Research Article

PPI-GA: A Novel Clustering Algorithm to Identify Protein Complexes within Protein-Protein Interaction Networks Using Genetic Algorithm

Naeem Shirmohammady, Habib Izadkhah , and Ayaz Isazadeh

Department of Computer Science, Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran

Correspondence should be addressed to Habib Izadkhah; izadkhah@tabrizu.ac.ir

Received 15 June 2020; Revised 12 January 2021; Accepted 10 March 2021; Published 25 March 2021

Academic Editor: Mojtaba Ahmadi Khansar

Copyright © 2021 Naeem Shirmohammady et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Comprehensive analysis of proteins to evaluate their genetic diversity, study their differences, and respond to the tensions is the main subject of an interdisciplinary field of study called proteomics. The main objective of the proteomics is to detect and quantify proteins and study their post-translational modifications and interactions using protein chemistry, bioinformatics, and biology. Any disturbance in proteins interactive network can act as a source for biological disorders and various diseases such as Alzheimer and cancer. Most current computational methods for discovering protein complexes are usually based on specific topological characteristics of protein-protein networks (PPI). To identify the protein complexes, in this paper, we, first, present a new encoding method to represent solutions; we then propose a new clustering algorithm based on the genetic algorithm, named PPI-GA, employing a new multiobjective quality function. The proposed algorithm is evaluated on two gold standard and real-world datasets. The result achieved demonstrates that the proposed algorithm can detect important protein complexes, and it provides more accurate results compared with state-of-the-art protein complex identification algorithms.

1. Introduction

Proteins are essential nutrients which carry out cellular activities. They are also vital to our understanding of molecular functions and biological processes. In biological organisms, proteins are typically organized into a protein complex. When two or more are in physical contact, due to electrostatic forces or chemical phenomena, protein-protein interaction occurs. Such interactions often lead to biological functions. Many important molecular processes within a cell such as DNA replication and even protein metabolism are performed by large molecular complexes which consisted of many organized proteins interacting with each other.

It is important to expose protein interactions because it improves our knowledge about diseases and can establish new therapeutic methods. Although there are many experimental and computational methods to detect protein complexes, all of them have their advantages and disadvantages. Experimental

approaches like yeast two-hybrid screening and tandem affinity purification with mass spectrometry can be used to detect protein complexes [1]. There are many biophysical methods to investigate the nature and properties of these interactions. These methods have certain limitations which are presented in [2]. For example, PPI data extracted from experiments typically have high false-negative and false-positive rates which have a negative impact on the accuracy of the experiment results. Although various computational density-based approaches were proposed to find connected subgraphs in PPI networks over the past decade, there are still some challenges to effectively detect protein complexes based on main properties and topology of the network which reduces the accuracy of results. Theoretically, graph theory is usually used to model the interactions among artifacts (here proteins). Thus, the developed computational approaches are used as a supplement to experimental methods. To detect the protein complexes, the set of all the protein-protein interactions of a given organism is

naturally modeled by an undirected graph $G(V, E)$ in which each vertex V and each edge E represent a protein and the interaction between two proteins, respectively.

Protein detection in an interactive protein network is challenging which depends on some factors. Extracted interactive data usually have high false-negative and false-positive rates. Also, each protein can belong to multiple protein complexes. Computational methods for protein complex detection mostly focus on extracting clusters, i.e., complexes, from PPI networks [1]. The purpose of PPI networks clustering is to find a group of proteins that interact and participate in the same biological process or proteins that perform a particular biological function. Since detecting protein complexes is an NP-hard problem, evolutionary algorithms such as genetic algorithm are utilized to find near-optimal solutions [3].

This paper presents a graph clustering algorithm with a novel multiobjective quality function and a new compacted encoding method to identify protein complexes from PPI networks. In this approach, we employ the genetic algorithm to find clusters for protein complexes. Comparing our results to previous algorithms shows that the proposed algorithm outperforms the existing clustering algorithms. Our approach has the highest computational accuracy among the compared approaches. The algorithm can detect clusters with different densities and complexes. Our algorithm is scalable so that the algorithm can cluster a large network without any disruption to its performance.

The contributions of the paper are summarized as follows:

- (1) Presenting a new multiobjective quality function aiming to identify protein complexes;
- (2) Presenting a new encoding method to show the solutions in the evolutionary process of the genetic algorithm, aiming to reduce the search space. The solution space produced by existing evolutionary algorithms is n^n , where n is the number of proteins. The encoding proposed in this paper reduces the state space up to $n!$ which is much smaller in comparison to n^n . This reduction in the search space makes (1) solving the problem easier for the GA and (2) converging faster and finding acceptable solutions during earlier generations.

The rest of this paper is organized into five sections. Section 2 addresses available algorithms to predict protein complexes. Section 3 describes the proposed clustering algorithm. In this section, we will present a new encoding and a new multiobjective function. Section 4 compares and analyzes the empirical results of different algorithms using various evaluation criteria. Section 5 presents the conclusion and future work.

2. Related Work

Proteins are a vital component of living beings, and they play a key role in managerial and executing most biological processes. Many clustering algorithms have been designed to detect meaningful groups of proteins from PPI networks [4].

There are three main approaches to cluster graphs aiming protein complex detection. The first one is to find subgraphs with specified connections which are called network motifs and is described as a complex or a part of it. Clique is one of the subgraphs detected using this approach. Due to the time complexity of this approach, its application is limited nowadays. The second one is graph-growing in which a cluster grows and completes around a vertex as a seed using greedy search algorithms. The third one includes several variants. The algorithms try to minimize or maximize measures of a specified cluster such as connection density, edge cut, or a metric distance between nodes. Generally, the algorithms aim to optimize an objective function for the entire graph.

Dongen et al. [5] introduced the MCL algorithm that uses expansion and inflation operators to characterize the random walk process on the graph and find protein families. It is shown in [6] that this algorithm is one of the best methods to cluster a graph. King et al. in [7] proposed the RNCS algorithm which apportions vertices of a graph using a cost function and a local search algorithm. In another research, Dunn et al. [8] used the Newman average edge algorithm to extract protein complexes. In this algorithm, the edges with the highest average value are removed iteratively until the desired clusters are achieved.

Bader et al. [9] proposed the MCODE algorithm which is one of the first and the most well-known algorithm for predicting protein complexes. In this algorithm, a cluster grows around a seed protein with a greedy approach. The algorithm is composed of three steps: weighting the vertices, predicting the clusters, and preprocessing to filter or add a vertex to clusters.

Jiang et al. [10] presented the SPICi algorithm which was proposed to cluster large biological networks. The clustering process is done by the formation of an initial seed pair and then adding nodes to the initial pair.

