total permanence is computed for the new structure.
- If the absolute value of the difference of permanence values obtained in above two steps is less than a threshold value, vertex v is replicated into the recent corresponding cluster; otherwise v is kept in the original community.
- As a result the new community membership of all the vertices is obtained.

PARAMETER	VALUE	DESCRIPTION
α	0.88	nodes similarity factor
P_n	100	Size of individuals
P_c	0.2	crossover factor
P_m	0.5	Mutation ratio
N_{max}	50	Number of generation
θ	1.13	Threshold of difference in new and current permanence of a vertex

Table 4.7: Constraint of the Experiment

The constraint values of the experiment fix as shown earlier. These could be altered as per the situation, depending on the size of the network, variation in convergence time and extent of overlap required among communities. For the networks used in this experiment, optimal values of Pc and Pm were found out to be 0.2 and 0.5 respectively.

The proposed work was keep running on a Microsoft Windows 10 (x64) working framework utilizing R 3.2.3 programming Platform and R-Studio 0.99.491 IDE; Intel ® Core ™ i5-3230M CPU @ 2.60 MHz processor, 4.00 GB memory and 1 TB hard disk.

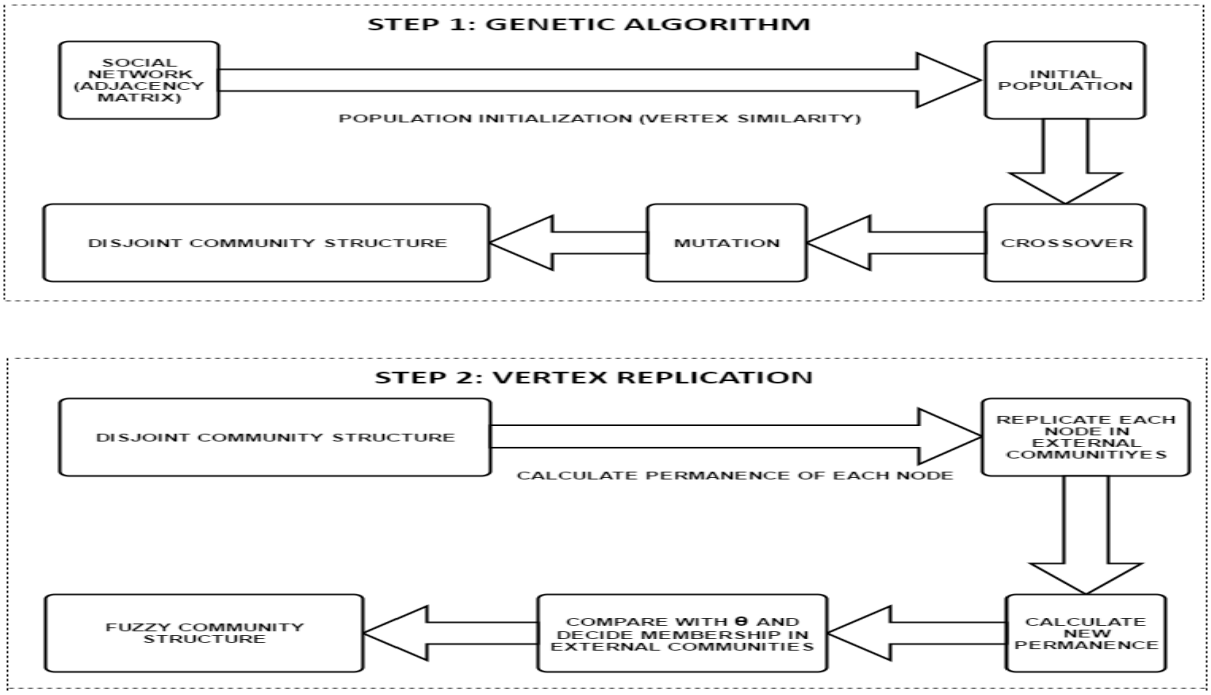

Figure 4.11 Block diagram of proposed Method

4.6.2. Experimental Result & Analysis

The genetic algorithm for disjoint community detection was run for a total of 6 times for each dataset, in each of which the number of iteration was taken as 50. For each of the community structures obtained, various quality and accuracy functions were computed. These are as follows:

