IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 January 2024, accepted 11 February 2024, date of publication 20 February 2024, date of current version 28 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3367746

RESEARCH ARTICLE

An Evolutionary Algorithm With Heuristic Operator for Detecting Protein Complexes in Protein Interaction Networks With Negative Controls

MUSTAFA N. ABBAS[1], BARA'A A. ATTEA[2], DAVID BRONESKE[3], AND GUNTER SAAKE[1]
[1]Research Group Databases and Software Engineering, Otto-von-Guericke-University Magdeburg, 39016 Magdeburg, Germany
[2]College of Science, University of Baghdad, Baghdad 10047, Iraq
[3]German Centre for Higher Education Research and Science Studies, 30159 Hannover, Germany

Corresponding author: Bara'a A. Attea (bara.a@sc.uobaghdad.edu.iq)

This work was supported by Otto-von-Guericke-University Magdeburg.

ABSTRACT Computational biology research faces a formidable challenge in the detection of complexes within protein-protein interaction (PPI) networks, critical for unraveling cellular processes, predicting functions of uncharacterized proteins, and diagnosing diseases. While evolutionary algorithms (EAs), particularly state-of-the-art methods, often partition PPI networks based on graph properties or biological semantics, their resilience to noisy or missing interactions remains an underexplored territory. In this paper, we propose a groundbreaking heuristic operator, termed ''strong neighbor-node migration'', specifically designed to elevate solution quality during the evolutionary process of our proposed EA. Through the application of EAs, we systematically evaluate the robustness of three single-objective models and two multi-objective models dedicated to addressing the complex detection problem. Our comprehensive assessment spans three well-known PPI networks, including two Saccharomyces cerevisiae datasets and the Human Protein Reference Database. To challenge the models further, we generate artificial networks by introducing varying percentages of noise to the original PPI networks. The experimental results showcase the superiority of the multi-objective model that incorporates our novel heuristic operator, demonstrating enhanced prediction accuracy compared to state-of-the-art models. Encouragingly, we advocate for the expansion of this research to integrate biological information, such as gene ontology. We propose the development of an objective function and heuristic operator based on this biological data, aiming to advance protein complex detection.

INDEX TERMS Evolutionary algorithm, multi-objective optimization, heuristic operator, protein-protein interaction network, complex detection.

I. INTRODUCTION

Proteins are a vital part and building blocks of every living organism. They are composed of amino acids, which make up a polypeptide chain, and they encode the information stored in genes. The main functions of a living organism are carried out or regulated by proteins that interact with one another within a cell or in vitro [3], [49], [52]. Due to the growth of bioinformatics, biochemical, and related fields of study, high-throughput experimental methods such as proteomics, metabolomics, and phenomics have become more prominent in recent decades. As a result of these methods, massive amounts of experimental data reflecting the interaction between proteins in complex networks (i.e., cellular networks) on protein-protein interaction (PPI) networks (i.e., yeast two-hybrid tests, or Y2H) are gathered to describe different protein structures and how they interact [28], [51]. However, protein interactions so far are still suffering from spurious interactions, as well as missing interactions [23]. Moreover, interactions with low confidence values may be discarded in subsequent analyses. However, different topological metrics and link prediction algorithms [12], [27]

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.

can be used to score false-negative interactions and contribute the top-scored ones to PPI networks [38]. There is a general agreement among biologists that proteins that are closely located to one another in the PPI network are perform similar functions, and genes that are regulated by the same transcription factors likely to have activities that are substantially similar to one another (genes causing similar diseases). In this scenario, perturbations in the interactions may cause the same disease or disease phenotype [8], [39].

In the PPI networks, the detection of protein complexes, or functional modules, is an ongoing challenge, but it is crucial for revealing the mechanism of biological functions and providing a valuable guide for comprehending the processes controlling cell life. In addition, detecting protein complexes can be useful for defining the evolutionary orthology signal, such as for the prediction of protein functions based on their biological functions that have not yet been identified, and most crucially, for medical uses [24], [46]. It is noteworthy to mention that proteins that interact with one another can be categorized as either ''protein complexes'' or ''functional modules,'' each of which having a distinct biological significance. Unlike to functional modules, protein complexes are composed of a number of protein molecules that work in a common location to perform a certain function. Protein complexes characterized by large transcription factors, anaphase-promoting complexes, Ribonucleic Acid (RNA) splicing machinery, protein export and transport machinery, and others can be identified. In contrast, a functional module is a collection of proteins that are involved in a particular cellular compartment and interact with each other in order to perform a specific biological function at different times and places (different phases of the cell cycle, different cellular compartments). There are several functional modules that can be identified. They include the cyclin module for determining the progression of a cell cycle, yeast pheromone response pathways, and Mitogen-Activated Protein (MAP) signaling pathways.

The protein complex detection problem in PPI networks is conventionally tackled through the application of clustering methods. The primary goal of clustering is to unveil the intricate topology, features, and functions embedded within PPI networks. However, finding the global optimal solution to the PPI network clustering problem has shown to be a non-deterministic polynomial-time hard (NP-hard) problem [13], [48]. This paper provides a comprehensive overview of clustering methods applied to PPI networks, categorizing them into distance-based (topology-free) and topology-based approaches.

- Distance-Based (Topology-Free) Clustering: This category encompasses methods (such as [2], [41]), which cluster proteins based on their distances from one another, adopting a topology-free paradigm.
- Topology-Based Clustering: Methods falling under this category leverage graph-based approaches, considering the overarching topology of the PPI network, (such as [4], [6], [17], [20], [47]).