Amin et al. [11] introduced the CPA algorithm which uses degree and distance length between two nodes in a cluster. A cluster is made and then removed from the network, and it is repeated until there is no edge left. To form a cluster, the node with the highest weight is chosen as the seed. Then, neighboring nodes are searched and added to the seed through the function. If the function value is the same for multiple nodes, the node with higher distance length is added to the current cluster, and if distance lengths are the same as well, all the nodes will be added to the cluster.

Nepuse et al. [12] proposed the ClusterOne algorithm which takes advantages of a greedy approach. Generally, a seed vertex with the highest degree in the network is chosen. Next, some vertices are added to or removed from the seed through a function with a greedy approach. To detect the overlapping in clusters, all the pair complexes at which the value of a function named Overlap Score exceeds a certain threshold are combined. Finally, the complexes with less than three proteins or density lower than a certain threshold are removed.

Adamcsek et al. [13] proposed the CFinder algorithm. This algorithm is based on the concept of the clique. A cluster can be considered as a small fully connected unit. A

parameter is used to determine the minimum number of common proteins. First, the algorithm extracts all the subgraphs of the entire network. Second, a matrix is formed to extract the number of common nodes between two cliques. Finally, the cliques with $(k - 1)$ common nodes are considered as adjacent and the two cliques are combined.

Pizzuti et al. [14] proposed an algorithm where the operators presented in the restricted neighborhood search clustering (RNSC) algorithm are modified. Instead of evaluating the cost function at each node, a set of nodes is used and the cost function is rewritten. The results showed that even though fewer clusters are predicted in compared with the RNSC method, but high quality clusters were obtained.

Cao et al. [15] presented the MOEPGA algorithm based on a multiobjective function. The main principle of the algorithm is to consider multiple topological properties of the network including name, size, density, and distance length. The algorithm initially analyzes an interactive protein network and then formulates the multiobjective function based on topological properties of the network. Three steps are taken in each subgraph creating the initial population, mutation, and selection.

Lei et al. [16] presented a moth-flame optimization-based protein complex prediction algorithm, called MFOC. In the MFOC, first, a weighted dynamic PPI network by synthesizing topological and biological information is created. Then, a layer-by-layer scheme is utilized to find the cores of protein complexes as the flames and let the moths fly spirally around the flames to form the complexes.

Sikandar et al. [17] presented an algorithm that considers both topological patterns and biological features for clustering a PPI network. In their algorithm, each complex subgraph is modeled by decision tree learners. They used a training set of known complexes to construct decision trees in depth-first and best-first manner using divide and conquer strategy.

Gu et al. [18] presented a Markov-based clustering algorithm that uses link similarity to identify the overlapping structures. They have shown in their research that their proposed method can find more modules with biological significance in PPI networks.

Ramadan et al. [3] proposed an algorithm in which the network is clustered by a proposed genetic algorithm so that its population is created by random and spectral methods. To enhance clustering quality, genetic operations and objective functions are adopted. Given that clustering aims to find subgraphs with the high intraconnections, the algorithm first uses spectral clustering to find the minimum cut between two subgraphs. The minimum cut of the network is derived by eigenvector and Laplacian and diagonal matrixes.

Table 1 lists some protein networks clustering algorithms.

3. Proposed Algorithm

In this section, we present a new PPI network clustering algorithm based on the genetic algorithm. The genetic Algorithm (GA) is a computational model based on

TABLE 1: Summary of some PPI networks clustering algorithms.

Method 1	Structure 2	Class 3	Overlapping 4	Reference
MCODE	Dense	LD	Yes	[9]
SPICi	Dense	LD	Yes	[10]
CPA	Dense	LD	Yes	[11]
ClusterOne	Dense	LD	Yes	[12]
CFINDER	Dense	LD	Yes	[13]
RNSC	Dense	CL	No	[6]
MOEPGA	Dense	PS	Yes	[15]
Ramadan	Dense	PS	Yes	[3]
MCL	Arbitrary	FS	Yes	[4]
DMCL-EHO	Dense	PS	Yes	[19]
K-means	Arbitrary	CL	Yes	[20]

1: method acronym; 2: topological structure a method searches for (arbitrary or dense subgraphs); 3: the class of the method, where LD: local neighborhood density, CL: cost-based local search, FS: flow simulation, and PS: population-based stochastic search if the method generates overlapping clusters; 4: if the method generates overlapping clusters.

biological evolution which has many advantages over traditional optimization methods. GA can adapt to complex optimization problems and thus a wide range of problem can be solved by it. In most cases, each GA starts with a randomly selected initial population which evolves to achieve a certain goal, i.e., optimizing a function through applying genetic operators (crossover and mutation) to the available chromosomes and creates a new population. The new population is reset on the previous one, and the process is repeated until the algorithm satisfies the termination condition.

Considering the adequate performance of evolutionary algorithms in finding a near-optimal solution to NP-complete problems, we employ an evolutionary algorithm (PPI-GA) to find the near-optimal solution to the problem of discovering protein complexes in interactive PPI networks. In the following, the operators used in the PPI-GA are described.

Encoding. GA encodes the possible solutions to a specific problem as a simple chromosome. Choosing the right encoding method is crucial for the performance of the genetic algorithm. Weak encoding scheme can lead a genetic algorithm to converge unacceptable results. We present a new encoding method to represent solutions for clustering PPI networks. To this end, each node (i.e., protein) in the PPI networks is assigned a unique numerical identifier. These identifiers determine which position in the encoded string is used to that node. The method encodes a partition of the PPI network as a string such that each string is a permutation of N integers. Formally, an encoding, P , is defined as follows:

$$P = c_1 \ c_2 \ c_3 \ c_4 \ \dots \ c_N, \quad (1)$$

where N is the number of proteins in the PPI networks and c_i ($1 \leq i \leq N$) holds a value in range $[1, N]$. Length of each chromosome is equal to a number of proteins, indicated by N . Algorithm 1 shows how we can decode a chromosome into a clustering.

For example, consider the following encoding:

```

Let P is a chromosome which holds a permutation of PPI network.
node 1 to N
For each element i in the chromosome
  If (P[i] ≥ i) then
    Create a new cluster and assign node i to it.
  Else
    Assign node i to the same cluster as node P[i].
End.

```

ALGORITHM 1: Decoding a chromosome.

$P: 9, 3, 2, 1, 4, 6, 10, 7, 5, 8.$ (2)

In the above permutation encoding, the first value in P is 9 which is greater than 1; hence, a new cluster called cluster 1 is created and assigned node 1 to it. The second value in P is 3 which is greater than 2; hence, node 2 is assigned to a new cluster (cluster 2). Node 3 is assigned to the same cluster as node 2. Nodes 4 and 5 are assigned to the same clusters as nodes 1 and 4, respectively, because the fourth and fifth values in P are 1 and 4. Figure 1 shows clustering obtained for this encoding.

Algorithm 2 shows the encoding method described in the above.

Input: a PPI network with N proteins and K complexes (clusters)

Output: a Chromosome P with N genes

Begin

Create an empty chromosome C with N genes.

