Journal of Theoretical and Applied Information Technology
15th March 2024. Vol.102. No 5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A MULTIOBJECTIVE EVOLUTIONARY ALGORITHM WITH A NOVEL MUTATION OPERATOR FOR OVERLAPPING COMMUNITY DETECTION

A.C. RAMESH [1] , G. SRIVATSUN [2]

[1]Assistant Professor (Selection Grade), Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, Tamilnadu, India.

[2]Associate Professor, Department of Electronics and Communication Engineering, PSG College of Technology, Coimbatore, Tamilnadu, India.

E-mail: [1]acr.cse@psgtech.ac.in, [2]gsn.ece@ psgtech.ac.in

ABSTRACT

The detection of overlapping communities in complex real-world networks is a difficult problem that is being addressed by different methods. The multiobjective evolutionary algorithm (MOEA) is a promising alternative that has shown competitive performance in this research over the past two decades. The representation scheme used by the MOEA affects the quality of the solutions obtained and the runtime of the algorithm. The length of the chromosome is significantly reduced when cliques are used as genes instead of the original nodes of the graph. The execution time of the evolutionary algorithm (EA) also depends on the combined execution times of the evolutionary operators, crossover and mutation. This paper proposes a novel mutation operator that uses community labels of cliques as genes rather than cliques. The proposed mutation operator results in fewer modifications on the chromosome than does the existing clique-based mutation. Experiments conducted on real-world and synthetic networks reveal that the proposed algorithm produces good community partitions of the network when compared with existing clique-based algorithms and state-of-the-art community detection algorithms.

Keywords: *Community-Based Mutation, Maximal Clique, Quantile, Overlapping Community Detection, Multiobjective Evolutionary Algorithm.*

1. INTRODUCTION

Many real-world applications, such as protein complexes, social network data, drug target interactions and metabolism mechanisms, are represented in terms of complex networks. These networks are modeled with objects as vertices and complex phenomena occurring in physical, social and biological systems as links connecting the vertices. Communities in a complex network are groups of closely associated objects and their interactions, as discussed in the seminal paper [1]. They represent groups of proteins in biological networks, different areas in brain networks, authors in collaboration networks, etc. Graphs are used to represent the objects in a network as nodes and their interactions as edges. The detection of groups in a network is equivalent to detecting communities in graphs. Some of the communities also have nodes that may belong to other communities. The detection of such overlapping communities can be accomplished by enumerating dense substructures, such as near-cliques, bicliques, and cliques.

A maximal clique in a graph is a clique formed by a subset of nodes that are completely connected. The maximal cliques are being employed in real-world community detection applications, such as finding driver microbes involved in virus infection [2]. The CFinder algorithm [3]uses the clique percolation method to find k-cliques containing small groups with high density. An improved solution using the fuzzy method termed the pseudoclique extension [4] was proposed for finding maximal cliques in protein complexes. An alternative solution has been implemented using formal concept analysis [5] for k-clique community detection in social networks.

Engineering optimization techniques model the community detection problem where single or multiple objective functions are optimized. The optimization methods initially aimed at optimizing a single objective function that optimizes modularity

[6] and has a disadvantage in terms of resolution limit [7] Therefore, the single objective function needs to be designed effectively since the algorithm solely depends on the function and the failure of which will produce undesirable results. Alternatively, multiobjective optimization algorithms have greater flexibility in accommodating conflicting objectives, where one objective minimizes the strength of connections between communities and the other maximizes the strength of connections within a community [8].

The multiobjective optimization algorithm based on decomposition framework (MOEA/D) [9] has gained attention as an important framework owing to its low computational complexity when compared with NSGA-II. The representation scheme of a chromosome plays a vital role in MOEAs, affecting its runtime and the effectiveness of evolutionary operators. The representation scheme generally falls into one of three types, namely, node-based, prototype/community-based or clique-based [10]. The prototype-based representation considers community prototypes such as centroids and medoids as the genes in the chromosome. This approach transforms the community detection problem into a clustering problem, creating spherically shaped communities [11]. The node-based representation has direct schemes [12] that are suited for nonoverlapping community detection. On the other hand, [13] proposed an indirect scheme for overlapping community representation in which decoding is performed to determine the community of a node. The clique-based representation introduced in MCMOEA [10] has the maximal cliques as genes in the chromosome. The cliques can share nodes, making them naturally fit for overlapping community detection. The encoding scheme of MEMOEA [14] is also based on cliques that further reduce the size of a chromosome, thereby reducing the dimensions of the solution search space.

The crossover and mutation operators are as important as the representation scheme because their execution time affects the runtime of the evolutionary algorithm. Six different crossover operators have been studied in different MOEA frameworks [15], and the authors concluded that crossover parameters can be fixed and do not change much with respect to each framework. The mutation operator in MOEA/D is adaptively changed by [16], which improves the population diversity and the performance of the MOEA. The MOEA/D for crowd sensing [17] uses a problem-specific mutation operator rather than a random number. The mutation operator in the existing clique-based MOEAs consumes more computation time owing to its application on every clique node in the chromosome [10] [14]. The representation scheme of the MEMOEA has the chromosome as a solution with merged-maximal cliques as genes. For large networks, the number of merged maximal cliques can be large, but not as many as the basic maximal cliques of MCMOEA [10].

In accordance with the above observation, a merged-maximal-clique-based multiobjective evolutionary algorithm named (MEMOEA-m) is proposed with the following improvements:

- An alternate representation scheme where each community is represented as a gene in the chromosome.
- The new mutation operator considerably reduces the computation time of the existing algorithm.
- The MEMOEA-m algorithm is evaluated on real and synthetic networks and compared with clique-based, non-clique-based and state-of-the-art algorithms. The results show that the MEMOEA-m algorithm is better than the other algorithms in most of the networks and gets closer to the MEMOEA algorithm despite the smaller changes made to the chromosome during evolution.

