Chapter 3

GENTEIC ALGORITHM WITH OBL

3.1 Introduction

A fundamental problem in community detection is identification of inaccurate communities. The properties of network elements are generally explored in three levels of abstraction to correctly detect communities. First, the node level properties that is associated with ground level entities of the network such as nodes or connections. The node level properties such as various node centrality measures (Crucitti, et al., 2006) and similarity between two nodes (Jeh and Widom, 2002; Watts, et al., 2002) are used extensively for community detection. Second, the group level or community level properties that are associated with group of nodes or connections or sub-graphs. Popularly used community level properties include modularity of community (Clauset, et al., 2004; Newman, 2012) , similarity of communities (Ahn, et al., 2010) and density of communities (Lozano, 2007).At Last, the network level properties that deal with various properties of the network. Network level properties are defined mainly in terms of cut, which include network level properties such as ratio cut (Yang, et al., 2014), normalized cut (Van Den Heuvel, et al., 2008) and conductance (Viswanath, et al., 2010) etc. Community detection algorithms utilize one or more of the properties discussed above to identify communities (Fortunato, 2010; Newman, 2004; Radicchi, et al., 2004). However, evolutionary algorithm like GA suffers a lot of due to their slow convergence rate. Random convergence of solutions in a alternate problem with respect to a fitness function is another disadvantage of GA (Erol and Eksin, 2006). Genetic algorithm is a random technique, so it has more options to search the best result but not in limited time duration (Ali, et al., 2012; Iqbal, et al., 2011) .

In this chapter, we employed the opposition based learning concept for the community detection in social network. We use OBL (Tizhoosh, 2005) technique for initialization phase in genetic algorithm. Most of the evolutionary algorithm's convergence rate is based on the initialization process, so it major role plays in the whole method and output also. Convergence rate of most of the evolutionary algorithms is mainly decided by its initialization prior. Modified Crossover Opposition Based Genetic Algorithm (MCOBGA) is proposed algorithm to enhance the properties and minimize the drawbacks of the genetic algorithm for community detection in social network. Detail procedure of the proposed algorithm is given below:

3.2 Modified Crossover Opposition Based Genetic Algorithm (MCOBGA)

In this section, detailed description of the proposed algorithm has been given. The key terms used in this work have been discussed followed by the algorithm.

3.2.1 Network Modularity Objective Function

Let G (V,E) represent a social network, where V represent the set of nodes and E represent the set of edges of the social network, respectively, and where n and m are the number of nodes and edges. Network modularity function is a well-known term in social networks. Network modularity is also known as Q-function, is widely used to quantitative evaluate the community detection of social networks. It is expressed as follows (Newman, 2006), modularity value for a network can be calculated by using Equation (3.1) as:

$$Q=\frac{1}{2m}\sum_{ij}\left[a_{ij}-\frac{k_ik_j}{2m}\right]\delta\left(c_i,c_j\right) \quad \text{......... (3.1)}$$

Where a_{ij} is an element of the network adjacency matrix $A=\left(a_{ij}\right)_{n\times n}$. If v_i and v_j are connected by an edge, then, $a_{ij}=1$ or $a_{ij}=0$; c_i and c_j represent the communities to which v_i and v_j belong, respectively; if $c_i=c_j$, then $\delta\left(c_i,c_j\right)=1$, or $\delta\left(c_i,c_j\right)=0$; k_i and k_j are respectively the degrees of v_i and v_j, with $k_i=\sum_{j=1}^{n}a_{ij}$, $kj=\sum_{i-1}^{n}a_{ij}$, $i,j=1,2,..........n$ .According to nature of Q value like as close to 1 means stronger community structure and value near about 0 means weak community partition of G.

3.2.2 Individual encoding

String encoding (Whitney and Berndt, 1999) and graph-based encoding (Menendez, et al., 2014) are two types of encoding techniques which are widely used. The traditional methods of encoding like value encoding does not support modified crossover and mutation operation. Binary encoding has found to be the best suited for the said problem as it support modified crossover and mutation operation. Individuals are encoded into a binary matrix, as described in (Oggier and Datta, 2011). In binary encoding, graph G is represented in the form of binary matrix M as shown below:

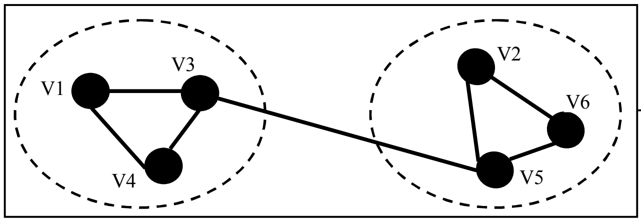
$$M = \begin{vmatrix} m_{11} & m_{12} & \dots & \dots & m_{1g} \\ m_{21} & m_{22} & \dots & \dots & m_{2g} \\ \dots & \dots & \dots & \dots & \dots \\ m_{f1} & m_{f2} & \dots & \dots & m_{fg} \end{vmatrix}$$

Where M is an $f \times g$ matrix, g$(1 < g < f)$ is the number of communities after partitioning G. Row i$(1 \le i \le f)$of M corresponds to the assigning result of v_i, and column j $(1 \le j \le g)$ corresponds to community c_j, if v_i belongs to c_j, then $m_{ij} = 1$, or $m_{ij} = 0$. Since any node of G must belong to a single community, M must follow the constraints defined in Eqs. (2) and (3):

$$\sum_{j=1}^{p} m_{ij} = 1 \qquad \text{........... (3.2)}$$

$$\sum_{i=1}^{r} m_{ij} = 0 \qquad \text{............ (3.3)}$$

A simple example of the encoding process is shown in Figure 3.1. The network consist of 6 nodes divided into two communities $v_{1,} v_3, v_4$ and$v_{2,} v_{5,} v_6$, respectively (denoted by the dotted lines). The community partition of the network is encoded by the matrix M in the right section of Figure 3.1.



$$\xrightarrow{\qquad} M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

Figure 3.1 Example of matrix encoding

Different community structures can be generated by different partitions of G, thus the number of columns M represent the number of community.