For $i = 1$ to k

A is an array containing nodes in cluster $\#i$

P is the number of nodes in A

 Sort A , in ascending order

 For $j = 1$ to $p-1$

 Assign $A(j)$ to $C(A(j+1))$

 Next j

Assign $A(p)$ to $C(A(1))$

Next i

End

By investigating the proposed encoding, it is obvious that for a given PPI network with n nodes, the search space produced by the encoding for the genetic algorithm may include up to $n!$ chromosomes which is much smaller in comparison with the n^n chromosomes produced by the existing value-based encoding (see [3]). This makes (1) solving the problem easier for the GA and (2) converging faster and finding acceptable solutions during earlier generations due to the reduction in the search space.

Fitness Function. Each chromosome in GA represents an individual (solution). The fitness of each chromosome is computed using an objective function. To evaluate the fitness of chromosomes during the evolutionary process of the algorithm, a multiobjective fitness function is proposed. The

FIGURE 1: A sample clustering.

genetic algorithm aims to optimize this multiobjective function. It is important to note that the output of GA is highly dependent on the fitness function. The first objective function, called CQ, is defined as follows:

$$CQ = \sum_{i=1}^k CF_i, \quad (3)$$

where CF_i is calculated as follows:

$$CF_i = \frac{2\mu_i}{2\mu_i + \sum_{j=1}^k (\varepsilon_{ij} + \varepsilon_{ji})}. \quad (4)$$

In equation (2), k denotes the total number of clusters, μ_i is the number of edges (i.e., communications) inside the cluster i , and ε_{ij} is the number of edges between cluster i and j . CF_i is defined as a normalized ratio between the total weight of the internal edges and half of the total weight of external edges (edges that exit or enter the cluster). The CQ fitness function calculates communications between the proteins of two different clusters (i.e., inter-relationship) and communications between the proteins of the same cluster (i.e., intrarerelationship). The CQ gives higher values while the intrarerelationships increase and the interrelationships decrease.

The second objective function is defined as follows:

$$of = \sum (\mu_i - \varepsilon_{ij}). \quad (5)$$

This function measures the difference between the number of internal communications of a cluster and the total external communication of that cluster. The larger the number, the better the cluster. Finally, the multiobjective function is introduced as (4):

```

Input: a PPI network with N proteins and K complexes (clusters)
Output: a Chromosome P with N genes
Begin
Create an empty chromosome C with N genes.
For i = 1 to k
    A is an array containing nodes in cluster #i
    P is the number of nodes in A
    Sort A, in ascending order
    For j = 1 to p-1
        Assign A(j) to C(A(j+1))
    Next j
    Assign A(p) to C(A(1))
Next i
End

```

ALGORITHM 2: Encoding method.

$$\text{Of} = \text{CQ} + \text{of}. \quad (6)$$

The aim of the CQ function is to maximize the number of intracluster edges (μ_i) and minimize the number of inter-cluster connections (ϵ). However, because the variables of the functions are not normal, it needs to be normalized. The value of properties should be normalized to common domains.

The range of CQ is between 0 and the maximum number of clusters, and the second objective function is between $(-\epsilon)$ and $(+\epsilon)$, which is different from the CQ. This difference leads to small values for the first objective function and large values for the second objective function at each chromosome which means that the effect of the first objective function is ignored. After normalization, the multiobjective fitness function is modified as follows:

$$\text{Of} = \left(\frac{\text{CQ}/K + (\text{sum}(\mu_i) - \text{sum}(\epsilon))/E}{2} \right). \quad (7)$$

In equation (5), K and E denote the total number of clusters and the total number of edges, respectively, and $\text{sum}(\mu_i) = \sum_{i=1}^k \mu_i$ and $\text{sum}(\epsilon) = \sum \epsilon_i$.

Algorithm 3 depicts the pseudocode of our algorithm. In each iteration, the evolutionary operators (selection, crossover, and mutation) are applied to the population, respectively, and the population is steered toward the optimal solution.

Input: protein-protein interaction network (PPI)

Output: a clustered PPI

Algorithm steps are as follows:

- (1) Generate the initial population
- (2) Evaluate the chromosomes fitness by Equation 5
- (3) Generate the intermediate population by employing the standard roulette wheel selection strategy for selecting potentially useful chromosomes for recombination
- (4) Apply crossover operator on the intermediate
- (5) Apply mutation operator on the intermediate generation

- (6) Apply elitism operator
- (7) Replace the old population with a new population
- (8) If the number of iteration is less than that of generation, go to Step 2; otherwise, stop the iteration steps and return the best result

Initial Population and Operators. To initialize the population, chromosomes to the number of the population are randomly created. Each chromosome is a permutation of N integers with equal probability. To select chromosomes, the tournament selection method is employed. At first, a certain and limited number of chromosomes (tournament size) is randomly selected. Then, the selected chromosomes are compared and the chromosome with the best fitness is chosen.

To make a crossover operator, because the presented encoding method is a permutation, we need a crossover operator that maintains the permutation form in each chromosome during the evolutionary process of the genetic algorithm. In the PPI-GA, a two-point crossover operator is used. Consider the following two parents:

P1: $C_1 C_2, \dots, C_k C_{k+1} C_{k+3} \cdot C_{k+4} C_L$

P2: $D_1 D_2, \dots, D_k D_{k+1} D_{k+3} \cdot D_{k+4} D_L$

To build two offspring O1 and O2, first, the algorithm generates randomly two positions $k1$ and $k2$. Middle parts of two offspring are built using the elements between two positions $k1$ and $k2$ of two parents P1 and P2:

O1: $\dots C_{k+1} C_{k+2} C_{k+3} \dots$

O2: $\dots D_{k+1} D_{k+2} D_{k+3} \dots$

To fill the other parts of O1 and O2, the following two lists are created:

List for O1, 2: $D_{k+1} \dots D_L D_1 D_2 \dots D_k$ and $C_{k+4} \dots C_L C_1 C_2 \dots C_k$

Starting from the 1st position in the list, if the element in the 1st position does not exist in the middle part of O1, it will be assigned to O1. Otherwise, it will be assigned to O2. If the element in the 2nd position does not exist in the middle part of O1, it will be assigned to

Input: protein-protein interaction network (PPI)

Output: a clustered PPI

Algorithm steps are as follows:

- (1) Generate the initial population
- (2) Evaluate the chromosomes fitness by Equation 5
- (3) Generate the intermediate population by employing the standard roulette wheel selection strategy for selecting potentially useful chromosomes for recombination
- (4) Apply crossover operator on the intermediate
- (5) Apply mutation operator on the intermediate generation
- (6) Apply elitism operator
- (7) Replace the old population with a new population
- (8) If the number of iteration is less than that of generation, go to Step 2; otherwise, stop the iteration steps and return the best result

ALGORITHM 3: The genetic algorithm process.