The paper is organized as follows: Section 2 presents the methodology used in the work. Section 3 explains the experimental setting, datasets used, and performance metrics employed and discusses the performance of MEMOEA-m. Section 4 concludes the work with future research directions.

2. METHODOLOGY

The flow diagram of the MEMOEA-m is shown in Fig. 1. The maximal cliques are generated from the graph, which represents the network and are recursively merged to produce merged-maximal cliques. A clique graph is constructed with the merged cliques as nodes, and the links between them are filtered by an appropriate quantile value of the link strength distribution. Then, the multiobjective evolutionary algorithm is applied to the chromosomes created from merged-maximal cliques.

Evolutionary operators such as one-way crossover and community-based mutation are executed in a single run of the evolutionary algorithm to produce the Pareto optimal solutions. After a fixed number of runs, the best among the optimal solutions is output by the algorithm

2.1 Graph representation

A network is represented by a graph G with N nodes and E edges. The edges of G are unweighted and undirected. Communities are defined in a weaker sense with the condition "total internal connections of all the communities should be greater than their total external connections" assumed in this work [13].

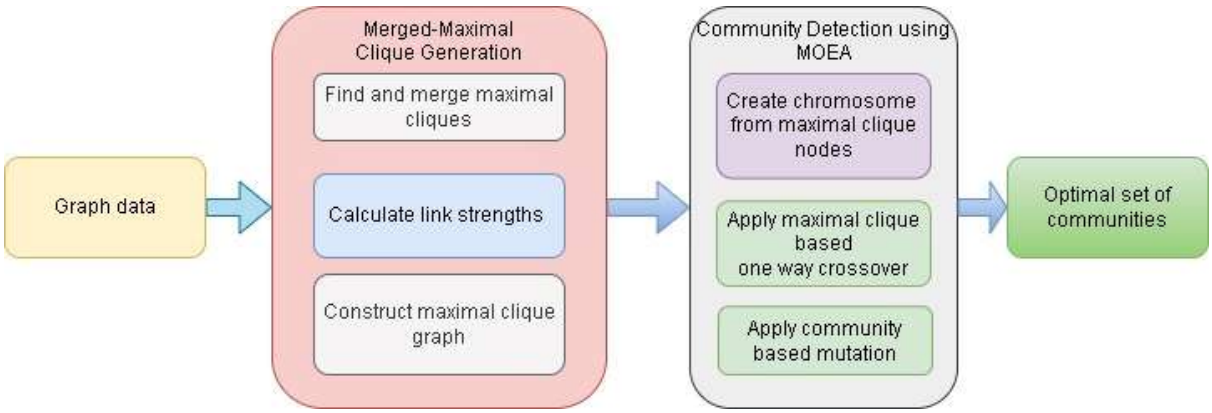

Fig. 1 MEMOEA-m flow diagram

Every node in the network belongs to one or more communities, as implied in equation 1.

$$V = \bigcup_{i=1}^{K} C_i \quad (1)$$

where $C_i \cap C_j = \emptyset$; if the communities are strictly nonoverlapping, $C_i \cap C_j \neq \emptyset$; if the communities are strictly overlapping, K is the total number of communities.

2.2 Merged-maximal clique generation

Maximal cliques are generated for each node of the graph [18] and are recursively merged using the algorithm in Fig. 2. (Line 1), and the process is summarized as follows. The nodes are arranged in increasing order of their degree frequency. The maximal cliques are generated for each node and sorted according to their size. The maximum-sized maximal cliques of the node are recursively merged. The nodes used for finding cliques are removed from the sorted list. Since the order of processing nodes is based on the degree frequency, higher-order nodes, usually fewer in number, are used to find cliques at the end.

2.3 Link strength calculation

The link strength between every pair of merged-maximal cliques is determined next to construct the clique graph (Line 2, Fig. 2). The link strength is the weighted sum of three types of interconnections viz. overlapping nodes (ONs), overlapping edges (OEs) and joint edges (JEs). The link strengths for each of these interconnections l_{on}, l_{oe} and l_{je} are determined as in MEMOEA.

The total link strength (L) of each pair of cliques is calculated from the three interconnection link strengths as given in equation (2).

$$L(v_m^{me}, v_n^{me}) = \alpha * l_{on}(v_m^{me}, v_n^{me}) + \beta * l_{oe}(v_m^{me}, v_n^{me}) + \gamma * l_{je}(v_m^{me}, v_n^{me}) \quad (2)$$

$$\alpha = \frac{count\ of\ ONs}{total\ count}, \beta = \frac{count\ of\ OEs}{total\ count}, \gamma = \frac{count\ of\ JEs}{total\ count}$$

$$total\ count = sum(ONs, OEs, JEs)$$

$$\alpha + \beta + \gamma = 1$$

where the counts of each type of link are calculated, and their proportionate contributions (α, β and γ), which are introduced in this work, are computed.

The link strengths are not normalized since proportionality multipliers are used. Different quantiles, such as quartiles (Q1=0.25, Q2=0.5, Q3=0.75), deciles (Q4=0.80, and Q6=0.90), and ventile Q5=0.85), are used for severing the weaker links. The appropriate quantile (Q) threshold for a specific network can be fixed (Line 3, Fig. 2). Lower quantile values such as Q1 and Q2 are used for less complex networks that have clear community structures. The higher quantile values from Q3 and above are used in networks where community structures are hidden by date hub nodes or by dense interconnections. The hub nodes can be either date hubs with membership in multiple communities or party hubs [19], which are central nodes in a community.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm MEMOEA-m

Input:

- Original graph G = (V, E)
- gen_{max}: the maximum number of generations
- T: the size of neighbourhood of each weight vector
- PS: the size of population
- $\lambda^1, \lambda^2, \ldots, \lambda^{PS}$: a uniform spread of PS weight vectors

Output:

NS: the set of non-dominated solutions

I. **Construction of Merged-maximal-clique graph G^{me}**

1. Determine the set of merged-maximal-clique nodes V^{me}
2. Measure the link strength between merged-maximal-clique nodes.
3. Reduce the number of edges E^{me} between merged-maximal-clique nodes based on link strength using appropriate quantile value.