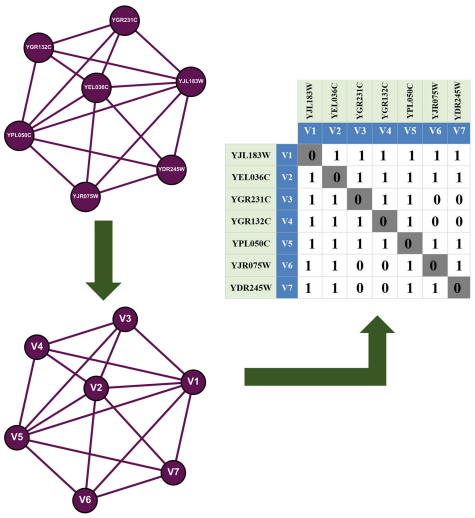
While these clustering methods significantly contribute to unraveling the complex organization of PPI networks, the computational challenges associated with NP-hard problems necessitate innovative approaches. This is where the importance of evolutionary algorithms (EAs) comes into play. The utility of EAs in this context becomes evident when considering the dynamic nature of biological systems and the need for adaptive strategies. In contrast to static clustering approaches, EAs are inherently designed to evolve solutions over multiple iterations, enabling them to adapt and optimize in response to changing conditions. This dynamic adaptability is particularly crucial in PPI network analysis, where the interactions among proteins can vary, and the completeness of data is often compromised by noise or missing information. Given that protein complex detection problem comes into the category of NP-hard problems, a recent study [9], [31], [47] revealed that metaheuristic and evolutionary algorithms are very competitive compared to state-of-the-art methods. Unfortunately, up to now, little interest has been paid to investigate the robustness of these state-of-the-art EAs in unraveling PPI networks with noisy or missing interactions. In this study, the main contribution is to examine and evaluate the effectiveness of the EA for detecting protein complexes within PPI networks with spurious and missing interaction data. To achieve this contribution, both single and multi-objective EAs are adopted to examine the robustness of three well-known single objective models and two multi-objective models that are used to define the complex detection problem.

A. PRELIMINARY CONCEPTS

In this section, we will explain some principles related to graphs that are used in PPI networks, and we will explain these principles using common formal expressions. The PPI network can be viewed as a complex cellular network $\mathcal{N}(\mathbb{P}, \mathbb{E})$, where $\mathbb{P}$ is a set of n different proteins, that is, $\mathbb{P} = \{P_1, P_2, \ldots, P_n\}$, and a set of m mutual interaction between any pair of proteins in $\mathcal{N}$ is composed of undirected edges (P_i, P_j) in $\mathbb{E}$, i.e., $\mathbb{E} = \{E_1, E_2, \ldots, E_m\}$.

In terms of mathematics, $\mathcal{N}$ can be expressed as a graph $\mathcal{G}(\mathbb{V}, \mathbb{E})$ with a set of nodes $\mathbb{V}$, where $\mathbb{V} = \{V_1, V_2, \ldots, V_n\}$, and a set of edges $\mathbb{E}$. In an undirected graph $\mathcal{G}$, an edge between nodes V_i and V_j can be used in both direction. Thus, if V_i and V_j are connected, then V_j and V_i are also connected. Proteins, RNA molecules, and gene sequences are represented as nodes in a graph $\mathbb{V}$. The graph edges $\mathbb{E}$ describe (physical, biochemical, or functional) interactions. Any edge $E \in \mathbb{E}$ can also be expressed as the pair (V_i, V_j), where V_i, and V_j correspond, respectively, to two interacting proteins P_i, and P_j in $\mathcal{N}$. The number of interactions that include a given protein, P_i, is denoted by its degree, $d(V_i) = |(V_i, V_j)|(V_i, V_j) \in \mathbb{E}|$.

The symmetric adjacency matrix, often known as the connection matrix, $\mathbf{A} = [a_{ij}]^{n \times n}$, is an example of a common way to describe an undirected graph ($\mathcal{G}$). This matrix contains all of the connections between the nodes in the graph, where



		YJL183W	YEL036C	YGR231C	YGR132C	YPL050C	YJR075W	YDR245W
		V1	V2	V3	V4	V5	V6	V7
YJL183W	V1	0	1	1	1	1	1	1
YEL036C	V2	1	0	1	1	1	1	1
YGR231C	V3	1	1	0	1	1	0	0
YGR132C	V4	1	1	1	0	1	0	0
YPL050C	V5	1	1	1	1	0	1	1
YJR075W	V6	1	1	0	0	1	0	1
YDR245W	V7	1	1	0	0	1	1	0

FIGURE 1. Seven proteins form Saccharomyces cerevisiae and their adjacency matrix.

two protein pairs (P_i and P_j) are adjacent, if they interact with each other (i.e., $a_{ij} = 1$ and $a_{ji} = 1$). Otherwise, ($a_{ij} = 0$ and $a_{ji} = 0$). Figure 1 depicts an example of an adjacency matrix representing a sample of PPI network relationship, where each element in the matrix is equal to 1, if P_i has an interaction with P_j; otherwise, it is equal to 0.

B. EA: SINGLE AND MULTI-OBJECTIVE

EAs imitate the mechanism of natural evolution by adopting heuristic search and optimization techniques. The concepts of competition, selection, reproduction, and random perturbation in evolution are all referred to in the same way by both natural evolution and EAs. The main purpose of the EA is to improve the fitness of a population of possible solutions. EAs are characterized by the fact that they partition the search space associated with an optimization problem, $F(X)$, into a finite set of points and then operate on a very tiny arbitrary subset of those points. This subset of points is referred to as the population of individuals. The population is described by $\mathbb{I}^{\mu} = (I_1, I_2, \ldots, I_{\mu})$. A fitness function, $F(I)$, evaluates different regions in the search space using the processed population. The composition of the EA's selection, recombination, and mutation operators are used to transform the population, $Trans : I^{\mu} \rightarrow \mathbb{I}^{\mu}$. Until a stop condition is reached, transformations are done in a general loop to generate succeeding populations, $\iota : \mathbb{I}^{\mu} \rightarrow \{true, false\}$. The selection operator, Θ_s, considers individuals from better regions of the search space. By crossing two elements of the selected population, the recombination operator (also called crossover), Θ_r, produces two new individuals. Last but not least, the mutation operator, Θ_m, locates and explores new regions of search by occasionally modifying the selected individuals. The general EA framework is sketched Algorithm 1.