O1. Otherwise, it will be assigned to O2, and so on. Analogously, offspring O1 may be generated as follows:

$$D_{k+4} \dots D_L C_{k+4} C_{k+2} C_{k+3} D_1 D_2 \dots D_k$$

As the same way, for parent O2, we have the following:

$$C_{k+4} \dots C_L D_{k+1} D_{k+2} D_{k+2} C_1 C_2 \dots C_k$$

To illustrate, we apply this method on two sample chromosomes. Consider the following two chromosomes.

$$\begin{aligned} C1: & 9, 3, 2, 1, 4, 6, 10, 7, 5, 8, \\ C2: & 1, 5, 3, 2, 9, 7, 10, 8, 4, 6. \end{aligned} \quad (8)$$

To build two offspring O1 and O2, two positions 4 and 8 are considered on both chromosomes. Middle parts of two offspring are built using the elements between two positions 3 and 5 of two parents P1 and P2:

$$\begin{aligned} O1: & -, -, -, 1, 4, 6, 10, 7, -, -, \\ O2: & -, -, -, 2, 9, 7, 10, 8, -, -. \end{aligned} \quad (9)$$

To fill the other parts of O1 and O2, the remaining numbers of O2 and O1 are, respectively, used as circular:

List for O1, 2 is as follows: 4, 6, 1, 5, 3, 5, 8, 9, 3, 2.

To complete O1, the numbers 4, 6, and 1 are assigned to O2 because they are in the middle of O1. The numbers 5, 3, 8, 9, and 2 are assigned to O1. The remaining numbers are assigned to O2.

After crossover, the mutation operator is applied to the remaining chromosomes. The swap-based mutation is used in our proposed algorithm.

In the following, we compute the time complexity of the algorithm. Let N , P , and G represent the number of proteins, population size, and the number of generations, respectively. We have the following:

- (1) To generate the population, several chromosomes each with length N is generated in which each chromosome is a permutation of N integers. So, the order of this step is $O(N \times P) = O(NP)$.

- (2) To evaluate the chromosome, for each cluster, it is necessary to count the internal and external relationships. Hence, the order of evaluation of a chromosome is $O(N^2)$. Due to the presence of P chromosomes, the overall order of this step is $O(N^2 \times P)$.

- (3) Selection step with roulette wheel is in order $O(P)$.
- (4) The crossover for each pair will be in $O(N)$, and the mutation is a simple swap in order $O(1)$. So, this step for whole population will be in order $O(N \times P)$.

Steps 2–4 will be repeated G times. Hence, the total order is $O(NP + GN^2 \times P + GP + GN \times P)$. In this paper, P is $10N$. So, the order is $O(N^2 + GN^3 + GN + GN^3) = O(GN^3)$.

4. The Results

The PPI-GA clustering algorithm is evaluated on an interaction protein network named Collins. This network is extracted from the BioGrid [21] dataset consisting of 1879 vertices and 140849 edges, which is one of the most important and large-scale datasets in protein-protein networks. The Collins network includes 1004 proteins and 8319 interactions among proteins. The average degree on the network is 16.57 (where the degree of a node in the network is equal to the number of edges connected to a vertex), and the density of the network is 0.016 (which is defined as the number of network interactions per total number of possible edges of network). Figure 2 shows the simulation of this network in the Cytoscape software, which is modeled as an interactive graph. Cytoscape is an open-source bioinformatics software platform for visualizing molecular interaction networks.

To compare the proposed algorithm against existing algorithms, we need a gold standard to evaluate clustering algorithms. We use also CYC2008 [22] and MIPS [23] complexes as the reference clusters which are of the most important benchmarks in detecting the complexity of interacting protein networks (Table 2). Also, the cellular component ontology from Gene Ontology (GO) is utilized to evaluate the coherence of the extracted clusters.

FIGURE 2: Collins interactive network in Cytoscape application software.

TABLE 2: Data sets specifications.

Benchmark database	#complexes	#proteins
MIPS	313	1237
CYC2008	349	1627

4.1. Evaluation. To evaluate the quality of predicted clusters, the Precision (also called positive predictive value), Recall (also known as sensitivity), and F-measure evaluation metrics are employed. These criteria are calculated over pairs of nodes (here proteins). “For each pair of proteins that share at least one cluster in the overlapping clustering results, these measures try to estimate whether the prediction of this pair as being in the same cluster was correct with respect to the underlying true categories in the data (i.e., reference cluster). Precision is calculated as the fraction of pairs correctly put in the same cluster, Recall is the fraction of actual pairs that were identified, and F-measure is the harmonic mean of Precision and Recall” [24]. Let the predicted cluster and the reference cluster be denoted by C and G , respectively, and we have

$$\text{Precision} = \frac{TP}{TP \cup FP}, \quad (10)$$

$$\text{Recall} = \frac{TP}{TP \cup TN}, \quad (11)$$

$$F - \text{measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (12)$$

where true positive (TP) is the number of common proteins (vertices) between the predicted cluster C and a cluster belonging to the reference cluster G and true negative (TN) is the number of protein present in the reference cluster G but not found from the predicted cluster C . Tables 3 and 4 show clustering results achieved by PPI-GA in 20

independent runs on CYC2008 and MIPS complexes reference, respectively, in terms of Recall, Precision, and F-measure.

These tables show that considering the different metrics, the results obtained by the proposed algorithm in different independent runs are not very different. This shows that the algorithm does not have a random behavior and can always reach an acceptable solution. However, the stability of the proposed algorithm will be examined statistically in the following.

Convergence. During the evolutionary process of algorithm, the population must converge to the optimal solution over generations due to the nature of evolutionary algorithms because these algorithms evolve and improve the entire population.

Figure 3 shows an example of the convergence diagram for the Collins data sample. We carried out the algorithm 20 independent runs for each sample, and we have shown the graph of the best answer here. As it can be seen in the figure, the population of the PPI-GA has consistently converged to the best.

Stability. The stability of an algorithm indicates that the algorithm’s performance is correct. The stability of the algorithm means that the results obtained from the algorithm are not scattered so that if the algorithm runs multiple times for the same sample and under the same conditions, the quality of the results in all the performances is at one range with no much difference. To show the stability of the algorithm, we collect the results of 20 independent runs. The results obtained for the Collins data sample in the CYC2008 and MIPS as the reference clusters are shown in Figures 4 and 5 in terms of F-measure, respectively. As seen in the figures, the values obtained are in a range and do not differ greatly, and therefore the solutions obtained from the algorithm are not random and have good stability.

t -Test. This test statistically compares the average of two independent samples with a normal distribution. In other words, in this test, the averages gained from random samples are analyzed. This means that we randomly select samples from two different communities and compare the mean of them. This method is based on a normal distribution that is best used when the comparator data have a normal distribution for small samples. In this test, it is assumed that the mean of the data has no significant difference and if the variable value Sig. (2-tailed) is less than the intended error level 0.05, the hypothesis is rejected and the mean of the data has no significant difference; otherwise, there is not enough reason to reject the hypothesis.