II. **Initialization:**

4. Initialize the population $P = \{I_1, I_2, \ldots, I_{PS}\}$, where each individual $I_i = \{g_{i1}, g_{i1}, \ldots, g_{iM}\}$ represents the current solution to the i-th subproblem.
5. Renumber the individuals in the population.
6. Initialize the neighbourhood of each weight vector.
    1. For each I = 1, 2, ..., PS, set B(i) = $\{i_1, i_2, \ldots, i_T\}$, where $\lambda^1, \lambda^2, \ldots, \lambda^{PS}$ are the closest weight vectors to λ^i in the Euclidean space.
7. Initialize z^* as $(KKM(I_1), RC(I_1))$
8. Initialize NS as an empty set.

III. **Population Evolution**

9. for g = 1 to gen_{max}

1. for i = 1 to PS
    1. Generate a random number cr_i from U(0,1)
    2. if cr_i < crossover probability
        1. Randomly choose a neighbour from B(i)
        2. Generate an offspring x by applying the robust one-way crossover operator on this neighbour and I_i
        3. Renumber the communities in offspring x.
        4. Update cohesiveness of offspring x.
    3. else
        1. Set x as I_i
    4. end if
    5. Generate an NC-dimensional random vector mr_i from U(0,1)
    6. for j = 1 to NC
        1. if mr_{ij} > cohesiveness(community 'j' in offspring x)
        2. Apply mutation on offspring x
        3. end if
    7. end for
    8. Renumber the communities in offspring x.
    9. Update cohesiveness of offspring x
    10. if KKM(x) < z_1^*, set z_1^* as KKM(x)
    11. if RC(x) < z_2^*, set z_2^* as RCx)
    12. for each individual I_j B(i)
        1. if $g^{te}(I_j \mid ^i, z^*) > g^{te}(x \mid ^i, z^*)$
        2. replace I_j with offspring x
        3. end if
    13. end for

14. Remove from NS all the solutions that are dominated by offspring x.
15. Add offspring x to NS if no solutions in NS can dominate offspring x
2. end for
10. end for

Fig. 2. MEMOEA-m algorithm

2.4 Multiobjective evolutionary algorithm

The proposed multiobjective evolutionary algorithm, as shown in Fig. 2, belongs to the MOEA/D category. The two objectives to be optimized are kernel K-means (KKM) [20] and ratio cut (RC) [21]. The second term of KKM is about the average internal degrees summed over all the communities that are to be maximized. Subtracting this term from the first term (a large constant) causes this objective to be minimized. RC is the average external degree summed over all the communities to be minimized. These conflicting objectives result in more than one solution termed the Pareto optimal solution.

$$KKM = 2(N-K) - \sum_{i=1}^{K} \frac{\sum_{m\in C_i, n\in C_i} A_{mn}}{|C_i|} \quad (3)$$

$$RC = \sum_{i=1}^{K} \frac{\sum_{j\in C_i, k\in\{TC-C_i\}} A_{jk}}{|C_i|} \quad (4)$$

where TC is the total number of communities and A is the adjacency matrix.

The objective of the multiobjective optimization problem is to find community partitions by minimizing the KKM and RC to create Pareto optimal solutions. The Tchebycheff approach [13] is used to decompose the multiobjective optimization problem into PS number of scalar optimization subproblems, where PS is the size of the population. The subproblem uses the aggregation function given in equation (5) to find the optimal solution.

$$\text{Minimize } f^{tc}\left(x : \gamma^i, z^*\right) = max(\gamma_1^i|KKM(x) - z_1^*|, \gamma_2^i|RC(x) - z_2^*|) \quad (5)$$

The optimal reference point is z^*, which represents the optimal values for KKM and RC pairs obtained from a fixed number of iterations. In each iteration of the evolutionary algorithm, the <KKM, RC> pair of the current offspring is compared with the current optimal solution z^* and is updated accordingly.

Each subproblem is associated with other subproblems by a uniform weight vector $\gamma^i =< \gamma_1^i, \gamma_2^i >$. The neighborhood of the subproblem is determined by the Euclidean distance with the weight vectors of the other subproblems. The final set of Pareto optimal solutions that cover the Pareto front is obtained from multiple runs of the MEMOEA-m algorithm (lines 9.1.10 – 9.1.15; Fig. 2).

2.4.1 Initial population

The merged-maximal clique nodes are initially permuted, taken one at a time, and assigned to a community if doing so enhances the cohesiveness of the community. If not, start a fresh community and include the clique node in it. The cohesiveness is defined as the ratio of the total strength of the links within the community to the total strength of the links outside the community. An individual I_i of the initial population is the chromosome where in each gene represents the community label of one clique node. Therefore, the size of the individual is equal to the total number of clique nodes. This procedure creates the initial population (lines 4 – 8, Fig. 2).

2.4.2 Crossover

The one-way crossover operator (lines 9.1.2 – 9.1.4; Fig. 2) uses a clique-node-based representation, where each gene has the community label of the corresponding clique node in an ordered fashion from the first clique node to the last one. The decision to perform the crossover operation is made when the generated random number is less than a fixed crossover probability value (pc). The individual I_i is mated with one of its neighbors N_i chosen on a random basis.

A clique node in the individual I_i is chosen as the seed position for crossover with its neighbor. The clique node positions (P) of the seed and its community members in the individual I_i are the positions in the offspring where the genetic information is transferred. The offspring O_i is a copy

of the neighbor N_i, where the positions P are replaced with the community label of the seed in the individual I_i. This manuscript also uses renumbering to avoid duplicate solutions in the population. The computational complexity of crossover operation is $O(M^2 * PS)$, where M is the number of clique nodes and PS is the size of the population.