FUNCTION	MODULARITY		NMI		ACCURACY		F-MEASURE		ENTROPY	
NETWORK	BEST	Avg.	BEST	Avg.	BEST	Avg.	BEST	Avg.	BEST	Avg.
STRIKE	0.560	0.558	1.000	0.870	1.000	0.908	1.000	0.753	0.000	0.192
KARATE	0.420	0.400	0.759	0.694	0.877	0.819	0.309	0.241	0.310	0.405
DOLPHIN	0.520	0.490	0.775	0.672	0.800	0.733	0.667	0.415	0.386	0.474
LFR	0.650	0.605	1.000	0.939	1.000	0.968	1.000	0.798	0.000	0.061

Table 4.8: Function values for disjoint community structures

FUNCTION	MODULARITY		NMI		ACCURACY		F-MEASURE		ENTROPY	
NETWORK	BEST	Avg.	BEST	Avg.	BEST	Avg.	BEST	Avg.	BEST	Avg.
STRIKE	0.553	0.544	0.855	0.796	0.934	0.875	0.96 7	0.732	0.197	0.197
KARATE	0.386	0.377	0.797	0.699	0.942	0.860	0.970	0.345	0.195	0.290
DOLPHIN	0.495	0.445	0.665	0.587	0.857	0.783	0.583	0.404	0.247	0.333
LFR	0.652	0.602	1.000	0.954	1.000	0.962	1.000	0.802	0.000	0.089

Table 4.9: Function values for overlapping community structures

As seen from the function values obtained, the genetic algorithm performs quite accurately on all the real networks used. For the optimum values of Pc and Pm as 0.2 and 0.5 respectively, α as 0.88, initial population as 100 and maximum number of generations as 50, the average accuracy of the algorithm is greater than 90% for Strike and LFR whereas 82% for Karate and 73% for Dolphin. The average Q-values also show the good nature of communities that are formed by the algorithm.

The function values obtained after applying the vertex replication algorithm with θ= 1.13, show an overall decrease which is understood as no new optimization is done at this step. However the overlapping community structures thus obtained, highly resemble the ground-truth community structures as can be seen from the Accuracy and F-measure. The average accuracy of the algorithm is greater than 95% for LFR, around 87% for Strike and Karate whereas 78% for Dolphin.