To use this test, the data should have a normal distribution. To prove this, we employ the Kolmogorov-Smirnov test in the IBM statistics SPSS 21 software.

Given the greater amount of Sig. (2-tailed) from 0.05 in all the cases, all of the data used have a normal distribution and t -test can be used. The results of the various runs are divided into two groups (first half and the second half). We perform the t -test on the samples using the IBM statistical

TABLE 3: Collins data sample clustering results in 20 independent runs (CYC2008 complex reference).

Run number	#Clusters	Recall	Precision	F-measure
1	69	0.878185	0.619988	0.72683747
2	74	0.890057	0.620371	0.731141007
3	66	0.88961	0.602463	0.718405897
4	71	0.89932	0.630497	0.741289537
5	70	0.889715	0.620035	0.730788777
6	68	0.871675	0.609987	0.717722023
7	73	0.880472	0.639006	0.740552862
8	67	0.868712	0.601529	0.710843363
9	75	0.909976	0.611199	0.731245808
10	72	0.875676	0.637904	0.738113906
11	66	0.879535	0.601817	0.714642983
12	73	0.881598	0.630397	0.735130386
13	71	0.894737	0.62963	0.7391304
14	68	0.895417	0.610328	0.725883817
15	75	0.913626	0.623105	0.740903813
16	72	0.889877	0.628898	0.7369647907
17	69	0.881598	0.617697	0.726422348
18	74	0.900642	0.611161	0.725199456
19	70	0.894916	0.617349	0.73065967
20	67	0.886518	0.602684	0.717552307

TABLE 4: Collins data sample clustering results in 20 independent runs (MIPS complex reference).

Run number	#Clusters	Recall	Precision	F-measure
1	69	0.513214	0.350924	0.42669397
2	74	0.498912	0.353224	0.413614006
3	66	0.505571	0.360712	0.421029909
4	71	0.523451	0.345013	0.415900716
5	70	0.490345	0.336897	0.399389681
6	68	0.534512	0.341284	0.416581928
7	73	0.482342	0.345498	0.402609674
8	67	0.509812	0.355872	0.419154832
9	75	0.495587	0.330721	0.396706865
10	72	0.514067	0.347659	0.414795446
11	66	0.523789	0.334581	0.408331715
12	73	0.493218	0.358721	0.415352869
13	71	0.518116	0.363826	0.427474985
14	68	0.509837	0.342161	0.409499406
15	75	0.492217	0.339464	0.401812598
16	72	0.532116	0.348917	0.421469612
17	69	0.517153	0.329129	0.402253739
18	74	0.487103	0.366129	0.418039957
19	70	0.507113	0.348912	0.413394072
20	67	0.528541	0.353215	0.423447324

software SPSS 21. The results of this test are shown in Tables 5–8. As indicated in the tables, in all the cases, the value Sig. (2-tailed) is more than 0.05, so there is not enough reason to reject the assumption that the mean of data has no noticeable difference [25].

Therefore, considering Figures 4 and 5 as well as Tables 5–8, we can claim that the proposed algorithm is stable.

4.2. Comparison. In this subsection, we compare the proposed algorithm to some state-of-the-art algorithms. The setting of parameters is necessary for search-based algorithms. For genetic-based algorithms, for our comparisons,

we followed the algorithmic parameters setting. Algorithmic parameters are dependent on the number of proteins.

To set the probability of mutation and crossover operators, twenty different pairs of mutation and crossover probabilities have been tested on the MIPS complex reference using the proposed algorithms. For example, Figure 6 depicts the results.

We obtained the implementations of five of the selected clustering techniques—ClusterONE (<https://paccanarolab.org/cluster-one>), MCODE (<http://apps.cytoscape.org/apps/mcode>), MCL (<https://micans.org/mcl/>), GA spectral population, and GA random population—from their original authors or official web sites. For the rest of the algorithms, we used the same parameters as those used by the authors of these algorithms.

FIGURE 3: Collins convergence diagram of the data sample.

FIGURE 4: The F-measure stability diagram for the CYC2008 data sample.

FIGURE 5: The F-measure stability diagram for the MIPS data sample.

TABLE 5: The Kolmogorov–Smirnov test for the Collins data sample (CYC2008).

	<i>F</i> measure
N	20
<i>Normal Parameters^{a,b}</i>	
Mean	.7347
Std. Deviation	.01438
<i>Most Extreme Differences</i>	
Absolute	.129
Positive	.121
Negative	-.129
Kolmogorov-Smirnov Z	.576
Asymp. Sig. (2-tailed)	.894

^a Test distribution is Normal ^b Calculated from data.

Let n denote the number of proteins; the parameter setting for experiments is given in Table 9.

We compared the PPI-GA against DMCL-EHO [19], K-means [20], MCODE [9], MCL [5], ClusterOne [12], and eight algorithms presented in [3] with objectives of Density cut, Maximum cut, Normalized cut, and Ratio cut algorithms on Collins dataset and CYC2008 and MIPS reference

clusters. The results are listed in Table 10. The MCODE algorithm performs clustering based on the density of the protein network and scoring proteins, and it is one of the reference algorithms for detecting the complexes of protein networks. This table shows that the PPI-GA clustering algorithm has an accuracy of 0.63 and 0.51 in clustering the CYC2008 and MIPS references, respectively. These numbers

TABLE 6: The Kolmogorov–Smirnov test for the Collins data sample (MIPS).

	VAR00001
N	20
<i>Normal Parameters^{a,b}</i>	
Mean	.4133776652
Std. Deviation	.0090807204
<i>Most Extreme Differences</i>	
Absolute	.151
Positive	.132
Negative	-.151
Kolmogorov-Smirnov Z	.674
Asymp. Sig. (2-tailed)	.754

^a Test distribution is Normal ^b Calculated from data.

TABLE 7: The *t*-test on the Collins sample (CYC2008 reference clusters).

Group Statistics									
Number	N		Mean		Std. Deviation		Std. Error Mean		
F-measure									
1	10		0.7343		0.01541		0.00487		
2	10		0.7351		0.00446		0.00446		
Independent Samples Test									
Levene's Test for Equality of Variances					t-test for Equality of Means				
	F	Sig.	t	dif	Sig. (2-tailed)	Mean Difference	Std. Error Differences	95% Confidence Interval of the Difference	
								Lower	Upper
F-measure									
1. Equal variances assumed	0.259	0.617	−0.109	18	0.915	−0.00072	0.00660	−.01459s	0.1316
2. Equal variances not assumed			−0.109	17.857	0.915	−0.00072	0.00660	−.01460	0.1317

TABLE 8: The *t*-test on the Collins sample (MIPS reference clusters).