3.2.3 Population initialization

Inspired by the quantitative description of the vertex similarity of complex network (Lancichinetti, et al., 2011), we propose a new population initialization method. The perception of OBL (opposition based learning) method is founded upon a common in section that, in real life, people oppose one another. The strengths and weaknesses of these opposing ones are relative, i.e., the opposition of a weak person is strong relation to him. In OBL method, it generate a second set of solutions which is opposite of the original solution set so that our probability of choosing better solutions can increase (Rahnamayan, et al., 2008).OBL can be applied not only to the initial population but also to the each and every solution that come after iteration. However, OBL has been applied only for initialization in this work.

Let $X = (X_1, X_2, X_3, \ldots\ldots\ldots\ldots X_n)$ be a vector representing one solution of network out of the population where n is the total number of nodes of the network. So after applying opposition based method on each of n individuals of this kind, we can have an opposite solution by:

$$X_i^{'} = (A_i + B_i - X_i) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.4)$$

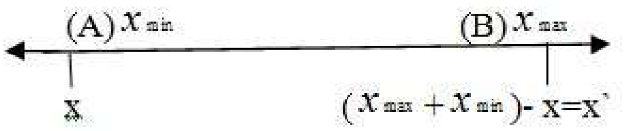


Figure 3.2. Calculating opposition of a point

In Figure 3.2, WhereA$_i$, B$_i$ represent the upper and lower values of the domain of that X_i can possess. We can say, in this particular case of evolutionary algorithms, A$_i$ is 1, that is a network can have a least of 1 community and B$_i$ is n (maximum number of nodes), that is a network can have a maximum of n communities.

3.2.4 Modified crossover operation

The crossover operation has been implemented with some modifications. Each iteration we are having a population of few individuals (here it is fixed it to 50).

Using Equation (3.1), calculation of the fitness of all individuals has been done, and the individuals are sorted in descending order according to their fitness. Let pc and pm represent the ratio of crossover and ratio of mutation similarly p_n is total number of population and Nmax is maximum number of iterations or p is node similarity. In the next step, selection of top P_n*P_c individuals possessing optimal fitness has been done and pairing them to cross. It has to be noted that Pc ($0<P_c<1$) is a fixed constant, and (P_n *P_c) mod 2= 0. As an illustrative example, let the top six individuals with optimal fitness sorted in descending order be represented by I_1, I_2, I_3, I_4, I5 and I_6. These individuals are paired for crossing. For example, the individuals may be crossed asI1 and I_6, I_2 and I_5, I_3 and I_4. After being sorted, the column in the matrix represents a community of each individual.

Measure the quality of the crossover individuals. Suppose that column $M_i = (m_{1i}, m_{2i}, ...m_{ni})^T$ contain r non zero elements, $m_{u_1 i}, m_{u_2 i}, ...m_{u_r i}$, where $1 < u_p < r, 1 < p < r, 1 < r < n,$ then use equation (3.5)to calculate the average similarity ($\overline{s_i}$) of $v_{u_1}, v_{u_1}, ..., v_{u_r}$, which also indicate the quality of gene I of the individual. In Equation (3.5), $\mathrm{S_{u_p u_q}}$ is the similarity between $\mathrm{v_{u_p}}$and $\mathrm{v_{u_q}}$[19].

$$Si = \frac{\sum_{p=1}^{r} \quad \sum_{p<q}^{r} SupUq}{\frac{r(r-1)}{2}} \qquad \text{..........................} \; (3.5)$$

Arrange each crossover individual’s genes in descending order according to $\overline{s_i}$, whose value is 0 when a community possesses only one node. Now, i_{th} individual is paired with $(i+1)_{th}$ individual for crossover. And let i_{th} individual (matrix form) is having p columns or p communities and the $(i+1)_{th}$ individual is having q columns or q communities. Each column as a gene of both the individual is sorted according to the fitness of column. Now, first few genes or columns are exchanged among each pair of individuals. And this “few” number is decided as the half of the minimum of total number of columns of both individual (Hill and Dunbar, 2003).Here, for example p is less than q so a total of first (p/2) genes or columns are exchanged.

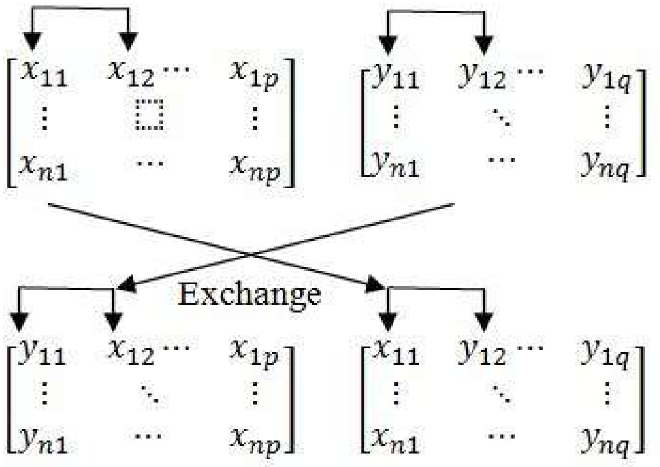


Figure 3.3. Schematic of crossover operation

If after modified crossover operation (MCO), some new individuals generated to be illegal, which means that their encoding matrixes may violate Equation (3.2) and (3.3). In this case, some nodes may not belong to a community, or may belong to more than one community; these individuals with invalid solutions are revised and discarded also. If an individual containing a column of which their entire element zeros that is that node is in none of the community and said to be invalid. If any row has more numbers of ones that is the node belongs to more than one community simultaneously then revision of the individuals has to be done, add an extra column with all zeros has to be added. Similarly contains rows whose entire are all 0 (called 0-rows), which means some nodes are not belong to any community then it is also revised with the help of add an extra column with all zeros.