2.4.3 Mutation

The mutation operator of MEMOEA is clique-based. If there are M cliques, M random numbers are generated, and each random number is compared with the cohesiveness value of the community of a particular clique. If this value is less than the random number, then the clique node is removed from its community and added to the community of its strongly connected neighbor. This neighbor is selected through the weighted roulette wheel method. The clique-based operator increases the run time of the mutation operator since there are a large number of cliques compared to the number of communities at any time during evolution.

In this article, an efficient mutation operator is proposed based on communities (lines 9.1.5 – 9.1.7; Fig. 2) instead of the clique-based mutation operator. If there are NC number of communities, then NC random numbers are generated. This random number is compared with the cohesiveness value of the community, and if it is smaller, one clique is randomly selected from the community. Using the weighted roulette wheel selection, a neighbor of the clique is chosen. If the neighbor is in a different community, then the clique node is moved from its community to the neighbor's community.

As shown in Fig. 3, there are four communities in the community-based representation. In community C1, clique node 5 is chosen for mutation. Its strongest neighbor is in community C2; therefore, it is moved to C2. In a similar manner, clique node 6 in C3 is moved to community C4. The clique nodes in C2 and C4 (8 and 19), which are selected for mutation, are not moved since the strongest neighbor is in the same community.

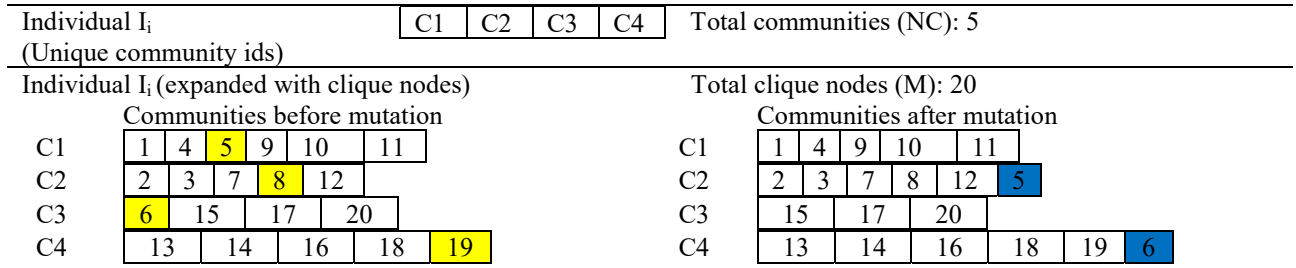

Fig. 3. Community-Based Mutation Operation

The computational complexity of the mutation in MEMOEA-m is $O\ (gen_{max}\ x\ PS\ x\ NC)$, where gen_{max} is the total number of generations and NC is the total number of communities in the chromosome. Compared to the computational complexity of crossover in MEMOEA which is $O\ (gen_{max}\ x\ PS\ x\ M)$, The speedup is still linear, but NC << M in large complex networks.

3. RESULTS AND DISCUSSION

In this section, we compare the performances of MEMOEA-m with those of three other MOEA algorithms and two state-of-the-art algorithms on six real-world and six synthetic networks using two metrics, the gNMI and Qov. The section concludes with a discussion about the community partitions obtained with respect to the gNMI and Qov values.

The real-world networks (Table 1) are Zachary's karate club [22], Bottlenose dolphins [23], American college football [1], U.S. political books [24], the jazz musicians' network [25] and the yeast protein–protein interactions Yeast-D2 [26]. Among the real-world networks, jazz does not have the ground truth. The synthetic networks are created from the well-known LFR method [27], as displayed in Table 2.

Table 1. Real-world networks

No	Network	Nodes (N)	Edges	Avg. degree	Assortativity Coefficient	Density	Ground Truth
1	karate	34	78	4.59	-0.4756	0.1390	2
2	dolphin	62	159	5.13	-0.0436	0.0841	4
3	football	115	613	10.66	0.1624	0.0935	12
4	polbooks	105	441	8.4	-0.1279	0.0808	3
5	jazz	198	2742	27.7	0.0202	0.1405	NA
6	Yeast-D2	1443	6993	9.69	0.4179	0.0067	150

Table 2. Synthetic Networks. Average Degree = 10, Max. Degree = 50, μ = 0.1, T1 = 2, T2 = 1, Minc = 10, Maxc = 50, OM = 2

No	Network	Nodes (N)	Edges	*ON* = N*10%	Assortativity Coefficient	Density	Ground Truth
1	LFR-1	100	527	10	-0.1018	0.1065	5
2	LFR-2	200	923	20	-0.076	0.0464	9
3	LFR-3	300	1461	30	-0.3085	0.0326	12
4	LFR-4	400	1857	40	-0.2441	0.0233	18
5	LFR-5	500	2615	50	-0.1917	0.0209	22
6	LFR-6	1000	5019	100	-0.2351	0.0100	41

The parameters for LFR network generation are the average degree of the nodes, the maximum degree a node can have, the mixing value μ that determines the fraction of connections a node can have outside its community, t1 for degree distribution, t2 for community distribution, the percentage of the nodes that belong to more than one community (ON), and how many community memberships a node can hold (OM). The density of the network is used to determine the relative sparseness of a network compared with other networks. The real-world networks are all sparse [28] from the density perspective since we cannot expect every node added to the network to have an edge to the majority of the existing nodes in the network. The density data are shown in Tables 1 and 2 to illustrate the difference between the relatively high sparsity (Yeast D2) and low sparsity (except Yeast D2) networks.

The metrics used for evaluating the performance of the algorithm are the generalized normalized mutual information (gNMI) [29] and the extended modularity (Qov) [30]. The gNMI is used for all the networks with ground truth, while the Qov is used for all the networks since ground truth is not required for its computation.