Group Statistics									
VAR00002	N	Mean	Std. Deviation	Std. Error Mean					
VAR00001									
1	10	0.412647703	0.0098440957	0.0031129764					
2	10	0.414107628	0.0087172708	0.0027566431					
Independent Samples Test									
Levene's Test for Equality of Variances					t-test for Equality of Means				
	F	Sig.	t	dif	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
								Lower	Upper
VAR00001									
1. Equal variances assumed	0.129	0.723	-0.351	18	0.730	-0.0014599250	0.0041580889	-0.0101957455	0.0072758955
2. Equal variances not assumed			-0.351	17.740	0.730	-0.0014599250	0.0041580889	-0.0102049179	0.0072850679

are 0.59 and 0.48 for the MCODE algorithm. For recalling, PPI-GA achieves the value of 0.89 and 0.36 compared with 0.66 and 0.27 for MCODE. The value of *F*-measure for our algorithm is obviously higher than that of the MCODE. Our proposed algorithm shows higher values for Precision, Recall, and *F*-measure in comparison to MCL which uses random walk process for clustering. For PPI-GA, MCODE, and MCL, the number of predicted clusters is close. Comparing our result with ClusterOne which is based on

network density, despite the smaller number of clusters than ClusterOne, PPI-GA presents higher values for Precision, Recall, and *F*-measure. Ramadan et al. [3] used GA and four different objective functions with the random and spectral initial population for clustering. Comparing the result of PPI-GA with [3], it is shown that PPI-GA can find better solutions in most cases. The value of Precision in [3] is 0.6 and 0.45. The value of Recall is 0.74 and 0.4 and *F*-measure is 0.66 and 0.41 for CYC2008 and MIPS reference clusters,

FIGURE 6: Parameter setting for crossover and mutation.

TABLE 9: The parameter setting for experiments.

Parameter	Values
Population size	10n
Generation	200n
Crossover rate	0.8
Mutation rate	0.05
Selection	Roulette wheel selection
Crossover operation	Order
Mutation operation	Swap mutation
Termination condition	There has been no improvement in the population for 20 iterations

respectively. It is only the Recall value that is higher for [15] than PPI-GA. Recall value is 0.4 and 0.39 for [3] in MIPS in comparison to PPI-GA which provides 0.36 for Recall value. In summary, PPI-GA has the best clustering performance in comparison to other algorithms.

To statistically analyze the results, Cliff's δ effect size metric is utilized. This test is a nonparametric effect size metric that quantifies the difference among two groups of observations (here, PPI-GA against other tested algorithms). The result of this metric is in range -1 to 1 , and higher value shows that results of the first group (here, PPI-GA) generally are better than those of the second group (other algorithms). To interpret, as [26], the following magnitudes are used: negligible ($|\delta| < 0.147$), small ($|\delta| < 0.33$), medium ($|\delta| < 0.474$), and large ($0.474 \geq |\delta|$). The results (Table 11) indicate that, in terms of Recall, Precision, and F-measure, the PPI-GA is better than the other algorithms.

Table 12 shows the running time of the proposed algorithm, PPI-GA, against other tested algorithms on MIPS and CYC2008 graphs. We ran the algorithms on a laptop with Intel core i7 processor 2.60 GHz and 16 GB of memory. On both graphs, the PPI-GA has a shorter running time compared with DMCL-EHO, K-means clustering, Ramadan-spectral population-DC (denoted by RS-DC), and Ramadan-random population-DC (denoted by RR-DC). It is important to note that MCODE, MCL, and ClusterOne use a greedy algorithm to cluster the graphs; hence, they have less execution time than search-based and evolutionary algorithms. K-means can take varying numbers of clusters as input. For K-means, the algorithm has been executed with different values of k in steps of 5 increment, up to a specified runtime.

In terms of the quality of the answers, the proposed algorithm outperforms the algorithms that use greedy methods and is close to them in terms of execution time. As shown in Figure 3 and Table 12, the proposed algorithm has an acceptable speed in reaching an acceptable answer, and this is due to the coding presented in Section 3. The provided coding greatly reduces the search space.

4.3. Clustering Quality. To evaluate the clusters' biological significance in the Collins network, we evaluate the PPI-GA algorithm with components in the Gene Ontology. The GO term finder is applied to obtain the most significant GO terms. The p value is generally applied to measure the probability of enriching a predicted complex via a given GO term by chance. This function is calculated via utilizing the software modules called SGD GO Term Finder (<http://search.cpan.org/dist/GO-TermFinder/>) [27]. The p value is defined via the hypergeometric distribution as follows:

$$p - \text{value} = 1 - \sum_{i=0}^{k-1} \frac{\binom{M}{i} \binom{N-M}{n-i}}{\binom{N}{n}}, \quad (13)$$

where N is number of proteins in the PPI network, M is size of a list of proteins, G marks to the reference term of interest (the number of proteins in a GO term), n is the number of proteins that are annotated with the same GO term, k is the number of proteins in an extracted cluster C , and i is the number of proteins shared between C and G . Therefore, the p value closer to zero, the more biologically significant the predicted complexes make. The biological significance of a group is settled by employing a cutoff value to determine significant from insignificant groups. Merely matches with p value below 0.05 are statistically remarkable.

Table 13 collects some of the Collins network' clusters that have a considerable p value. As you can see, the percentages (N%) are 100 for most clusters, showing that these clusters in the network fit well with the corresponding GO components. Table 13 also clearly demonstrates that genetic algorithm-based methods can identify protein complexes.

The proportion of significant complexes to total predictions is used to evaluate the overall performance of an algorithm. Table 14 shows the comparison results based on this measure. The PPI-GA algorithm has identified 71 protein complexes, of which 67 protein complexes contain gene ontology annotations. In Table 14, the majority of our predicted complexes (94.36%) are significant and our algorithm also predicts a higher proportion of significant complexes than other algorithms. Meanwhile, Ramadan GA, ClusterOne, and K-means algorithms predict many protein complexes with extremely small size (e.g., with two proteins) and generally predicted complexes with small size tend to have large p value. Therefore, Ramadan GA, ClusterOne, and K-means algorithms only predicted a small proportion of significant complexes. These results are also consistent with the results in Table 11 where these algorithms achieve low

TABLE 10: Comparison of PPI-GA algorithm results with existing algorithm results.