3.2.5 Mutation operation

- Column of each individual (a matrix) representing a community are sorted according to the fitness value calculated using Equation 3.5 and also all the individuals (P_n given in section 3.2.5) are sorted according to the Q-function value as calculated by using Equation 3.1.
- Choosing a total of P_n*P_m (as given in section 3.2.5) bottom ranked individuals from the list of sorted individuals to perform mutation.
- Mutation will be performed on each of P_n*P_m individual in two phases, namely; Split and Fuse.
- If total no. of columns or say community in an individual is not more than two than only Split operation would be encountered otherwise both Split and Fuse operations would be done.
- Split operation involves the last column that has at least one element is split into no. of columns equal to no. of elements in that column or community such that each new column has only one element.
- Fuse operation involves, for each element or node, the neighbor most nodes is to be found and that node will then be placed in community of neighbor most nodes community (see section 3.2.5).

3.2.6 Proposed algorithm description

Input: In this algorithm, network adjacency matrix A as input variable and the parameters of MCOBGA given in Table 1.

Output: Encoding matrix M representing the community partition of a complex network.

Terminal condition: The algorithm runs through N_{max} iterations.

Parameter	Value	Description
P	0.97	Parameter used to calculate the node similarity
P_n	100	Number of individuals in population
P_c	0.8	Ratio of crossover individuals to total no. of individuals of population
P_m	0.2	Ration of mutation individuals to total no. of individuals of population
N_{max}	50	Maximum number of iterations

Table 3.1: Parameters of MCOBGA

Algorithm pseudo-code:

```
Step-1: Use PIVOB to generate initial population $P_n$ (section 3.2.1)
fori=1:Nmax
Step-2: Calculate the fitness of each individual (Q-value) using Eq. (3.1), and sort individuals in descending fitness order; maintain space for extra $P_n$ ($P_c$+$P_m$) individuals.
Step-3: Select the fittest $P_n$*$P_c$ individuals, and calculate the quality of each community represented by a column in matrix using Eq-3.5: sort the communities according to the quality of each column of each crossover individual in order of descending quality;
Step-4: Pair crossover individuals and exchange their columns (section 3.2.4);
Step-5: Revise invalid individuals generated by the crossover operation (section 3.2.4);
Step-6: Select the least fit $P_n$*$P_m$ individuals to mutate;
Step-7: Allow mutation individuals to mutate non uniformly (Section 3.2.5); Following completion of Steps 4to 8, a new population is obtained $P_{new}$;
Step-8: Compute the fitness of all individuals in $P_{new}$(Eq-1), integrate $P_{new}$ and $P_{original}$, select the top $P_n$ individuals for next iteration;
i=i+1;
END
Step-9: Select the maximally fitted individual as the final result.
```

3.3 Experimental Description

The algorithms discussed so far have been tested by a series of experiments. The experiment was conducted on Microsoft Windows 7 professional operating system using a MATLAB 11 programming platform with Intel (R) Dual-Core 2.50 GHz processor and 4.0 GB RAM. The values of different parameters P_n, P_c, Pm and Nmax like were fixed through a series experiments for which proposed algorithm in the given scenario works at best. Parameter values so set has been given in Table 1. It has to be noted that the values of Pn, Pc, Pm and Nmax could be altered as appropriate for the situation. Performance of MCOBGA is tested on four real networks, and is compared with modified Genetic Algorithm (referred as SGA) (Pizzuti, 2008). While evolving to MCOBGA from SGA, an intermediate algorithm, modified genetic algorithm with OBL initialization (referred as OBGA) had been proposed. Results from MCOBGA and SGA have also been compared with OBGA. The four networks used for testing are Strike (Michael, 1997), Zachary's karate club (Zachary, 1977),Dolphin sociality (Lusseau, et al., 2003) and American College Football (Girvan and Newman, 2002). The specifications of each dataset have been summarized in Table 3.2.

Dataset name	Number of nodes	Number of edges	Number of communities
Strike	24	34	3
Zachary’s karate club	34	78	2
Dolphin sociality	62	159	2
American College Football	115	613	12

Table 3.2: Specifications of real world datasets used in experiments

3.3.1 Experimental Analysis

Results obtained with given parameter (Table 3.1), all data sets (Table 3.2) and SGA and MCOBGA have been shown in Figure 3.4 to Figure 3.7. Convergence of fitness function, Q, has been taken as metrics of evaluation of algorithms. The patterns of fitness function Q for OBGA have not been shown. However, pattern remains same for OBGA compared to MCOBGA. As evident from the Figure 3.4-3.7, MCOBGA outperforms SGA for given 50 iterations. In Figure 3.4-3.7, MCOBGA represent magenta colour and SGA for green colour line.

Table 3.3 represents the different values of fitness function for SGA, OBGA and MCOBGA over different datasets. It is evident from the values that MCOBGA out performs SGA and OBGA for all dataset. For Strike dataset, increase in modularity for MCOBGA has been almost 1% for both SGA and OBGA respectively. For Karate dataset, increase in modularity for MCOBGA has been almost 6% and 5% for SGA and OBGA respectively. For dolphin dataset, increase in modularity for MCOBGA has been almost 4% and3% for SGA and OBGA respectively. For football club dataset, increase in modularity for MCOBGA has been almost 12% and 10% for SGA and OBGA respectively.

Apart from improvement in the fitness value, faster stability or convergence of MCOBGA has been observed for all data sets. Convergence of SGA occurred at 15th iteration for karate’s club dataset whereas MCOBGA converged at 10th iteration. Similarly, for Strike dataset SGA converged at 10thiteration and MCOBGA at 4th iteration, for Dolphin dataset it has been 20th and 15th iteration for SGA and MCOBGA respectively. For football dataset which is the largest one used here, stability in SGA occurs at around 12th iteration whereas in

MCOBGA firstly stability occurs at approximately 8th iteration and then it improves it value again at around 30th iteration.

It can be inferred from observations of Figure 3.4-3.7and Table 3.3 that the percentage increment in values of function modularity has been largest for Football dataset that is the proposed algorithm can be used at its best for large dataset with faster convergence rate.