The MEMOEA-m algorithm was compared with the other three existing clique-based MOEAs, namely, the MEMOEA, the maximal-clique-based multiobjective evolutionary algorithm (MCMOEA) [10], the multiobjective evolutionary algorithm for signed networks (MEAs_SN) [13], two state-of-the-art community detection algorithms, the nonnegative matrix factorization (NMF) [31], and the Democratic Estimate of the Modular Organization of a Network (DEMON) [32].

The runtime setting of all the evolutionary algorithms are as follows: the size of the population is 100, the number of runs is 20, and the number of generations in each run is 50, the crossover probability is 0.7 to allow high exchange of information between individuals, and the total number of neighbors (T) for each individual is fixed at 20. The default parameters of the state-of-the-art community detection algorithms are used without any modification.

In Table 3, gNMI_max is the maximum gNMI, and gNMI_avg is the average obtained from the 20 runs of the algorithm. The Wilcoxon rank sum test was carried out between the gNMI values from the algorithms being compared. The null hypothesis is that the two sets of samples are drawn from the same distribution. The alternative hypothesis is that the values in one set are more likely to be larger than those in the other. The test is conducted with a significance level of 5%. The symbols +, -, and ≈ indicate that the corresponding samples are significantly larger than, significantly smaller than,

and statistically similar to those of MEMOEA-m, respectively, in terms of the community partitions obtained.

Table 3. Comparison of gNMI values of MEMOEA-m with different algorithms.

Network	Metric	MEMOEA-m	MEMOEA		MCMOEA		MEAs_SN		NMF		DEMON	
karate	gNMI_max	**0.8372**	0.8372		0.2785		0.1709		0.2059		0.1612	
	gNMI_avg	**0.8372**	0.8372	≈	0.2315	-	0.1709	-	0.2059	-	0.1612	-
	Std	0	0		0.0190		0		0		0	
dolphin	gNMI_max	**0.6170**	**0.6170**		0.2759		0.1392		0.4125		0.2753	
	gNMI_avg	0.5273	**0.5871**	+	0.2397	-	0.1308	-	0.4125	-	0.2753	-
	Std	0.0689	0.0435		0.0157		0.0066		0		0	
football	gNMI_max	0.7682	**0.8035**		0.3534		0.5426		0.7828		0.3240	
	gNMI_avg	0.7682	**0.8011**	+	0.3252	-	0.4803	-	0.7828	+	0.3240	-
	Std	0	0.0024		0.0137		0.0325		0		0	
polbooks	gNMI_max	**0.5057**	**0.5057**		0.1926		0.2052		0.1952		0.4414	
	gNMI_avg	**0.5018**	0.4503	-	0.1722	-	0.1902	-	0.1952	-	0.4414	+
	Std	0.0175	0.0373		0.0110		0.0075		0		0	
Yeast-D2	gNMI_max	0.3143	**0.3164**		0.2156		0.1916		0.2492		0.2592	
	gNMI_avg	0.3036	**0.3083**	+	0.2101	-	0.1853	-	0.2492	-	0.2592	-
	Std	0.0045	0.0048		0.0036		0.0044		0		0	
LFR-1	gNMI_max	0.7651	0.7651		**0.9408**		0.5857		0.5091		0.3588	
	gNMI_avg	0.6983	0.7639	+	**0.9345**	+	0.3890	-	0.5091	-	0.3588	-
	Std	0.0625	0.0054		0.0102		0.1158		0		0	
LFR-2	gNMI_max	0.8676	0.9016		**0.9237**		0.5842		0.6934		0.4567	
	gNMI_avg	0.8146	0.8571	+	**0.8557**	+	0.5197	-	0.6934	-	0.4567	-
	Std	0.0323	0.0128		0.0342		0.0440		0		0	
LFR-3	gNMI_max	**0.8995**	0.8995		0.9365		0.5977		0.6781		0.4512	
	gNMI_avg	**0.8955**	0.8995	+	0.8681	-	0.5105	-	0.6781	-	0.4512	-
	Std	0.0099	0		0.0289		0.0660		0		0	
LFR-4	gNMI_max	0.8788	**0.9292**		0.9306		0.5364		0.8330		0.5673	
	gNMI_avg	0.8440	**0.9265**	+	0.8704	+	0.4648	-	0.8330	+	0.5673	-
	Std	0.0198	0.0046		0.0300		0.0487		0		0	
LFR-5	gNMI_max	0.8985	0.9222		**0.9603**		0.5901		0.8019		0.7346	
	gNMI_avg	0.8619	0.9060	+	**0.9392**	+	0.5298	-	0.8019	-	0.7346	-
	Std	0.0161	0.0132		0.0139		0.0507		0		0	
LFR-6	gNMI_max	0.9102	**0.9315**		0.8536		0.5931		0.8910		0.7333	
	gNMI_avg	0.8736	**0.9197**	+	0.8224	-	0.5337	-	0.8910	+	0.7333	-
	Std	0.0171	0.0075		0.0176		0.0385		0		0	

Table 3 shows that the MEMOEA-m algorithm yields statistically similar results on the karate network, which has two communities. In each community, one party hub node is the central component holding that community together. The MEMOEA-m performed significantly better than all the other algorithms in polbooks network. In all the remaining networks, the performance of MEMOEA-m is significantly worse than that of the MEMOEA algorithm, but it is very close to that of MEMOEA. The MCMOEA algorithm outperforms the MEMOEA-m algorithm on all the synthetic networks except for LFR3 and LFR6. The MCMOEA uses single nodes and edges as cliques in its chromosome representation. It is obvious that the density of the networks gradually decreases from LFR1 to LFR6 because the average degree is fixed at 10 in all the synthetic networks. The average degree also contributes to a node having more connections with other nodes, irrespective of whether they are inside or outside a community.