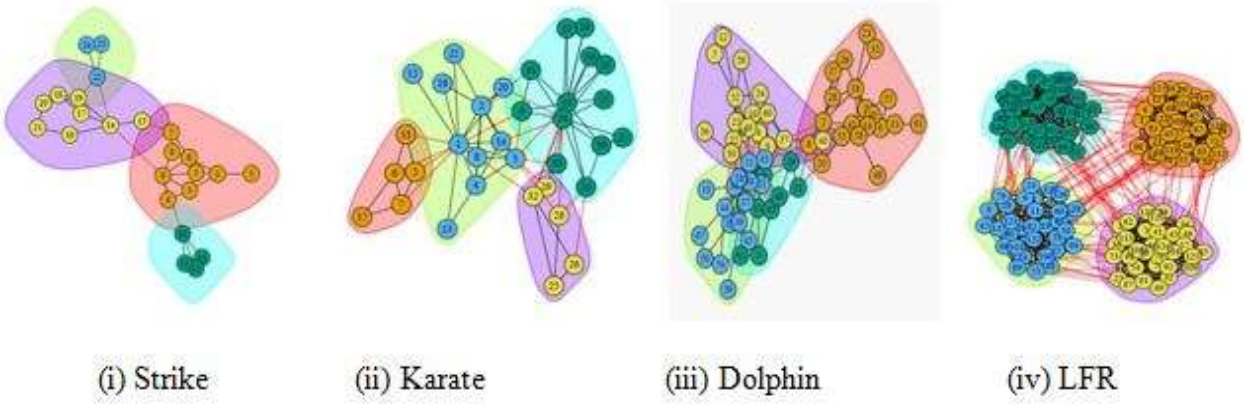

Figure 4.12 Community based graphical representation for datasets

In Figure 4.12, these show the community structures obtained by applying the proposed method. Each community is depicted by a separate color and boundary. The overlapping nature of the communities can be clearly seen for all the used datasets.