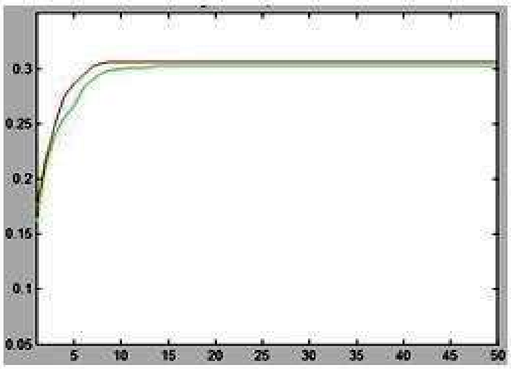


Figure 3.4 Q-fun value(on Y axis) vs no. of iteration (on X axis) for for Comparision between SGA and MCOBGA For strike dataset.

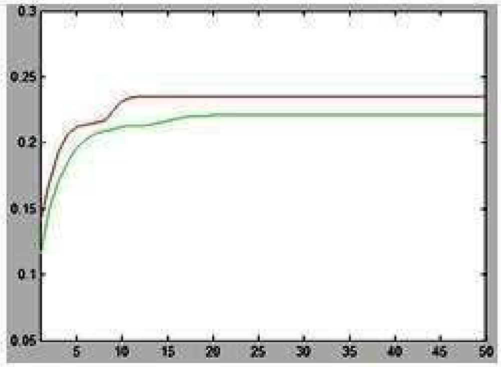


Figure 3.5 Q-fun value(on Y axis) vs no. of iteration (on X axis) Comparision between SGA and MCOBGA for karate club dataset.

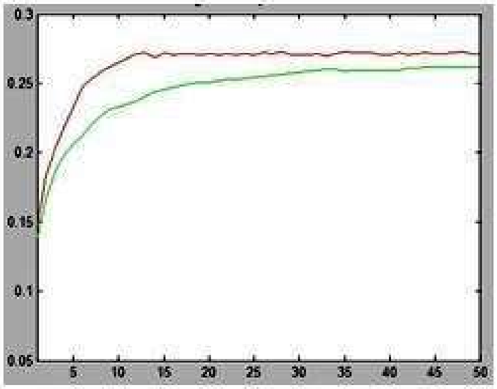


Figure 3.6 Q-fun value(on Y axis) vs no. of iteration (on X axis) for for comparision between SGA and MCOBGA for Dolphin sociality dataset

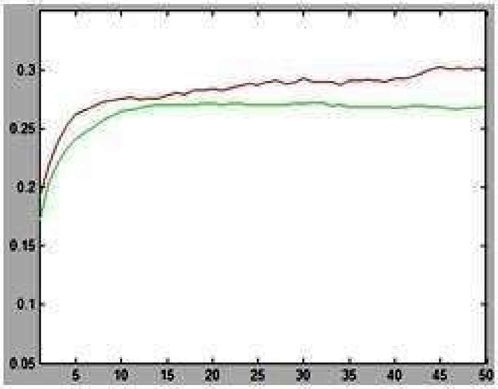


Figure 3.7 Q-fun value(on Y axis) vs no. of iteration (on X axis) comparision between SGA and MCOBGA for American college football dataset.

Dataset name	Q-fun value of SGA	Q-fun value of OBGA	Q-fun value of MCOBGA
Strike dataset	0.3025	0.3025	0.3056
Karate's club	0.2210	0.2225	0.2348
Dolphin's dataset	0.2608	0.2620	0.2717
Football's dataset	0.2683	0.2741	0.3017

Table 3.3: Average Q-function values of SGA, OBGA and MCOBGA running on 4 real social networks

3.3.2 Accuracy and Quality Measure

The communities so obtained by different algorithms have been evaluated for quality and accuracy. For evaluation of the communities based on quality and accuracy, different metrics listed in has been taken into account. Accuracy of communities are more important than quality related matters. Therefore, evaluation methods used in this work have been more inclined towards realization and evaluation of accuracy of outcomes. NMI, ARI and F-measure are metrics used for accuracy whereas Modularity (Newman, 2006) has been used to measure quality (Biswas and Biswas, 2017). Further, Multiple Criterion Decision Making (MCDM) based ranking have been performed. The advantage of MCDM ranking is that it accumulates all accuracy metrics and quality metrics under one single score. For MCDM ranking, TOPSIS method (Boran, et al., 2009) have been utilized. TOPSIS method can assign weights to each of the metric where summation of all weights assigned to different metrics has to be 1. The weightage assigned to each metric depends on the priority of that metric. In this work, concern has been to gain more accuracy in communities, so 75% weightage has been assigned to accuracy metrics and 25% weightage are assigned to quality metrics. As in, weightage of both measures has been distributed equally among the metrics in that category. To measure accuracy, three metrics (NMI, ARI and F-measure) has been considered and to measure quality, only one metric (Modularity) have been deployed. So 75% weightage assigned to accuracy have been distributed among three metrics assigning each metric with 25% weightage. Modularity as only metric for quality has been assigned 25% weightage which have been for quality measurement.

Datasets	Algorithms	NMI	ARI	Modularity	F-measure
Dolphin	SGA	0.6022	0.5601	0.2608	0.4146
	OBGA	**0.6024**	0.57	0.262	0.5
	MCOBGA	0.5112	0.57	**0.2717**	0.4028
American football	SGA	0.7982	0.2983	0.2755	0.1954
	OBGA	**0.82**	0.3026	0.2741	0.1954
	MCOBGA	0.7103	0.23	**0.3015**	0.1781
Karate club	SGA	0.8041	0.7414	0.221	0.5556
	OBGA	**0.8041**	0.7508	0.222	0.5556
	MCOBGA	0.7103	0.5998	**0.2225**	0.2031
Strike	SGA	0.8232	0.8428	0.3025	0.7514
	OBGA	**0.8314**	0.8428	0.3025	0.7514
	MCOBGA	0.785	0.7811	**0.3056**	0.5051

Table 3.4: Accuracy and quality metric values in various datasets whose ground truth communities are known.

Table 3.4 shows the results in terms of accuracy and quality for all datasets and all algorithms. Clearly, for Dolphin data set OBGA shows higher NMI, ARI and F-measure than all other competitors. MCOBGA also shows higher Modularity, ARI and F-measure than other algorithms, but lagging behind OBGA. Whereas, MCOBGA shows highest Modularity value. OBGA is slightly behind MCOBGA. Though, OBGA shows that all accuracy metrics (NMI, ARI, F-measure) are gain the highest values compare to quality metric modularity. To summarize OBGA outperforms in accuracy measure whereas MCOBGA shows best results for quality measures. It has to be noted that though MCOBGA has not shown best results in terms of accuracy, still it has been outperformed SGA and has very competitive performance with OBGA.