Table 4. Communities found by different algorithms – gNMI perspective

Number of communities							
Network	Ground Truth	MEMOEA-m	MEMOEA	MCMOEA	MEAs_SN	NMF	DEMON
karate	2	**2**	2	10	26	5	2
dolphin	4	4	**4**	30	38	7	4
football	12	14	**12**	42	19	10	9
polbooks	3	**2**	2	39	30	5	5
jazz	NA	:	:	:	:	:	:
Yeast-D2	150	162	**163**	756	573	100	96
LFR-1	5	4	4	**5**	8	6	5
LFR-2	9	9	9	**9**	16	10	7
LFR-3	12	**12**	12	12	20	17	8
LFR-4	18	17	**18**	19	35	22	14
LFR-5	22	23	22	**22**	43	29	23
LFR-6	41	40	**41**	46	87	39	42

Therefore, the MEMOEA-m performs well when the networks become sparser. Sparsity also introduces fewer intraconnections, causing communities to break with higher threshold values. This performance improvement is also obvious in the case of real-world networks such as Yeast-D2 and polbooks.

The reason for the good performance of MEMOEA is that it allows all the clique nodes in the chromosome to be mutated, whereas the MEMOEA-m allows for only one clique node from each community to be mutated. If this clique node's community has a greater cohesiveness value than the generated random number, the community will not be modified by mutation. MEMOEA-m showed good results despite making few changes in the chromosome. Interestingly, the communities found by MEMOEA-m are close to the ground truth and to the number of communities found by MEMOEA, as shown in Table 4. In Table 4, the values (shown in bold) are the total community partitions found by the best performing algorithm corresponding to the highest gNMI_max value.

Table 5. Comparison of Qov values of MEMOEA-m with different algorithms.

Network	Metric	MEMOEA-m	MEMOEA		MCMOEA		MEAs_SN		NMF		DEMON	
karate	Qov_max	0.2108	**0.2166**		0.0868		0.0533		0.1297		0.0998	
	Qov_avg	0.2108	**0.2163**	+	0.0822	-	0.0533	-	0.1297	-	0.0998	-
	Std	0	0.0013		0.0116		0		0		0	

dolphin	Qov_max	0.2573	**0.2570**		0.1137		0.1122		0.1747		0.0917	
	Qov_avg	0.2380	**0.2527**	+	0.0939	-	0.1101	-	0.1747	-	0.0917	-
	Std	0.0010	0.0083		0.0093		0.0014		0		0	
football	Qov_max	0.2936	**0.3059**		0.0731		0.2213		0.2957		0.1494	
	Qov_avg	0.2936	**0.3051**	+	0.0675	-	0.2056	-	0.2957	+	0.1494	-
	Std	0	0.0006		0.0041		0.0132		0		0	
polbooks	Qov_max	0.2508	**0.2531**		0.0792		0.2054		0.1809		0.0948	
	Qov_avg	0.2445	**0.2464**	+	0.0696	-	0.2009	-	0.1809	-	0.0948	-
	Std	0.0048	0.0031		0.0055		0.0043		0		0	
jazz	Qov_max	0.1703	**0.1887**		0.1629		0.1995		0.0620		0.0035	
	Qov_avg	0.1631	**0.1789**	+	0.1491	-	0.1114	-	0.0620	-	0.0035	-
	Std	0.0065	0.0076		0.0085		0.0824		0		0	
Yeast-D2	Qov_max	0.3734	**0.3793**		0.0853		0.3336		0.2721		0.2212	
	Qov_avg	0.3658	**0.3723**	+	0.0819	-	0.3159	-	0.2721	-	0.2212	-
	Std	0.0053	0.0032		0.0017		0.0075		0		0	
LFR-1	Qov_max	0.1871	0.1871		**0.2031**		0.1671		0.1009		0.0326	
	Qov_avg	0.1608	0.1868	+	**0.1996**	+	0.1169	-	0.1009	-	0.0326	-
	Std	0.0200	0.0013		0.0029		0.0342		0		0	
LFR-2	Qov_max	0.3267	**0.3302**		0.3234		0.3036		0.2541		0.1144	
	Qov_avg	0.3065	**0.3226**	+	0.3091	≈	0.2890	-	0.2541	-	0.1144	-
	Std	0.0067	0.0026		0.0074		0.0073		0		0	
LFR-3	Qov_max	0.3612	0.3612		**0.3620**		0.3413		0.2585		0.1886	
	Qov_avg	**0.3576**	0.3544	-	0.3554	≈	0.3216	-	0.2585	-	0.1886	-
	Std	0.0030	0.0058		0.0040		0.0136		0		0	
LFR-4	Qov_max	0.3613	**0.3679**		0.3644		0.3341		0.3135		0.1764	
	Qov_avg	0.3545	**0.3654**	+	0.3557	+	0.3166	-	0.3135	-	0.1764	-
	Std	0.0040	0.0010		0.0052		0.0094		0		0	
LFR-5	Qov_max	0.3672	**0.3711**		0.3682		0.3460		0.2901		0.1633	
	Qov_avg	0.3625	**0.3708**	+	0.3646	+	0.3343	-	0.2901	-	0.1633	-
	Std	0.0018	0.0007		0.0026		0.0091		0		0	
LFR-6	Qov_max	0.3780	**0.3798**		0.3651		0.3534		0.3565		0.2080	
	Qov_avg	0.3747	**0.3780**	+	0.3572	-	0.3478	-	0.3565	-	0.2080	-
	Std	0.0030	0.0013		0.0046		0.0041		0		0	

Table 5 shows the average and maximum Qov values from the 20 runs of the MEMOEA-m algorithm. The Wilcoxon rank sum test was conducted on the average Qov values from the 20 runs for every pair of algorithms being compared. According to the Qov metric, the performance of MEMOEA-m is significantly better than that of MEMOEA and MCMOEA for LFR3.