Numerous real-world issues have multiple, commonly conflicting, objectives, where addressing them needs to be fair and satisfying at the same time. To this end, several

Algorithm 1 The General Framework of EA

Input: $\mu, \Theta_s, \Theta_r, \Theta_m, p_c, p_m, \iota$
Output: optimal I^*;
$t \leftarrow 0$;
Initialize population $\mathbb{I}^{\mu}(t) \leftarrow (I_1, I_2, \ldots, I_{\mu})$;
for $i \leftarrow i \in (1, 2, \ldots, \mu)$ **do**
  Evaluate $F(I_i(t))$;
end for
while $\iota(\mathbb{I}^{\mu}(t)) \neq true$) **do**
  $t \leftarrow t + 1$;
  for $i \leftarrow i \in (1, 2, \ldots, \mu)$ **do**
    $I_{i,1}(t) \leftarrow \Theta_s(\mathbb{I}^{\mu}(t-1))$
    $I_{i,2}(t) \leftarrow \Theta_s(\mathbb{I}^{\mu}(t-1))$
    $I'_i(t) \leftarrow \Theta_r(I_{i,1}(t), I_{i,2}(t), p_c)$;
    $I'_i(t) \leftarrow \Theta_m(I_i(t), p_m)$;
    Evaluate $F(I'_i(t))$;
  end for
end while
Return $I^*(t)$

optimization algorithms with multiple objectives (MOOs) are formulated. Rather than finding a single optimal solution, a MOO algorithm allows us to obtain many solutions that are characterized as non-dominant solutions, and this provides the decision maker with the optimal trade-off solutions among these contradictory objectives.

Mathematically, suppose that the MOO problem has n variables drawn from the multi-objective universe Ω. Let $X = (x_1, x_2, \ldots, x_n) \in \Omega$, where $\Omega \in \mathbb{R}^n$, and the objective functions $F(X) = (f_1(x), f_2(x), \ldots, f_{\mathbb{k}}(x))$, where $\mathbb{k}$ refers to the number of objective functions, such that $F : \Omega \rightarrow \mathbb{R}^{\mathbb{k}}$. The optimization of the objective function $F(X)$ is accomplished by obtaining a set of solutions, wherein not any solution is dominated by any other solution in the set. These non-dominated solutions $X^* = (x_1^*, x_2^*, \ldots, x_n^*) \in \Omega$ are collected and placed in a repository known as the Pareto set (PS). To clarify the non-dominated solutions, suppose there are two vectors, $\mathcal{U} = (u_1, u_2, \ldots, u_n)$ and $\mathcal{V} = (v_1, v_2, \ldots, v_n)$, both in Ω. If neither $mathcalU$ nor $mathcalV$ is superior to the other, then the two solutions are said to be mutually non-dominating, or $\mathcal{U}$ and $\mathcal{V}$ are assumed to be equal. Consequently, a decision subspace $\overline{\Omega}$ within the universe Ω that contains solutions that are not dominated by any other solution can be referred to as a set of non-dominated solutions. This definition can be used to describe a non-dominated set of solutions.

In this study, two well-known multi-objective evolutionary algorithms (MOEAs) are adopted. The first is the non-dominated sorting genetic algorithm (NSGA-II) proposed by Deb et al. [18]. Consider an MOO problem with k objective functions formulated in Eq. 1. The definition of genetic algorithm operators is expanded to take into account two characteristics. First, based on the non-dominated fitness assignment, the individuals of the population $\mathbb{I}$ are sorted

into fronts. Secondly, maintain the fitness contrast between individuals of the same non-dominated front.

$$\min\ F(X) = f_1(X), f_2(X), \ldots, f_{\mathbb{k}}(X) \tag{1}$$

The mechanism for selecting individuals is done by arranging the individuals depending on the rank of the individuals in the Pareto-front, and niching schemes. A distance is calculated for each individual, which is called the crowding distance, as all these individuals are ranked depending on the non-domination. However, all of these non-dominated individuals are grouped together into a single front and allocated a dummy maximal fitness. After that, this front of classed individuals is neglected, and another front of individuals is taken into consideration for the position of the next front. This process of classification continues until all the individuals have been categorized into the proper fronts. During the phases of selection and population front-classification, mutually non-dominant solutions are evaluated on the basis of their contribution to population diversity. However, individuals on the first front always get more copies since they possess the highest fitness value. By searching for non-dominated regions through successive generations t, the population is converged towards these regions. As a result of NSGA-II, the non-dominated solution set of individuals is not explicitly archived. Algorithm 2 describes the framework of NSGA-II.

The decomposition-based MOEA (MOEA/D) algorithm proposed by Zhang et al. [54] is the second well-known algorithm that has been effectively applied to a variety of real-world problems. Assuming that MOEA/D algorithm has found the optimum solution, which is Z_i^* for each objective $i \in [1, 2, \ldots, \mathbb{k}]$, then a reference point $Z^* = (z_1^*, z_2^*, \ldots, z_{\mathbb{k}}^*)$ can be used to represent a vector of the best possible solutions for the objectives. In a more formal context, $i \in [1, 2, \ldots, \mathbb{k}]$:

$$z_i^* = f_i(X^* \in \overline{\Omega}) :\Leftrightarrow: \nexists X \in \Omega | f_i(X) < f_i(X^*)| \ \overline{\Omega} \subset \Omega \tag{2}$$

the population $\mathbb{I}^\mu = (I_1, I_2, \ldots, I_\mu)$ is the notation that is utilized in order to describe μ distinct scalar optimization sub-problems. Each point in the search space Ω is considered to be a distinct sub-problem for the scalar optimization of $\mathbb{k}$ objectives. In other words, this is to break the MOO problem down into *mu* sub-problems using the MOEA/D algorithm. Each individual $I_i | i \in [1, 2, \ldots, \mu]$ is attached with one weight vector w_i from a collection of μ evenly distributed weight vectors $W = (w_1, w_2, \ldots, w_\mu)$. Recall, the value of k represents the total number of objective functions for the MOO problem, and define the weight vector associated with each I_i as follows: $w_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,\mathbb{k}})$

$$\sum_{i=1}^{\mu} \sum_{j=1}^{\mathbb{k}} w_{i,j} = 1 \tag{3}$$

In addition, it is feasible for each individual I_i, $1 <= i <= \mu$, to expand utilizing information obtained directly from the solutions that are located in its immediate neighborhood g. Neighboring solutions to I_i, denoted by $G_i = (I_1, I_2, \ldots, I_{i,g})$, are those g with the closest distance weight vectors using euclidean distance to w_i, as expressed in Eq. 4. Then, a vector of neighbor solutions to the whole population $\mathbb{I}^\mu$ is $\mathbb{G}^\mu = (G_1, G_2, \ldots, G_\mu)$.