Figure 3.8 - 3.11 and Table 3.6 shows MCDM rankings obtained corresponding to communities predicted by different algorithms in each data set. As we have allocated more weights to accuracy metrics so scores obtained indicate algorithm's inclination towards accuracy. OBGA shows higher MCDM rank for small datasets (karate & strike) and MCOBGA shows higher scores for big data sets (Dolphin & Football).

Datasets	**NMI**	**ARI**	**Modularity**	**F-Measure**
Dolphin	OBGA	OBGA	MCOBGA	OBGA
American Football	OBGA	OBGA	MCOBGA	OBGA
Karate club	OBGA	OBGA	MCOBGA	OBGA
Strike	OBGA	OBGA	MCOBGA	OBGA

Table 3.5: Accuracy and quality based performance for different algorithms

Datasets	**Algorithms**	**MCDM Rank**
Dolphin	SGA	0.43
	OBGA	0.48
	MCOBGA	**0.5**
American Football	SGA	0.28
	OBGA	0.33
	MCOBGA	**0.69**
Karate club	SGA	0.65
	OBGA	**0.68**
	MCOBGA	0.32
Strike	SGA	0.62
	OBGA	**0.63**
	MCOBGA	0.38

Table 3.6: MCDM ranking score obtained with 75% accuracy and 25% quality. Higher score indicates more inclination of algorithm towards accuracy.

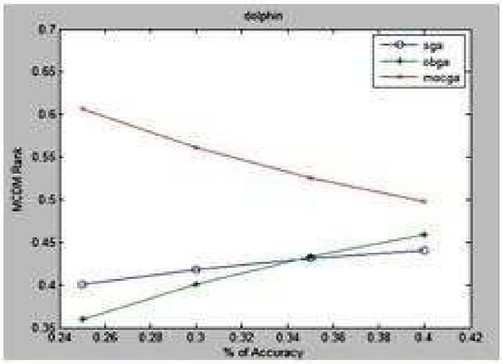


Figure 3.8 MCDM ranking acquried by each algorithm in Real world known network for Dolphin dataset with Variation of accuracy contribution

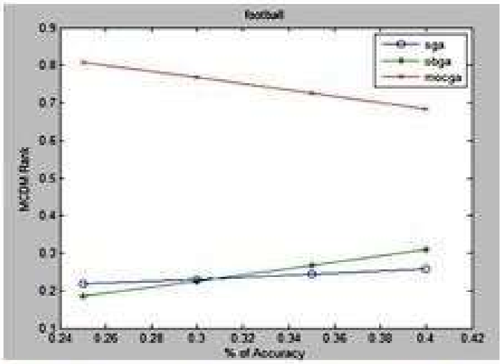


Figure 3.9 MCDM ranking acquried by each algorithm in Real world known network for Football dataset with Variation of accuracy contribution

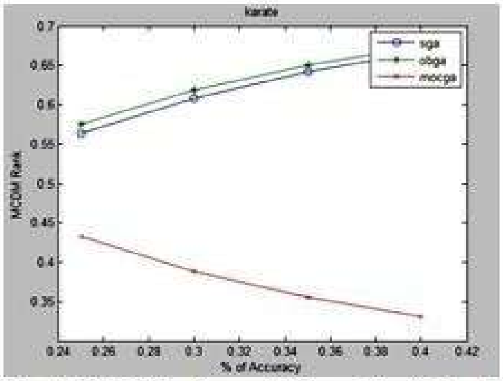


Figure 3.10 MCDM ranking acquried by each algorithm in Real world known network for Karate dataset with Variation of accuracy contribution

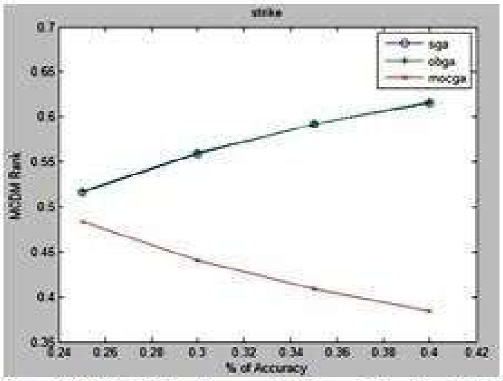


Figure 3.11 MCDM ranking acquried by each algorithm in Real world known network for Strike dataset with Variation of accuracy contribution

After this experiment, we have found that the quality and accuracy of the communities are not very well and the problem is the initialization phase and slow convergence rate of the Genetic algorithm. So-that in future scope of this experiment, we employed the regenerative genetic algorithm for community detection in social network.

Finally we proposed a new generation of genetic algorithm for the optimization technique. We employed that method for the community detection in social network. The detail description is given below and pseudo code is also given below.

3.4 Regenerative Genetic Algorithm

In our algorithm, we set some parameters which are (size, gens, Pc, Pm).where size refers to the total population size. Gens represent the running generation of the community, Pc represents the rate of crossover and Pm represents the mutation rate. Pc and Pm ranges between 0 to 1, such that Pc + Pm=1. Now we consider a solution set Pt and then we represent each individual in the form of gi, where range from 1 to size. After this OBL is compared as complement of gi now the evaluation of each individual (gi) is done i.e. fitness value of each individual is calculated. Then the generated individuals are included in the solution set Pt. now these individuals are sorted according to their fitness values. Now we consider each individual from the current generation and we randomly pick two individuals (gj, gk),let r be their index in Pt and if r value is less than Pc the crossover operation takes place else regeneration or re-initiation operation excluding mutation as random search. Now we evaluate the fitness of newly generated individuals (gj, gk), in this process the generated two individuals are replaced in the solution set. Now the fittest of the both the newly generated individuals and the previous individuals are selected and now these obtained individuals are sorted according to their fitness value and the process is repeated till we get the optimized individual from which we get the partition of network into communities.