MEMOEA-m outperforms MCMOEA on all the real-world networks. For the case of the jazz network for which the ground truth is unavailable, MEMOEA-m yields better results than does MCMOEA. The proposed algorithm outperforms NMF and DEMON in all the synthetic networks when the Qov metric is used.

Table 6. Communities found by different algorithms – Qov perspective

Network	Ground Truth	Number of communities					
		MEMOEA-m	MEMOEA	MCMOEA	MEAs_SN	NMF	DEMON
karate	2	2	**3**	10	26	5	2
dolphin	4	4	**4**	20	38	7	4
football	12	14	**11**	40	19	10	9
polbooks	3	3	**4**	31	28	5	5
jazz	NA	5	**6**	14	18	39	1
Yeast-D2	150	157	**157**	727	556	100	96
LFR-1	5	4	4	**5**	6	6	5
LFR-2	9	9	**9**	8	17	10	7
LFR-3	12	**11**	9	10	17	17	8
LFR-4	18	16	**15**	18	39	22	14
LFR-5	22	22	**20**	22	43	29	23
LFR-6	41	36	**39**	45	76	39	42

From the perspective of the Qov metric, the total communities identified by MEMOEA-m (Table 6) are closer to the ground truth, although they are not better than those identified by MEOMEA; however, getting closer for the sparser LFR6 network (refer the density values in Table 2). Compared with MEMOEA, MEMOEA-m shows consistently similar performance.

From the above discussions, it is evident that although the MEMOEA-m algorithm does not always yield statistically better solutions than the MEMOEA algorithm does, it has similar performance with respect to the gNMI average values and Qov average values. The total number of communities detected by MEMOEA-m is closer to the ground truth than that detected by MCMOEA and other state-of-the-art algorithms.

4. CONCLUSION

The networks that model the interactions between entities in the real world are of different kinds when viewed from different perspectives, such as the density of the networks, well-formedness of communities, and embeddedness of communities [33]. The problem of finding overlapping communities is difficult because the communities may have weak intraconnectivity, they may be embedded in dense interconnections, or they may be blurred by the date hub nodes that behave as the influential nodes connecting to multiple communities.

In this article, a new mutation operator is designed that works on a community-based chromosome representation scheme. The execution of the mutation operation now depends on the number of communities rather than the large number of clique nodes thereby reducing the computation time considerably. The mutation operator of MEMOEA-m yields good results when compared with those of state-of-the-art community detection algorithms. Also, it showcases comparable performance with the best performing clique-based algorithm, MEMOEA based on the gNMI and Qov outcome measures. The MEMOEA-m algorithm performs on par with MEMOEA in networks that are relatively sparse, such as LFR6, and that have ill-formed community structures, such as Yeast D2.

The proposed mutation operator gives equal importance to all communities irrespective of their size in its procedure by choosing only one clique node per community to participate in the mutation. This research can be further improved by adapting the mutation operator to consider the different sizes of communities. Another research direction is to redesign the algorithm to work on undirected weighted networks either with certain or uncertain edges. The mutation operator can also be applied to other MOEA frameworks, such as NSGA-III [34] or knee point-driven EA [35], to detect communities in large networks.

AVAILABILITY OF SUPPORTING DATA

The real-world networks, karate, dolphin, polbooks and jazz can be downloaded from the network repository[1]. The football network from The KONECT Project[2] and the Yeast-D2 from ProRank: Supplementary Materials[3], hosted by Nazar Zaki. The LFR synthetic networks can be generated using package 1 from the Resources[4] page provided by Santo Fortunato.

REFERENCES

[1] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[2] A. Bhar *et al.*, "Application of a maximal-clique based community detection algorithm to gut microbiome data reveals driver microbes during influenza A virus infection," *Front Microbiol*, vol. 13, Oct. 2022, doi: 10.3389/fmicb.2022.979320.

[3] B. Adamcsek, G. Palla, I. J. Farkas, I. Derényi, and T. Vicsek, "CFinder: locating cliques and overlapping modules in biological networks," *Bioinformatics*, vol. 22, no. 8, pp. 1021–1023, Apr. 2006, doi: 10.1093/bioinformatics/btl039.

[4] B. Cao, J. Luo, C. Liang, S. Wang, and P. Ding, "PCE-FR: A Novel Method for Identifying Overlapping Protein Complexes in Weighted Protein-Protein Interaction Networks Using Pseudo-Clique Extension Based on Fuzzy Relation," *IEEE Trans Nanobioscience*, vol. 15, no. 7, pp. 728–738, Oct. 2016, doi: 10.1109/TNB.2016.2611683.

[5] F. Hao, G. Min, Z. Pei, D.-S. Park, and L. T. Yang, "K-Clique Community Detection in Social Networks Based on Formal Concept Analysis," *IEEE Syst J*, vol. 11, no. 1, pp. 250–259, 2017, doi: 10.1109/JSYST.2015.2433294.

[6] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys Rev E*, vol. 69, no. 2, p. 26113, Feb. 2004, doi: 10.1103/PhysRevE.69.026113.

[7] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, Jan. 2007, doi: 10.1073/pnas.0605965104.

[8] C. Shi, Z. Yan, Y. Cai, and B. Wu, "Multi-objective community detection in complex networks," *Appl Soft Comput*, vol. 12, no. 2, pp. 850–859, Feb. 2012, doi: 10.1016/j.asoc.2011.10.005.

[9] Qingfu Zhang and Hui Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, Dec. 2007, doi: 10.1109/TEVC.2007.892759.

[10] X. Wen *et al.*, "A Maximal Clique Based Multiobjective Evolutionary Algorithm for Overlapping Community Detection," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, p. 1, 2016, doi: 10.1109/TEVC.2016.2605501.