Algorithm 2 The General Framework of NSGA-II

```
Input: μ, Θ_s, Θ_r, Θ_m, p_c, p_m, ι
Output: Population 𝕀^μ with fronts
t ← 0;
Initialize population 𝕀^μ(t) ← (I_1, I_2, ..., I_μ);
for i ← i ∈ (1, 2, ..., μ) do
    Evaluate F(I_i(t)) ← (f_1(I_i(t)), f_2(I_i(t)), ..., f_𝕜(I_i(t)));
end for
for i ← i ∈ (1, 2, ..., μ) do
    Rank assignment of r(I_i) based on the Pareto rank of dominance;
end for
while ι(𝕀^μ(t)) ≠ true) do
    for i ← i ∈ (1, 2, ..., μ) do
        I_{i,1}(t) ← Θ_s(𝕀^μ(t))
        I_{i,2}(t) ← Θ_s(𝕀^μ(t))
        I'_i(t) ← Θ_r(I_{i,1}(t), I_{i,2}(t), p_c)
        I'_i(t) ← Θ_m(I_i(t), p_m
        Evaluate F(I'_i(t));
    end for
    𝕀^μ(t) ← 𝕀^μ(t) ∪ 𝕀^{r^μ}(t)
    for i ← i ∈ (1, 2, ..., 2μ) do
        Assign rank r(I_i) based on Pareto dominance sort;
    end for
    calculate crowding distance of I in each front
    Determine which 𝕀^μ(t) value to select depending on rank and crowding distance.
    t ← t + 1;
end while
Return 𝕀*(t)
```

$$I_j \in G_i : \Leftrightarrow \nexists(w_l \in W \wedge I_l \notin G_i)$$

$$\sum_{x=1}^{\mathbb{k}} (w_{i,x} - w_{j,x})^2 > \sum_{x=1}^{\mathbb{k}} (w_{i,x} - w_{l,x})^2 \tag{4}$$

$$D_i^{te}(I_i | w_i, Z^*) = \min_{1<=j<=\mathbb{k}} w_{i,j} | f_j(I_i) - z_j^* | \tag{5}$$

In addition, if the value of Z^* is greater than the value of $f_j(I_i)$, then the value of $f_j(I_i)$ will be substituted for Z^* at the reference point $Z^* = (z_1^*, \ldots, z_{\mathbb{k}}^*)$. Also, the content of the non-dominated archive $\mathbb{I}^*$ can also be influenced by I_i. This ensures that all the solutions I^* that are dominated by I_i are removed from the archive, and if there is no I^* in the archive that dominates I_i, it is finally added to the archive. The general framework of the MOEA/D algorithm can be expressed in Algorithm 3.

II. RELATED WORK

With the beginning of the 21st century, numerous researchers have shown significant interest in the topic of complex

Algorithm 3 The General Framework of MOEA/D

Input: $\mu, \Theta_s, \Theta_r, \Theta_m, p_c, p_m, \iota$
Output: non-dominated $\mathbb{I}^*$
Initialize $W \leftarrow (w_1, w_2, \ldots, w_\mu)$; /* Where $w_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,\mathbb{k}})$ */
$t \leftarrow 0$;
Initialize population $\mathbb{I}^\mu(t) \leftarrow (I_1, I_2, \ldots, I_\mu)$;
Initialize neighbors $\mathbb{G}^\mu(t) \leftarrow (G_1, G_2, \ldots, G_\mu)$; /* Where $G_i = (I_{i,1}, I_{i,2}, \ldots, I_{i,g})$ */
Initialize non-dominated archive $\mathbb{I}^*(t) \leftarrow \phi$;
Initialize reference point $Z^* \leftarrow (z_1^*, z_2^*, \ldots, z_\mathbb{k}^*)$;
for $i \leftarrow i \in (1, 2, \ldots, \mu)$ **do**
  Evaluate $\mathbb{F}(I_i(t)) \leftarrow (f_1(I_i(t)), f_2(I_i(t)), \ldots, f_\mathbb{k}(I_i(t))$;
end for
while $\iota(\mathbb{I}^\mu(t)) \neq true$) **do**
  for $i \leftarrow i \in (1, 2, \ldots, \mu)$ **do**
    $I_{i,1}(t) \leftarrow \Theta_s(G_i(t))$
    $I_{i,2}(t) \leftarrow \Theta_s(G_i(t))$
    $I_i(t) \leftarrow \Theta_r(I_{i,1}(t), I_{i,2}(t), p_c)$
    $I_i(t) \leftarrow \Theta_m(I_i(t), p_m)$
    Evaluate $F(I_i(t))$;
    Update $G_i(t)$;
    Update $\mathbb{I}^*(t)$;
    Update $\mathbb{Z}^*$;
  end for
  $t \leftarrow t + 1$;
end while
Return $\mathbb{I}^*(t)$

detection in PPI networks. We classify the approaches into two groups: complex detection based on local heuristic algorithms and complex detection based on evolutionary algorithms.