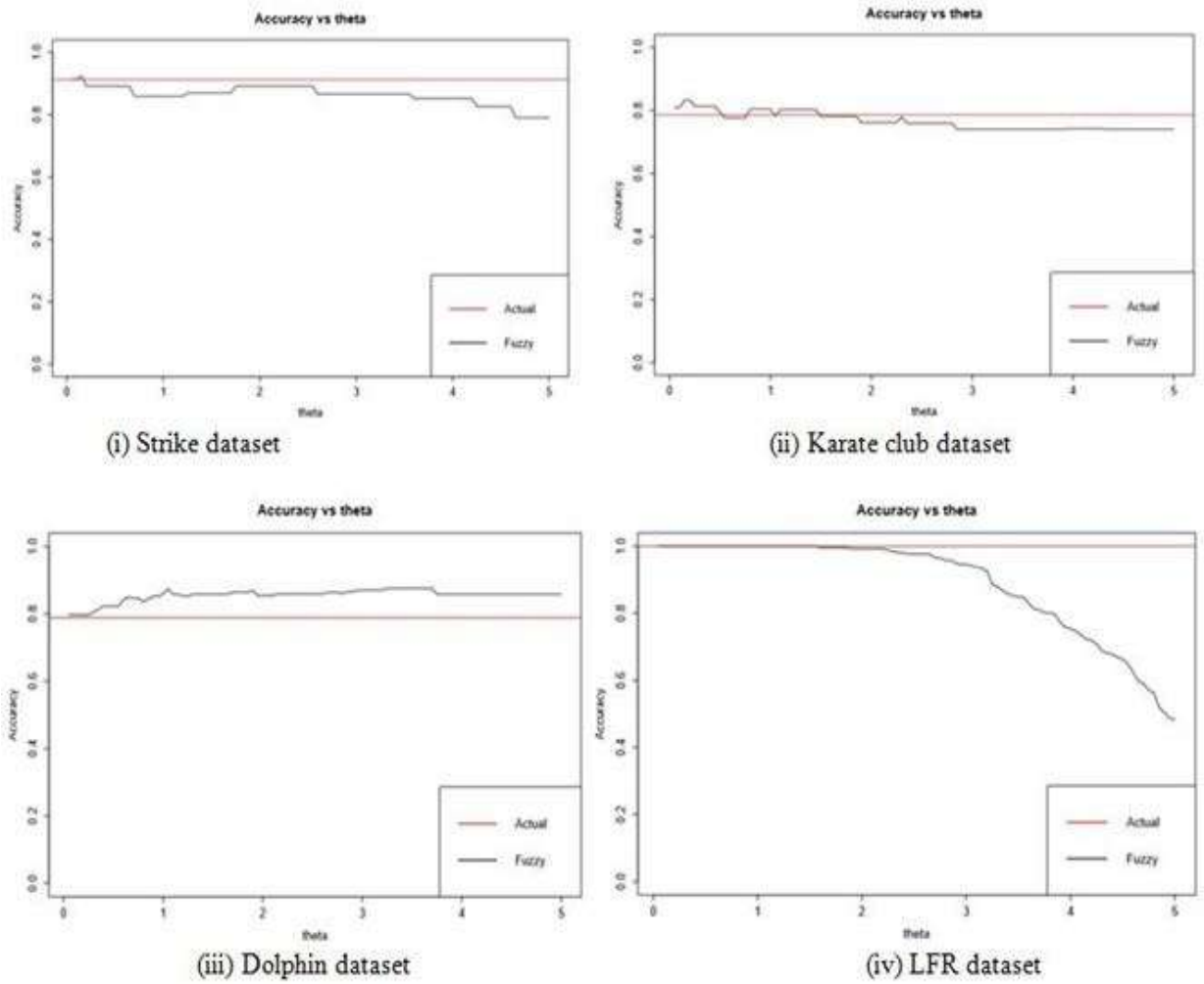

Figure 4.13 Accuracy based comparison between overlapping and non-overlapping communities for datasets

Table 4.8 & 4.9 shows that the values of Q (modularity) with different datasets for the disjoint and overlapping communities. A traditional rise in modularity is seen in the beginning. As number of iterations becomes 40, the algorithm begins to converge; it means convergence rate of this algorithm is fast for the disjoint community compare to overlapped communities. The modularity becomes almost constant for number of iterations more than 40.The Modularity values in the plot for fuzzy are slightly lower than that in the plot for disjoint but become constant for number of iterations more than 40 just like in the case of disjoint. This trend can be seen for all the used four real world datasets. In this experiment, proposed algorithm performed well accuracy wise and quality-wise for both disjoint and overlapped communities in all used real world datasets.

In Figure 4.13, shows the Plot of Accuracy verses θ for all real world datasets. The accuracy on y-axis is for the overlapped & disjoint communities and x-axis is representing the θ value. When θ is low, the accuracy is similar to that of disjoint & overlapped community structures. As θ is increased, the accuracy decreases, due to the higher degrees of membership of vertices in other communities. As θ increases, the overlapping nature of the communities increases thus making the result less accurate. This trend can be seen for all the four real networks used.

Experimental results show that this approach provides meaningful overlapping community structures, which are of high similarity with the ground-truth community structures. The extent of overlap among communities can be varied using the θ parameter. The fuzzy community structures that we obtained by following the proposed method showed high resemblance with the ground truth community structures but showed slight decrease in the corresponding Q-values. We would like to devise a method which incorporates overlapping memberships of vertices during the crossover and mutation operations, further reducing the overhead.

4.7. Conclusion of the Chapter

In this Chapter, we used the genetic algorithm with fuzzy concepts like as fuzzy modularity and resultant compared to simple genetic algorithm & vertex based genetic algorithm. We have employed the real world datasets like as karate club, strike and dolphin sociality. We validate our output through the accuracy and quality functions and also provide rank of with the help of multiple criteria decision making method. In future, the complexity of the proposed algorithm could also be minimized by using a better quality function & Accuracy could also be increased. In further we will use the fuzzy concept with new version of Genetic algorithm means GAFCD (Genetic algorithm for fuzzy community detection). In that method we have done some modifications such as Combination of roulette wheel selection and square quadratic knapsack problem.

Another experiment after MGAFCD (Modified Genetic algorithm for fuzzy community detection), NSGAP (Node Similarity Based Genetic Algorithm with Permanence Concept) in this experiment, initial individuals are generated by vertex similarity approach. As a result initialization is diverse yet retains a satisfactory level of accurateness. In this proposed work, we

would like to put emphasis on the fact that, considering the importance of analyzing social networks both effectively and efficiently, the proposed method eliminates the need of a separate detection algorithm for fuzzy community structures. It fulfills the role of both disjoint community detection and fuzzy community detection without adding any extra complexity to that of genetic algorithm.

However, a significant disadvantage of this approach is that it provides the same quantity of overlapping clusters that the disjoint group structure contains. It is probably that due to the overlapping nature of clusters, new groups may form. To compensate for this, we would like to incorporate a novel changes in the existing algorithm that would incorporate the formation of new clusters. We would like to introduce a novel parameter that will decide the extent to which new communities can be tolerated.