3.5 Result and Discussion

3.5.1 Experimental Description

Our present algorithm is tested on some real life data sets (networks) which are Zachary's karate club, Dolphin sociality, American college football, Jazz Musician, Books of US Politics and compared with the algorithms GN (Girvan and Newman, 2001) and FN (Newman, 2004), TGA (Gog, et al., 2007), SGA (Li, et al., 2013).

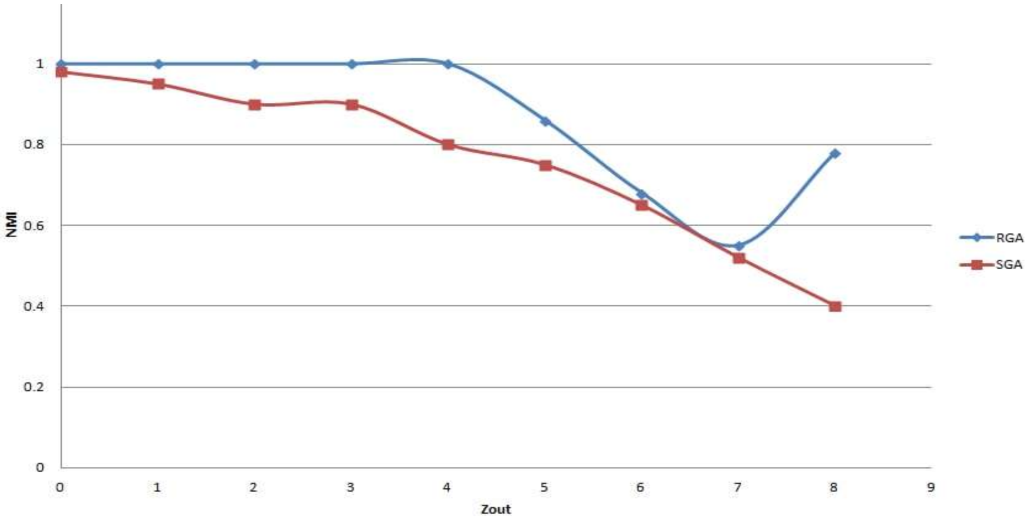


Figure 3.12: Accuracy of SGA and RGA running on artificial random networks

In order to, check the ability of our approach to successfully detect the community structure of a network. We tested our algorithm on a random artificial network and we studied the NMI with respective to the Z_{out} values (Duch and Arenas, 2005) and we graphically analyzed our results and compared it with the SGA algorithm. We got better NMI values with respective to Z_{out} ranging from (0-0.8).the graph of (NMI values Vs. Z_{out} values) (Tang, et al., 2016) of the both RGA and SGA algorithms is given below.

3.5.2 Experimental Analysis

3.5.2.1 Zachary karate club

It's a network related to a karate club in American university consists of 38 nodes (which represents the club members) and 78 edges (indicates the social connection between the club members) between them. We ran this algorithm 5 times on this data set and average modularity function values (Q), Normalized mutual information and number of partitioned communities are given in the following table.

S No	Average Q Value	NMI Similarity	No Of Communities
1	0.4193	0.6873	4
2	0.4198	0.6873	4
3	0.3964	0.6873	4
4	0.4198	0.6873	4
5	0.4188	0.6873	4

Table 3.7: Average Q-values and NMI-values for karate club dataset

After 5 sequential runs of our algorithm the fitness value which determines the quality of community partition is encountered to be in the range of 0.3-0.42.The highest fitness value is 0.4198, which was found out on 2nd and 4th run of our algorithm and the least is found out be 0.3964.The normalized NMI which is similarity measure determining the accuracy of the partition of network is found to be 0.6873 and the number of communities detected are 4.

Our algorithm produces 50 new generations of solution members on each run and average fitness value of each generation of solution members is calibrated and related variance of average fitness values corresponding to their iterations is compared with SGA and graphically reported below.

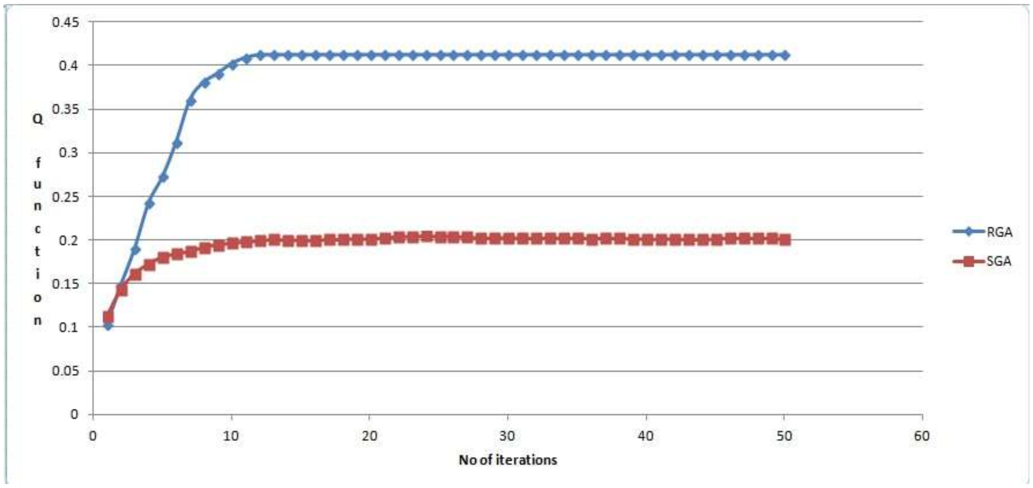


Figure 3.13: Q-value of SGA and RGA running on karate club dataset

From the graph mentioned above, in RGA the Q value is exponentially increasing from 0-10 iterations and from 10-50 iterations q value is almost same i.e. the curve is almost parallel and the highest value near 0.42.in contrast with SGA ,here the Q value is increasing from 0-10 iteration but not much as RGA and from 10-50 iterations variation is of Q value very less (almost same) like RGA but the highest Q value is found out to be 0.2 which is very less as compared to our algorithm (RGA).