A. COMPLEX DETECTION BASED ON LOCAL HEURISTIC ALGORITHMS

These methods can be defined as methods that rely mainly on enhancing the local cost for characterizing protein complexes by their density in PPI networks. Bader et al. [5] proposed a molecular complex detection (MCODE) algorithm for finding highly linked nodes in PPI networks. To identify protein complexes, MCODE uses local graph density. It consists of three distinct phases: vertex weighting (or node scoring), complex prediction, and optimal post-processing. First, node scoring provides each node with a weight representing its local neighborhood density. Second, the algorithm iteratively expands outward from the highest-weighted seed node, encompassing the nodes of a complex whose weight is over the threshold. The value of this threshold is specified as a fraction of the seed node's total weight. Finally, MCODE identifies the densely connected regions of a molecular interaction network using only connectivity information. These regions are mapped onto identified molecular complexes.

Li et al. [29] proposed the dense-neighborhood extraction using connectivity and confidence features (DECAFF) algorithm to combine functional information and identify dense protein complexes. To mine several potentially overlapping dense subgraphs, a hub-removal technique and a local clique merging algorithm (LCMA) are applied. To ensure that proteins in predicted protein complexes are interconnected via strong confidence protein interactions, DECAFF eliminates potentially spurious protein complexes with low reliability. In this context, the reliability of a subgraph is calculated using a probabilistic model for estimating the reliability of edges within a complex.

The restricted neighborhood search clustering (RNSC) algorithm proposed by King et al. [26] is a cost-based local search algorithm that searches for promising solutions while minimizing a cost function reflecting the total number of intra-relation and inter-relations. The algorithm assigns weights to each node in the graph depending on the density of its nearest neighbors. The algorithm begins with a random solution and repeatedly shifts nodes between complexes to reduce total cost. Complexes are formed by repeatedly adding high-scoring nodes towards the complex around extremely highly weighted nodes (seed nodes). Finally, sparse complexes (complexes that cannot be satisfactorily detected) are weeded out of the final partition.

Nepusz et al. [35] proposed Clustering with Overlapping Neighborhood Expansion (ClusterOne), which seeks to find clusters with a high degree of cohesion. It begins from the seed node with the highest degree. Then, a greedy algorithm is used to add or eliminate nodes to form clusters with a high cohesiveness. Adding or removing multiple nodes may fix overlapping clusters. The algorithm calculates the overlapping score between cluster pairs and combines them when the score exceeds a predetermined threshold. ClusterOne considers the dependability of interacting proteins; however, it only predicts dense clusters and does not consider the impact of false-negative interactions.

The Core and Peel algorithm [42] peels out the vertices with the lowest degree while using a strict upper bound to regulate core decomposition for the detection of quasi-cliques. The Core and Peel approach aims to increase the density of the generated clusters. The approach quantifies the core decomposition of an initial network in which each node is a part of a fully connected subgraph in which each node has a degree of at least (k) in the initial phase. Then, the node with the largest k-core is selected as the seed. The number of nodes in the induced subgraph of a selected node and its neighbors, which are a part of the same or a larger k-core, should be greater than a predefined threshold (q), and the density should be higher than a specified value (δ). Once the cluster density is above or equal to the user-defined (δ) or the number of nodes decreases below the threshold (q), the peeling process iteratively removes nodes with a minimum degree. Duplicates and clusters entirely embedded in other clusters will be removed first before the final cluster set is obtained. The Core and Peel algorithm may

discover overlapping clusters while not considering PPIN noise.

The Dynamic Core-Attachment (DCA) [50] uses the three-sigma approach to design a dynamic PPIN that integrates the inherent organizational structures of protein complexes and uses an outward expanding strategy to identify protein complexes that have the core-attachment structure's. This approach is based on dynamic PPINs from gene expression data and PPIN topological characteristics to identify protein complexes.

The protein complex detection (PROCODE) proposed by Haque et al. [21] is based on common neighborhood and belongs to the local neighborhood density search (LD) technique. In predicting protein complexes, PROCODE considers the dense regions of a PPIN and the inherent organization of proteins within a network. After the initial identification of core proteins, the merging approach is used to append proteins according to their density value.

Omranian et al. [37] proposed a greedy approximation algorithm, known as protein complexes from coherent partition approach (PC2P). It does not require any parameter, and it frames the problem as partitioning the network into biclique spanning subgraphs, with the aim of removing the fewest possible edges. This heuristic approach provides a high search capability, but it is time-consuming and resource-intensive.

Meng et al. [32] proposed a complex detection approach using hierarchical compression network embedding and core-attachment structures. In this approach, topological information from the PPI network can be preserved both locally and globally.

An embedding method based on multi-level networks called (DPCMNE) was proposed by Meng et al. [33] to detect protein complexes. The topological information of biological networks can be preserved both locally and globally. In its first step, DPCMNE recursively compresses the input PPI network into multiple levels of smaller networks. Next, protein embeddings of different levels of granularity are learned using a network embedding method. All PPI embeddings from the compressed networks are concatenated to produce the final embeddings for the input protein network. Last but not least, based on pairwise similarity of protein embeddings, a core-attachment based strategy is adopted to detect protein complexes.

The complex prediction algorithm based on network motif (CPNM) was proposed by Patra et al. [40] to predict protein complexes. It consists of two main steps. The first step is to identify network motifs, followed by defining the role played by each protein in each motif. PPI networks quantify the role of proteins by their degree. Consequently, proteins with similar roles in different network motifs can be considered similar. The second step in the CPNM procedure uses the original PPI network, NMVector, and NMWeight as input arguments to perform neighborhood search predictions for protein complexes approach. Therefore, by selecting a seed node, CPNM iteratively adds neighboring nodes based on three constraints: (1) the attached node should be the neighbor of the nodes in the complex, (2) the Manhattan distance between the NMVectors of two adjacent nodes should be the lowest between all the adjacent nodes, and (3) the average weight of the complex should not be less than the threshold set by the node addition.