Method	#Clusters	CYC2008			MIPS		
		Recall	Precision	F-measure	Recall	Precision	F-measure
PPI-GA	71	0.89	0.63	0.74	0.36	0.51	0.42
DMCL-EHO	85	0.74	0.61	0.66	0.42	0.41	0.41
K-means clustering	120	0.55	0.67	0.60	0.28	0.40	0.32
MCODE	54	0.66	0.59	0.63	0.27	0.48	0.35
MCL	75	0.65	0.45	0.54	0.27	0.34	0.30
ClusterOne	114	0.55	0.43	0.49	0.20	0.34	0.25
<i>Ramadan GA spectral population</i> ¹							
(a) DC	162	0.74	0.6	0.66	0.32	0.45	0.37
(b) MC	180	0.71	0.47	0.6	0.38	0.40	0.39
(c) NC	193	0.67	0.5	0.57	0.39	0.37	0.37
(d) RC	161	0.73	0.38	0.5	0.39	0.33	0.36
<i>Ramadan GA random population</i> ¹							
(a) DC	164	0.72	0.54	0.62	0.3	0.41	0.35
(b) MC	162	0.71	0.45	0.56	0.4	0.35	0.38
(c) NC	138	0.66	0.57	0.61	0.36	0.44	0.41
(d) RC	154	0.61	0.55	0.58	0.34	0.43	0.38

¹DC: Density cut; MC: Maximum cut; NC: Normalized cut; RC: Ratio cut.

TABLE 11: Cliff's δ effect size test.

Algorithm	Recall	Precision	F-measure
DMCL-EHO	.0 (neg.)	.5 (large)	.5 (large)
K-means	.5 (large)	.0 (neg.)	.5 (large)
MCODE	.5 (large)	.5 (large)	.5 (large)
MCL	.5 (large)	1 (large)	.5 (large)
ClusterOne	.5 (large)	1 (large)	.5 (large)
Ramadan GA spectral-DC	.5 (large)	.5 (large)	.5 (large)
Ramadan GA spectral-MC	.0 (neg.)	1 (large)	.5 (large)
Ramadan GA spectral-NC	.0 (neg.)	1 (large)	.5 (large)
Ramadan GA spectral-RC	.0 (neg.)	1 (large)	.5 (large)
Ramadan GA random-DC	.5 (large)	.5 (large)	.5 (large)
Ramadan GA random-MC	.0 (neg.)	1 (large)	.5 (large)
Ramadan GA random-NC	.25 (small)	.5 (large)	.5 (large)
Ramadan GA random-RC	.5 (large)	.5 (large)	.5 (large)

A positive value indicates the effect size favor of the PPI-GA. The interpretation of the effect size is indicated in parenthesis; neg. stands for negligible; med. stands for medium.

TABLE 12: Comparison of average execution time of the proposed algorithm with the existing algorithms in terms of time (minutes).

	PPI-GA	DMCL-EHO	K-means Clustering	MCODE	MCL	ClusterOne	RS-DC	RR-DC
MIPS	61	68	130 [†]	59	49	53	112	97
CYC2008	102	123	190 [†]	134	98	110	159	140

RS-DC and RR-DC indicate Ramadan-spectral population-DC and Ramadan-random population-DC, respectively. [†] scores in Table 10 show the intermediate clustering results obtained at that time.

scores. In addition, Table 13 shows 14 protein complexes with very low p value, predicted by our PPI-GA algorithm.

5. Threats to Validity

In this section, to clarify the validity of PPI-GA, the limitations that can affect the results of the algorithm are discussed. Several factors may bias the validity of the study. These are typically divided into two categories external and internal validity. External validity is about the ability to generalize the results to other than used case studies or indifferent settings for them.

- (i) The input of the algorithm is a PPI graph, and as other works, in this paper, inter-relationships and intrarelations are considered as an indicator for PPI clustering. These indicators may not be a good indicator for clustering protein-protein networks or at least may not be sufficient and may require other indicators. However, there is no research that discusses which indicators can improve the quality of PPI clustering.
- (ii) In search-based techniques for PPI clustering, generalizing a technique to any protein-protein interaction is an important threat to the validity of

TABLE 13: Some clusters in the Collins network with the lowest p value with GO component.

	Size	GO-ID/GO-term	P value	N%
1	17	"GO:0030880/RNA polymerase complex"	3.30986E-39	100%
2	8	"GO:0044428/Nuclear part"	3.70274E-05	100%
3	27	"GO:0005739/Mitochondrion"	9.82318E-22	100%
4	18	"GO:0000502/Proteasome complex"	1.76807E-40	100%
5	7	"GO:0030008/TRAPP complex"	1.02802E-20	100%
6	10	"GO:0044425/Membrane part"	4.18992E-10	100%
7	5	"GO:0035097/Histone methyl-transferases complex"	1.31389E-11	100%
8	5	"GO:0030126/COPI vesicle coat"	1.18247E-14	100%
9	15	"GO:0000502/Proteasome complex"	2.20275E-33	100%
10	12	"GO:0016514/SWI/SNF complex"	4.98150E-37	100%
11	60	"GO:0005634/Nucleus"	2.15384E-32	100%
12	81	"GO:0043227/Membrane-bound organelle"	4.87516E-23	98.4%
13	63	"GO:0044464/Cell part"	3.42642E-05	95.4%
14	4	"GO:0031011/INO80 complex"	4.13601E-07	79.0%

TABLE 14: Comparison of various algorithms in terms of significance of the predicted complexes for Collins data.

Method	#Predicted complexes	#Significant complexes	Proportion (%)
PPI-GA	71	67	94.36
DMCL-EHO	85	77	90.58
K-means clustering	120	88	73.33
MCODE	54	48	88.88
MCL	75	37	49.33
ClusterOne	114	85	74.56
<i>Ramadan GA spectral population¹</i>			
(a) DC	162	150	92.59
(b) MC	180	148	82.22
(c) NC	193	135	69.94
(d) RC	161	110	68.32
<i>Ramadan GA random population¹</i>			
(a) DC	164	143	87.19
(b) MC	162	113	69.75
(c) NC	138	124	89.85
(d) RC	154	120	77.92

¹DC: Density cut; MC: Maximum cut; NC: Normalized cut; RC: Ratio cut.

results. So, in this paper, CYC2008 and MIPS datasets, as popular and large datasets, are selected for evaluation and comparison.

Internal validity is concerned with experimental treatments that affect the algorithm results, leading to poor results.

- (i) Similar to other encoding methods, duplicate solutions can be generated in the proposed encoding. This may increase the search time in the evolution process of the algorithm. Another threat is that the probability of producing all solutions in the proposed encoding is not the same. This may lead to solutions that are more likely to produce solutions.
- (ii) In this paper, metrics such as Precision, Recall, and F-measure are used to compare study results with the current PPI clustering algorithms. Other metrics may produce different results.
- (iii) The related rate of crossover and mutation operators used in GA is achieved from several experiments on

the CYC2008 and MIPS datasets. However, these numbers may not work well on other datasets.