3.5.2.2 Dolphin Sociality

This network is a graphical representation of the contacts of dolphins between male and female communities. In this network nodes represents dolphins and edge indicates that two dolphins met frequently. The network consists of 62 nodes (dolphins) and 159 edges (interactions) between them; the same procedure is done as of previous network's (Zachary's karate club), a table of average q values, NMI values, number of communities of dolphin's sociality is given below.

S.No.	Average Q value	NMI Similarity	No. of communities
1	0.5411	0.4807	7
2	0.5489	0.5987	4
3	0.5080	0.552	4
4	0.5269	0.537	5
5	0.5433	0.5823	5

Table 3.8: Average Q-values and NMI-values for Dolphin dataset

After 5 sequential runs of our algorithm on this network we obtained different results at different runs. In the 2ndrun we found out the highest Q value (0.5489) and highest NMI similarity (0.5897) and with the network partitioning into 4 communities. Least fitness value (0.5080) is found on 3rd run with NMI similarity 0.552 and network partitioning in to 4 communities. In the 1strun we see the partitioning of the network into 7 structures of communities with quality of 0.5411 i.e. modularity value, and NMI similarity being 0.4807.

We also observed the change of fitness function with no of iterations made on solution members and compared the results with the SGA algorithm and graphical representation of Q vs no. of iterations made of our RGA and SGA algorithms respectively is shown below.

From the graph we notice that in our algorithm (SGA) the Q is exponentially increasing on first 10-12 iterations and very little change or almost same Q value is found up to 50 iterations. And the highest Q value is found as 0.5.on the contrast in SGA the increase in Q value is very less as compared to RGA in starting 10 iterations and the q remained almost constant up to 50 iterations. And the highest Q value is found out to be 0.2.so clearly we observe that our algorithm produces best quality partitioning of community structure as compared to SGA.

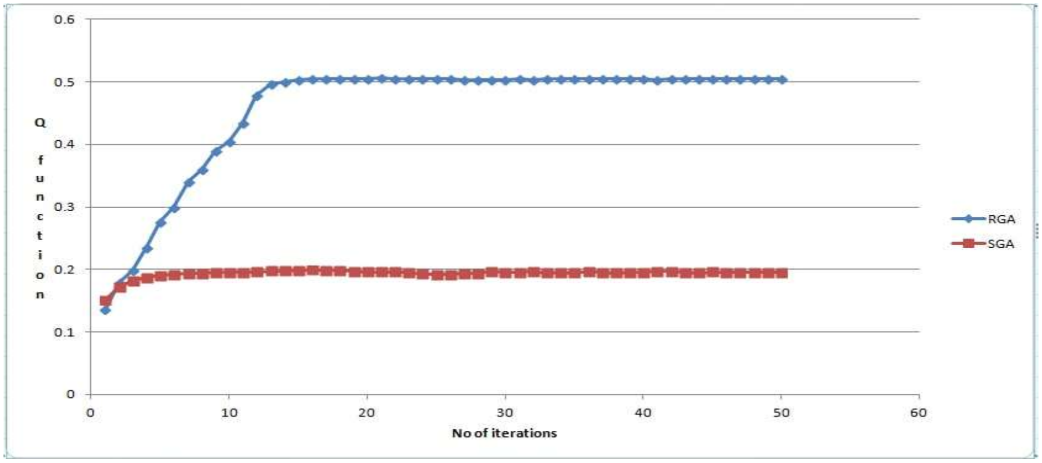


Figure 3.14: Q-value of SGA and RGA running on Dolphin dataset

3.5.2.3 American College Football

This network is graphical representations of American football teams as nodes and edges are treated as match between them i.e. if a match is held between two teams then we say that there is an edge between them. This network comprises of 115 nodes (teams) and 613 edges (total number of matches held).Same procedure is done as of previous networks, a table of average q values, NMI values, number of communities of American college football is given below.

S No	Average Q Value	NMI Similarity	No Of Communities
1	0.55261	0.8073	10
2	0.5819	0.8376	10
3	0.5751	0.7809	8
4	0.57107	0.902	14
5	0.5448	0.7865	9

Table 3.9: Average Q-values and NMI-values for American College Football dataset

After 5 sequential runs of our algorithm on this network we obtained different results at different runs. In the 2ndrun we found out the highest Q value (0.5819) and NMI similarity (0.5897) and with the network partitioning into 10 communities. Least fitness value (0.5448) is found on 5th run with NMI similarity 0.7865 and network partitioning in to 9 communities. We see the highest accuracy of partitioning on 4th run i.e. NMI similarity being 0.902 and splitting the network into 14 different community structures with average fitness of 0.57107(Q).

We also observed the change of fitness function with no of iterations made on solution members and compared the results with the SGA algorithm and graphical representation of Q vs. no of iterations made of our RGA and SGA algorithms respectively is shown below.

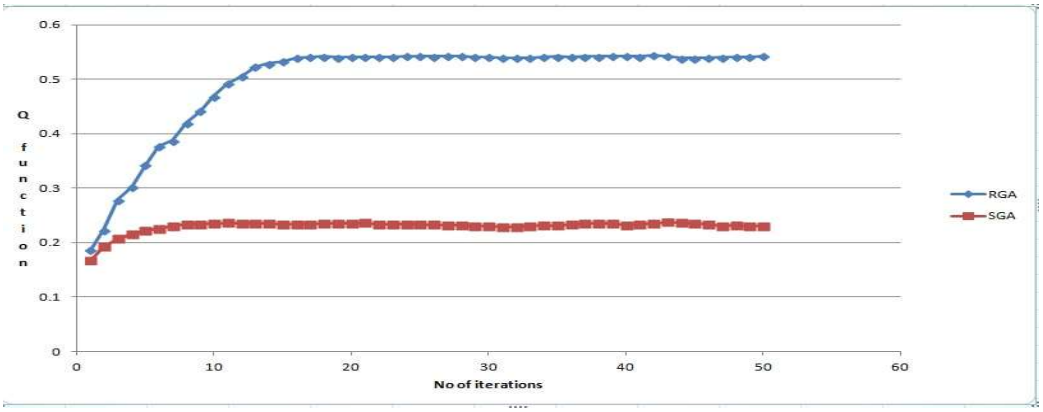


Figure 3.15: Q-value of SGA and RGA running on American College Football dataset

From the graph we notice that in our algorithm (SGA) the Q is exponentially increasing on first 10-12 iterations and very little change or almost same Q value is found up to 50 iterations. And the highest Q value is found as 0.54.on the contrast in SGA the increase in Q value is very less as compared to RGA in starting 10 iterations and the q remained almost constant up to 50 iterations. And the highest Q value is found out to be 0.2.so clearly we observe that our algorithm produces best quality partitioning of community structure as compared to SGA.