B. COMPLEX DETECTION BASED ON EVOLUTIONARY ALGORITHMS

Evolutionary algorithms (EAs) represent recent attempts to imitate nature, which use prior knowledge to explore a variant space in search of new data information. In the area of PPI, Pizzuti et al. [47] were the first to propose a single-objective EA to address the problem of complex detection in PPI networks. They relied on the suggestion of many cost functions to solve the problem. After that, Attea and Abdullah [4] proposed a multi-objective model to detect protein complexes. They also proposed protein complex attraction and a repulsion operator to enhance the performance of single and multi-objective-based EAs. At the complex level, the proposed operator seems to release inter-connections. At the protein level, on the other hand, the proposed operator works as proposed in [7] and [22] to migrate proteins over complexes such that more intra-connections are detected. c

Abdulateef et al. [1] re-designed the MOEA proposed in [4] to have another topological-based mutation operator. This operator is based on the assumption that two proteins can be split into distinct classes on the basis of their interactions. Depending on the degree of topological similarity between two proteins, the proteins are labeled as intra-delineation pairs or inter-delineation pairs. Proteins are more likely to create an intra-delineation structure if they have many similarities in their topologies, but they may also create inter-delineation pairs. The number of formed intra-delineation pairs must be considerably higher than that of inter-delineation pairs across complexes, and high number of inter-delineation pairs and few intra-delineation pairs should indicate distinct complexes in the identification of a superior complex structure. This operator was proved to leverage the detection capability of a variety of single and multi-objective EAs.

M'barek et al. [11] proposed a single-objective genetic algorithm, named label GA-PPI-Net. They suggested a similarity function to compare genes or proteins, and then they search for the optimal community by attempting to maximize the concept of community measure. Reference [6] proposed a MOEA to optimize three objectives for identifying protein complexes in human PPI networks, and also to discovering their associations with disease. Consequently, some performance metrics consistently showed better results with the predicted complexes.

III. METHOD

A. SINGLE-OBJECTIVE COMPLEX DETECTION MODELS

To solve the problem of complex detection, many researchers have relied on the topological properties of the PPI network

TABLE 1. Statistic reporting the addition of spurious interactions to Yeast-D1.

Noise	Add_{Rand}			Add_{Most}			Add_{Least}		
	m	$\|n\|_{d=1}$	d_{Avg}	m	$\|n\|_{d=1}$	d_{Avg}	m	$\|n\|_{d=1}$	d_{Avg}
0%	4687	28	9.4687	4687	28	9.4687	4687	28	9.4687
10%	5189	28	10.4828	5065	21	10.2323	4778	20	9.6525
20%	5689	28	11.4929	5443	15	10.9959	4868	10	9.8343
30%	6179	28	12.4828	5821	13	11.7595	4959	7	10.0181
40%	6684	28	13.5030	6199	5	12.5232	5049	6	10.2
50%	7147	28	14.4383	6578	5	13.2888	5140	4	10.3838

TABLE 2. Statistic reporting the deletion of true interactions from Yeast-D1 dataset.

Noise	Del_{Rand}			Del_{Most}			Del_{Least}		
	m	$\|n\|_{d=1}$	d_{Avg}	m	$\|n\|_{d=1}$	d_{Avg}	m	$\|n\|_{d=1}$	d_{Avg}
0%	4687	28	9.4687	4687	28	9.4687	4687	28	9.4687
10%	4249	53	8.5838	4309	28	8.7050	4596	32	9.2848
20%	3851	70	7.7797	3931	28	7.9414	4506	36	9.1030
30%	3480	89	7.0303	3553	29	7.1777	4415	43	8.9191
40%	3191	119	6.4464	3175	30	6.4141	4325	45	8.7373
50%	2899	168	5.8565	2796	35	5.6484	4234	48	8.5535

TABLE 3. Performance evaluation in terms of recall, precision, and F for Yeast-D1, Yeast-D2, and Human.

Algorithm	Yeast-D1			Yeast-D2			HPRD		
	Recall	Precision	F	Recall	Precision	F	Recall	Precision	F
EA_Q	0.5346	0.5371	0.5356	0.3162	0.2669	0.2893	0.2738	0.4619	0.3438
EA_{QD}	0.6034	0.5768	0.5896	0.3407	0.2810	0.3078	0.2912	0.4835	0.3711
EA_{CS}	0.6090	0.5272	0.5649	0.3453	0.2571	0.2947	0.3110	0.4218	0.3527
$MOCD_1$	0.7051	0.5515	0.6060	0.3853	0.2858	0.3223	0.3095	0.4973	0.3873
$MOCD_2$	0.8359	0.6023	0.6898	0.4453	0.3114	0.3507	0.3176	0.5101	0.3957
$MOCD_{SNN}$	**0.8423**	**0.6462**	**0.6907**	**0.4976**	**0.3706**	**0.4460**	**0.4211**	**0.5403**	**0.4703**

TABLE 4. Statistical significance (p-value) of the proposed $MOCD_{SNN}$ over the state-of-the-art algorithms using Wilcoxon Signed Rank test.

Algorithm	Yeast-D1			Yeast-D2			HPRD		
	Recall	Precision	F	Recall	Precision	F	Recall	Precision	F
EA_Q	**0.9978**	**0.7216**	**1.0**	**0.9978**	**0.9199**	**1.0**	**0.9911**	**0.9925**	**0.9931**
EA_{QD}	**0.9978**	**0.9902**	**0.9902**	**0.9946**	**0.9199**	**0.9931**	**0.9782**	**0.9921**	**0.9953**
EA_{CS}	**0.9929**	**0.8623**	**1.0**	**0.9961**	**0.9863**	**0.9990**	**0.9913**	**0.9916**	**0.9941**
$MOCD_1$	**0.9814**	**1.0**	**1.0**	**0.9891**	**1.0**	**1.0**	**0.9954**	**0.9971**	**0.9982**
$MOCD_2$	**0.9978**	**0.9345**	**1.0**	**1.0**	**0.9580**	**1.0**	**0.9751**	**0.9821**	**0.9914**