6. Conclusion

Detecting protein complexes is an important analysis method which enables us to understand cellular structures and biological functions within PPI networks. In this paper, we proposed a new approach to detect and predict protein complexes in PPI networks. We proposed a multiobjective function to maximize cluster intra-connections and minimize cluster interconnections. The results showed that the objective function had a better performance compared with other approaches. Our clustering approach is more accurate than MCODE, MCL, ClusterOne, and Ramadan et al.'s algorithms so that it returns better output and higher F-score. In the future, we suggest that the proposed approach can be used for other biological networks such as metabolic networks. The proposed method can be improved for overlapping

proteins (so that a protein may belong to several clusters). We will try to apply some of the methods presented in [28] to the PPI networks clustering problem and also use other existing methods such as [29, 30] instead of genetic algorithms.

Data Availability

The data used to support this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

References

- [1] Y. Zhang, H. Lin, Z. Yang, J. Wang, Y. Li, and B. Xu, "Protein complex prediction in large ontology attributed protein-protein interaction networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 10, no. 3, pp. 729–741, 2013.
- [2] J. Zahiri, A. Emamjomeh, S. Bagheri et al., "Protein complex prediction: a survey," *Genomics*, vol. 112, no. 1, pp. 174–183, 2020.
- [3] E. Ramadan, A. Naef, and M. Ahmed, "Protein complexes predictions within protein interaction networks using genetic algorithms," *BMC Bioinformatics*, vol. 17, no. 7, p. 269, 2016.
- [4] Z. Wu, Q. Liao, and B. Liu, "A comprehensive review and evaluation of computational methods for identifying protein complexes from protein-protein interaction networks," *Briefings in Bioinformatics*, vol. 21, no. 5, pp. 1531–1548, 2020.
- [5] S. V. Dongen, "A cluster algorithm for graphs," Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science, Amsterdam, The Netherlands., 2000.
- [6] S. Brohee and V. Helen, "Evaluation of clustering algorithm for protein-protein interaction networks," *BMC Bioinformatics*, vol. 7, no. 1, p. 488, 2004.
- [7] N. Przulj, D. A. Wigle, and I. Jurisica, "Functional topology in a network of protein interactions," *Bioinformatics*, vol. 20, no. 3, pp. 340–348, 2004.
- [8] R. Dunn, F. Dudbridge, and C. M. Sanderson, "The use of edge between clustering to investigate biological function in protein interaction network," *BMC Bioinformatics*, vol. 6, no. 1, p. 39, 2005.
- [9] G. Bader and C. Hoguean, "Automated method for finding molecular complexes in large protein interaction networks," *BMC Bioinformatics*, vol. 4, no. 1, p. 2, 2003.
- [10] P. Jiang and M. Singh, "SPICi: a fast clustering algorithm for large biological networks," *Bioinformatics*, vol. 26, no. 8, pp. 1105–1111, 2010.
- [11] M. Atlaoui-Ul-Amin, Y. Shinbo, K. Mihara, K. Kurokawa, and S. Kanaya, "Development and implementation of an algorithm for detection of protein complexes in large interaction networks," *BMC Bioinformatics*, vol. 7, no. 1, p. 207, 2006.
- [12] T. Nepuse, H. Yu, and A. Paccanaro, "Detection overlapping protein complexes in the protein-protein interaction network," *Nature Methods*, vol. 9, no. 5, p. 471, 2012.
- [13] 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, 2006.
- [14] C. Pizzuti and S. E. Rombo, "An evolutionary restricted neighborhood search clustering approach for PPI networks," *Neurocomputing*, vol. 145, pp. 53–61, 2014.
- [15] B. Cao, J. Luo, C. Liang, S. Wang, and D. Song, "MOEPGA: a novel method to detect protein complexes in yeast protein-protein interaction networks based on MultiObjective Evolutionary Programming Genetic Algorithm," *Computational Biology and Chemistry*, vol. 58, pp. 173–181, 2015.
- [16] X. Lei, M. Fang, and H. Fujita, "Moth-flame optimization-based algorithm with synthetic dynamic PPI networks for discovering protein complexes," *Knowledge-Based Systems*, vol. 172, pp. 76–85, 2019.
- [17] A. Sikandar, W. Anwar, U. I. Bajwa et al., "Decision tree based approaches for detecting protein complex in protein protein interaction network (PPI) via link and sequence analysis," *IEEE Access*, vol. 6, pp. 22108–22120, 2018.
- [18] L. Gu, Y. Han, C. Wang, W. Chen, J. Jiao, and X. Yuan, "Module overlapping structure detection in PPI using an improved link similarity-based Markov clustering algorithm," *Neural Computing and Applications*, vol. 31, no. 5, pp. 1481–1490, 2019.
- [19] R. R. Rani, D. Ramyachitra, and A. Brindhadevi, "Detection of dynamic protein complexes through Markov clustering based on elephant herd optimization approach," *Scientific Reports*, vol. 9, no. 1, pp. 1–18, 2019.
- [20] S. Kalaivani, D. Ramyachitra, and P. Manikandan, "K-means clustering: an efficient algorithm for protein complex detection," in *Progress in Computing, Analytics and Networking*, pp. 449–459, Springer, Singapore, 2018.
- [21] T. Reguly, "Comprehensive curation and analysis of global interaction networks in *saccharomyces cerevisiae*," *Journal of Biology*, vol. 5, no. 4, p. 11, 2006.
- [22] H. W. Mewes, "MIPS: a database for genomes and protein sequences," *Nucleic Acids Research*, vol. 30, no. 1, pp. 31–34, 2002.
- [23] H. W. Mewes, "The bioinformatics of the yeast genome-A historical perspective," *Yeast*, vol. 36, no. 4, pp. 161–165, 2019.
- [24] P. N. Tan, M. Steinbach, and V. Kumar, "Data mining cluster analysis: basic concepts and algorithms," *Introduction to Data Mining*, vol. 13, pp. 487–533, 2013.
- [25] F. A. Paniagua, "The null hypothesis is always rejected with statistical tricks: why do you need it?" *Interamerican Journal of Psychology*, vol. 53, no. 1, pp. 17–27, 2019.
- [26] N. Teymourian, H. Izadkhah, and A. Isazadeh, "A fast clustering algorithm for modularization of large-scale software systems," *IEEE Transactions on Software Engineering*, p. 1, 2020.
- [27] C. Pizzuti and S. E. Rombo, "A coclustering approach for mining large protein-protein interaction networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 3, pp. 717–730, May. 2012.
- [28] B. Pourasghar, H. Izadkhah, A. Isazadeh, and S. Lotfi, "A graph-based clustering algorithm for software systems modularization," *Information and Software Technology*, vol. 133, Article ID 106469, 2021.
- [29] S. Pourbahrami and S. Azimpour, "A new method for detection of clustering based on four zones Apollonius circle," *Iran Journal of Computer Science*, vol. 3, no. 1, pp. 59–64, 2020.
- [30] M. Bodaghi and K. Samieefar, "Meta-heuristic bus transportation algorithm," *Iran Journal of Computer Science*, vol. 2, no. 1, pp. 23–32, 2019.