3.5.2.4 Books on US Politics

This is network of political books where the nodes represent 105 recent books on us politics whereas edges refer to the purchase of books by same buyer .i.e. If two books are purchased by same buyer then there is an edge between those books. There are total 441 edges in this network; same procedure is done as of previous networks, table of average q values, NMI values; number of communities of books on US politics is given below (Bagrow and Bollt, 2005).

After 5 sequential runs of our algorithm on this network we obtained different results at different runs. In the 2nd run we found out the highest Q value (0.5489) and highest NMI similarity (0.5897) and with the network partitioning into 4 communities. Least fitness value (0.5080) is found on 3rd run with NMI similarity 0.552 and network partitioning in to 4 communities. In the 1strun we see the partitioning of the network into 7 structures of communities with quality of 0.5411 i.e. modularity value, and NMI similarity being 0.4807.

S No	Average Q Value	NMI Similarity	No Of Communities
1	0.50967	0.8253	3
2	0.5034	0,8578	4
3	0.50212	0.7956	4
4	0.50967	0.8867	5
5	0.5087	0,8673	4

Table 3.10: Average Q-values and NMI-values for Books on US Politics dataset

We also observed the change of fitness function with no of iterations made on solution members and compared the results with the SGA algorithm and graphical representation of Q v_s no of iterations made of our RGA and SGA algorithms respectively is shown below.

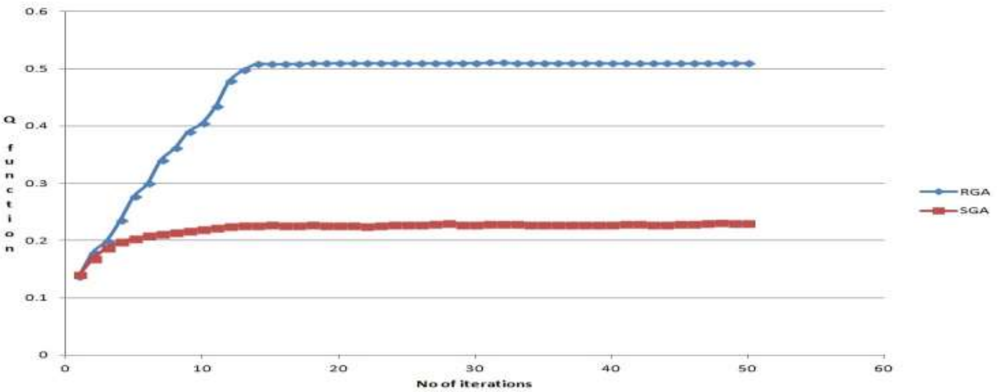


Figure 3.16: Q-value of SGA and RGA running on Books on US Politics dataset

From the graph we notice that in our algorithm (SGA) the Q is exponentially increasing on first 10-12 iterations and very little change or almost same Q value is found up to 50 iterations. And the highest Q value is found as 0.5.on the contrast in SGA the increase in Q value is very less as compared to RGA in starting 10 iterations and the q remained almost constant up to 50 iterations. And the highest Q value is found out to be 0.2.so clearly we observe that our algorithm produces best quality partitioning of community structure as compared to SGA.

DATASET	SGA	FN	GN	TGA	RGA
Zachary's karate club	0.195	0.252 8	0.4013	0.4039	**0.4198**
Dolphin sociality	0.195	0.371 5	0.470	0.5341	**0.5489**
Books of US Politics	0.2895	0.5020	0.5168	0.5245	0.50967
American college football	0.2311	0.454 9	0.599	0.5937	0.55523

Table 3.11: Average Q-values of SGA, FN, RGA, GN and TGA running on and for four dataset

In this given table, we have defined the comparative result of the proposed algorithm and the other existing algorithms. In given table represent the average modularity value of all the algorithms. We show that RGA is outperforming the excellent result for the karate club and the dolphin sociality datasets. RGA is also the good performance for remaining datasets like as American football and books on us politics. The accuracy of the RGA is superior to that of some traditional algorithms and is similar to that of some recent high-precision algorithms.

3.6 Conclusion of the Chapter

In this chapter, we have introduced genetic algorithm to detect community structure in social network using some improvisation. To best of our knowledge, it is the first time GA with OBL (opposition based learning) has been applied to community detection problems. The proposed algorithm MCOBGA uses GA to search the best network partition of a social network that can achieve an optimal network modularity value. Based on opposition learning initialization of GA, we designed a modified single point crossover to transmit some important information about the community structure during evolution. We have also first time introduced OBL with vertex similarity initialization process to improve the quality of the individuals in the population .We have tested our MCOBGA on real world social networks in comparison with simple GA (SGA) and opposition based (OBGA) algorithms. MCOBGA requires no additional optimizing steps. It adopts matrix encoding, uses OBL with vertex similarity to initialize the population and conducts simple (single point) crossover and mutation operations. The experimental results have demonstrated that MCOBGA is very effective for community detection in social networks. The two major conclusions drawn out of the experimental results have been: 1. higher convergence rate of MCOBGA especially on bigger size data network; 2. Accuracy and quality of communities obtained by MCOBGA is better in bigger size data networks. Therefore, it is a new and effective genetic algorithm for community detection in social networks especially when applied to big networks.

Another experiment after MCOBGA, Regeneration is used instead of mutation as it not only maintains the diversity of the population but improves quality of genes of the individual increases the percentage of selection for next iteration. Split runtime is done to allow the solution to converge quickly and efficiently improving the modularity. The simplicity and efficiency of the algorithm are uncovered in experimental tests using artificial random networks and real-world dataset.