to detect complexes. These complexes are characterized by being complexes with proteins densly connected to each other, but these proteins sparsely connected with proteins that belong to other complexes. In order to calculate the modular structure of complex networks, including PPI networks, this feature is normally computed using modularity-based methods. An additional crucial step in obtaining optimal or near-optimal solutions in the context of community detection is selecting a suitable fitness function, which was initially presented by [36] using the modularity metric. It is known that modularity (mathematically denoted as Q) is one of the most important quality functions when it comes to understanding and creating community/complex structures. In complexes, modularity is a single objective function that reflects the internal structure score. Modularity is defined as:

$$Q(\mathcal{C}) = \sum_{i=1}^{K} \left[\frac{\underline{m}(C_i)}{m} - \left(\frac{\sum_{v_i \in C_i} m(v_i)}{2m} \right)^2 \right] \tag{6}$$

where $\underline{m}(C_i)$, m_i, m, and K describe, respectively, the number of intra-connections for community C_i, the number of connections for protein v_i, the total number of connections in the network, and the number of predicted complexes. In this case, Q is a metric that measures the fraction of intra-connections that fall within communities as opposed to the number that would be predicted in an equivalent network with the same number of communities but even a random distribution of edges within the communities. Consequently, Q approaches its minimum, i.e. 0, if the number of intra-connections, $\underline{m}(C_i)$, is no better than the random distribution. Alternatively, Q approaches 1 while meeting strong community structures. However, modularity encounters a resolution limit when many small communities remain undetected even when they are well defined, such as cliques. Modularity performance is highly impacted by this resolution problem in many real networks, such as PPI networks. To avoid the resolution limit of Q, another variant, called modularity density (QD in Eq. 7), is proposed [16], [30]. Basically, it is based on the average degree or the density of subgraphs. According to the size of the community, QD measures the difference between internal and external degrees.

$$QD(\mathcal{C}) = \sum_{i=1}^{K} \frac{\underline{m}(C_i) - m(C_i)}{\mid C_i \mid} \tag{7}$$

Another well-known model to define complex detection problem as a single objective is community score of [44] and [45]. It is proposed here to maximize the Community

TABLE 5. Robustness evaluation in terms of recall, precision, and F. False interactions are randomly added to protein pairs.

Noise	Algorithm	Yeast-D1 Recall	Yeast-D1 Precision	Yeast-D1 F	Yeast-D2 Recall	Yeast-D2 Precision	Yeast-D2 F
0%	EA_Q	0.5346	0.5371	0.5356	0.3162	0.2669	0.2893
	EA_{QD}	0.6034	0.5768	0.5896	0.3407	0.2810	0.3078
	EA_{CS}	0.6090	0.5272	0.5649	0.3453	0.2571	0.2947
	$MOCD_1$	0.7051	0.5515	0.6060	0.3853	0.2858	0.3223
	$MOCD_2$	0.8359	0.6023	0.6898	0.4453	0.3114	0.3507
	$MOCD_{SNN}$	**0.8423**	**0.6462**	**0.6907**	**0.4976**	**0.3706**	**0.4460**
10%	EA_Q	0.4545	0.5196	0.4844	0.3130	0.3201	0.3159
	EA_{QD}	0.4872	0.5367	0.5104	0.2491	0.2555	0.2519
	EA_{CS}	0.5462	0.4965	0.5196	0.2769	0.2333	0.2529
	$MOCD_1$	0.5407	0.5728	0.5362	0.2742	0.2828	0.2690
	$MOCD_2$	0.8231	0.6000	0.6835	0.4227	0.3079	0.3445
	$MOCD_{SNN}$	**0.8821**	**0.7077**	**0.7138**	**0.6960**	**0.4257**	**0.4857**
20%	EA_Q	0.3427	0.4534	0.3897	0.2388	0.2889	0.2607
	EA_{QD}	0.3863	0.5069	0.4378	0.1682	0.2133	0.1872
	EA_{CS}	0.4585	0.4440	0.4508	0.2053	0.1985	0.2014
	$MOCD_1$	0.4680	0.5652	0.4812	0.2362	0.2814	0.2423
	$MOCD_2$	0.8154	0.6060	0.6870	0.3927	0.2997	0.3284
	$MOCD_{SNN}$	**0.8756**	**0.7012**	**0.7043**	**0.6473**	**0.4174**	**0.4812**
30%	EA_Q	0.2713	.04054	0.3242	0.1801	0.2552	0.2102
	EA_{QD}	0.2962	0.4566	0.3585	0.1333	0.2192	0.1654
	EA_{CS}	0.3726	0.4036	0.3867	0.1724	0.1920	0.1811
	$MOCD_1$	0.4154	0.5643	0.4348	0.2012	0.2909	0.2161
	$MOCD_2$	0.8128	0.6160	0.6896	0.3867	0.3006	0.3261
	$MOCD_{SNN}$	**0.8667**	**0.6731**	**0.6915**	**0.6380**	**0.4513**	**0.4796**
40%	EA_Q	0.1890	0.3217	0.2375	0.1310	0.2134	0.1612
	EA_{QD}	0.2060	0.4014	0.2715	0.1084	0.2234	0.1453
	EA_{CS}	0.3145	0.3593	0.3350	0.1413	0.1812	0.1586
	$MOCD_1$	0.3437	0.5218	0.3588	0.1610	0.2966	0.1767
	$MOCD_2$	0.8038	0.6012	0.6731	0.3853	0.211	0.3275
	$MOCD_{SNN}$	**0.8654**	**0.6425**	**0.6942**	**0.6120**	**0.4167**	**0.4581**
50%	EA_Q	0.1387	0.2669	0.1819	0.0942	0.1804	0.1229
	EA_{QD}	0.1376	0.3338	0.1942	0.0649	0.1679	0.0932
	EA_{CS}	0.2509	0.3097	0.2767	0.1076	0.1442	0.1224
	$MOCD_1$	0.3005	0.4735	0.3067	0.1406	0.2803	0.1494
	$MOCD_2$	0.7795	0.6009	0.6654	0.3727	0.3641	0.3185
	$MOCD_{SNN}$	**0.8513**	**0.6439**	**0.6835**	**0.5667**	**0.4032**	**0.4370**

Score (CS), which can be defined as follows:

$$\text{Maximize} \quad CS(\mathcal{C}) = \sum_{i=1}^{K} \left(\frac{2\underline{m}(C_i)}{\mid C_i \mid} \right)^r \tag{8}$$

As an attempt to increase the weight of the degree of the internal node within a community, r controls the size of the communities. Accordingly, CS is calculated as the sum of local scores for each community.