[11] Y. Li, Y. Wang, J. Chen, L. Jiao, and R. Shang, "Overlapping community detection through an improved multi-objective quantum-behaved particle swarm optimization," *Journal of Heuristics*, vol. 21, no. 4, pp. 549–575, Aug. 2015, doi: 10.1007/s10732-015-9289-y.

[12] C. Pizzuti, "A Multiobjective Genetic Algorithm to Find Communities in Complex Networks," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 3, pp. 418–430, Jun. 2012, doi: 10.1109/TEVC.2011.2161090.

[13] Chenlong Liu, Jing Liu, Zhongzhou Jiang, C. Liu, J. Liu, and Z. Jiang, "A Multiobjective Evolutionary Algorithm Based on Similarity for Community Detection From Signed Social Networks," *IEEE Trans Cybern*, vol. 44, no. 12, pp. 2274–2287, Dec. 2014, doi: 10.1109/TCYB.2014.2305974.

[14] A. C. Ramesh and G. Srivatsun, "Evolutionary Algorithm for overlapping community detection using a merged maximal cliques representation scheme," *Appl Soft Comput*, vol. 112, 2021, doi: 10.1016/j.asoc.2021.107746.

[15] K. Sekine and T. Tatsukawa, "A Parametric Study of Crossover Operators in Multi-objective Evolutionary Algorithm," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, Nov. 2018, pp. 1196–1203. doi: 10.1109/SSCI.2018.8628707.

[16] W. Wang, K. Li, X. Tao, and F. Gu, "An improved MOEA/D algorithm with an adaptive evolutionary strategy," *Inf Sci (N Y)*,

[1] https://networkrepository.com/
[2] http://konect.cc/networks/dimacs10-football/
[3] https://faculty.uaeu.ac.ae/nzaki/ProRank.htm
[4] https://www.santofortunato.net/resources

vol. 539, pp. 1–15, Oct. 2020, doi: 10.1016/j.ins.2020.05.082.

[17] J. Ji, Y. Guo, D. Gong, and W. Tang, "MOEA/D-based participant selection method for crowdsensing with social awareness," *Applied Soft Computing Journal*, vol. 87, 2020, doi: 10.1016/j.asoc.2019.105981.

[18] C. Bron and J. Kerbosch, "Algorithm 457: finding all cliques of an undirected graph," *Commun ACM*, vol. 16, no. 9, pp. 575–577, Sep. 1973, doi: 10.1145/362342.362367.

[19] J.-D. J. Han *et al.*, "Evidence for dynamically organized modularity in the yeast protein–protein interaction network," *Nature*, vol. 430, no. 6995, pp. 88–93, Jul. 2004, doi: 10.1038/nature02555.

[20] L. Angelini, S. Boccaletti, D. Marinazzo, M. Pellicoro, and S. Stramaglia, "Identification of network modules by optimization of ratio association," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 17, no. 2, p. 23114, Jun. 2007, doi: 10.1063/1.2732162.

[21] M. Gong, Q. Cai, X. Chen, and L. Ma, "Complex Network Clustering by Multiobjective Discrete Particle Swarm Optimization Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 82–97, Feb. 2014, doi: 10.1109/TEVC.2013.2260862.

[22] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J Anthropol Res*, vol. 33, no. 4, pp. 452–473, 1977.

[23] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations," *Behav Ecol Sociobiol*, vol. 54, no. 4, pp. 396–405, Sep. 2003, doi: 10.1007/s00265-003-0651-y.

[24] V. Krebs, "Orgnet LLC." [Online]. Available: http://www.orgnet.com/

[25] P. M. Gleiser and L. Danon, "Community structure in jazz," *Adv Complex Syst*, vol. 06, no. 04, Dec. 2003, doi: 10.1142/S0219525903001067.

[26] N. Zaki, J. Berengueres, and D. Efimov, "ProRank," in *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, New York, NY, USA: ACM, Jul. 2012, pp. 209–216. doi: 10.1145/2330163.2330193.

[27] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys Rev E*, vol. 78, no. 4, p. 46110, Oct. 2008, doi: 10.1103/PhysRevE.78.046110.

[28] C. I. Del Genio, T. Gross, and K. E. Bassler, "All Scale-Free Networks Are Sparse," *Phys Rev Lett*, vol. 107, no. 17, p. 178701, Oct. 2011, doi: 10.1103/PhysRevLett.107.178701.

[29] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New J Phys*, vol. 11, no. 3, p. 33015, Mar. 2009, doi: 10.1088/1367-2630/11/3/033015.

[30] H. Shen, X. Cheng, K. Cai, and M.-B. Hu, "Detect overlapping and hierarchical community structure in networks," *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 8, pp. 1706–1712, Apr. 2009, doi: 10.1016/j.physa.2008.12.021.

[31] J. Yang and J. Leskovec, "Overlapping community detection at scale," in *Proceedings of the sixth ACM international conference on Web search and data mining*, New York, NY, USA: ACM, Feb. 2013, pp. 587–596. doi: 10.1145/2433396.2433471.

[32] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi, "Uncovering Hierarchical and Overlapping Communities with a Local-First Approach," *ACM Trans Knowl Discov Data*, vol. 9, no. 1, pp. 1–27, Oct. 2014, doi: 10.1145/2629511.

[33] C. H. Yong and L. Wong, "Prediction of problematic complexes from PPI networks: sparse, embedded, and small complexes," *Biol Direct*, vol. 10, no. 1, p. 40, Dec. 2015, doi: 10.1186/s13062-015-0067-4.

[34] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, Aug. 2014, doi: 10.1109/TEVC.2013.2281535.

[35] X. Zhang, Y. Tian, and Y. Jin, "A Knee Point-Driven Evolutionary Algorithm for Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, Dec. 2015, doi: 10.1109/TEVC.2014.2378512.