B. MULTI-OBJECTIVE COMPLEX DETECTION MODELS

For MOO, Bandyopadhyay et al. [6] proposed a model that relies on topology and biological properties to detect protein complexes and formulate them as MOO problems. The first two objectives are formulated as maximization functions based on the topological properties of the PPI networks. These are node-to-cluster-contribution, and node-to-cluster closeness centrality. The third function measures the semantic similarity.

Another MOO model was proposed by Attea et al. [4], to formulate the problem with two conflicting, topological-based objectives. The first objective represents the intra-topological properties, while the second objective represents the inter-topological properties. The internal complex score summarizes the effect of each complex as carried out by different topological properties: volume (L_i), cardinality (n_i) of the PPI network, neighborhood nodes that have a significant relevance (ST_i), and shortest-path closeness centrality (SCC_i). Both L_i and ST_i were formulated with regard to the maximization function as in Eq. 9. The SCC_i parameter was formulated as a minimization function. Therefore, NC_i must be negated, by incorporating the effects of all of these parameters into a minimization function. Consequently, the formula for the intra-score can be written as:

$$\min \text{Intra}(\mathcal{C}) = \left(n^2 - \sum_{i=1}^{K} \frac{|L_i| + ST_i}{n_i} \right) + \sum_{i=1}^{K} SCC_i \tag{9}$$

Those proteins within one complex have the greatest degree of interaction with other proteins are represented by the ST_i parameter. In other words, the set of proteins $\mathbb{P} = \{p_1, p_2, \ldots, p_n\}$ within complex C_i has the degree of interactions greater than other sets of proteins, $d_{in_i}(p) > d_{out_i}(p)$ as presented in Eq. 10.

$$ST_i = \sum_{p \in C_i} \frac{d_{in_i}(p)}{d_{in_i}(p) + d_{out_i}(p)} \tag{10}$$

For complex C_i, the total number of interactions between inter-complex proteins, $d_{out_i}(p)$ to the cardinality n_i and the number of proteins in a complex C_i that have the fewest number of interactions, were combined into one score. After that, the inter-metric of a whole partial solution $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ was specified as a minimization function

TABLE 6. Robustness evaluation in terms of recall, precision, and F. False interactions are added to proteins of maximum number of interactions.

Noise	Algorithm	Yeast-D1 Recall	Yeast-D1 Precision	Yeast-D1 F	Yeast-D2 Recall	Yeast-D2 Precision	Yeast-D2 F
0%	EA_Q	0.5346	0.5371	0.5356	0.3162	0.2669	0.2893
	EA_{QD}	0.6034	0.5768	0.5896	0.3407	0.2810	0.3078
	EA_{CS}	0.6090	0.5272	0.5649	0.3453	0.2571	0.2947
	$MOCD_1$	0.7051	0.5515	0.6060	0.3853	0.2858	0.3223
	$MOCD_2$	0.8359	0.6023	0.6898	0.4453	0.3114	0.3507
	$MOCD_{SNN}$	**0.8423**	**0.6462**	**0.6907**	**0.4976**	**0.3706**	**0.4460**
10%	EA_Q	0.4966	0.5368	0.5154	0.3426	0.3259	0.3335
	EA_{QD}	0.5466	0.5745	0.5598	0.2998	0.2989	0.2990
	EA_{CS}	0.5957	0.5295	0.5604	0.3236	0.2623	0.2894
	$MOCD_1$	0.5840	0.5805	0.5652	0.3072	0.2986	0.3462
	$MOCD_2$	0.8218	0.6049	0.6867	0.4427	0.2986	0.3462
	$MOCD_{SNN}$	**0.8846**	**0.6699**	**0.7095**	**0.7027**	**0.4220**	**0.4929**
20%	EA_Q	0.3827	0.5027	0.4338	0.2341	0.2890	0.2579
	EA_{QD}	0.4256	0.5412	0.4762	0.2362	0.2903	0.2599
	EA_{CS}	0.5111	0.4876	0.4988	0.2724	0.2552	0.2632
	$MOCD_1$	0.5023	0.5891	0.5118	0.2478	0.3066	0.2579
	$MOCD_2$	0.8167	0.6093	0.6834	0.4167	0.3032	0.3398
	$MOCD_{SNN}$	**0.8705**	**0.6684**	**0.6829**	**0.6647**	**0.4392**	**0.4762**
30%	EA_Q	0.2853	0.4568	0.3504	0.1619	0.2520	0.1964
	EA_{QD}	0.3380	0.5538	0.4193	0.1660	0.2678	0.2044
	EA_{CS}	0.4278	0.4598	0.4424	0.2138	0.2317	0.2219
	$MOCD_1$	0.4238	0.5633	0.4374	0.1940	0.3013	0.2107
	$MOCD_2$	0.8077	0.5802	0.6656	0.4173	0.2899	.03372
	$MOCD_{SNN}$	**0.8641**	**0.6887**	**0.6824**	**0.6633**	**0.4160**	**0.4851**
40%	EA_Q	0.2049	0.4070	0.2718	0.1220	0.2334	0.1596
	EA_{QD}	0.2551	0.5225	0.3422	0.1249	0.2669	0.1695
	EA_{CS}	0.3786	0.4376	0.4055	0.1849	0.2267	0.2031
	$MOCD_1$	0.3562	0.5255	0.3662	0.1634	0.2944	0.1754
	$MOCD_2$	0.7949	0.5804	0.6565	0.4100	0.2987	0.3323
	$MOCD_{SNN}$	**0.8564**	**0.6456**	**0.6922**	**0.6353**	**0.4249**	**0.4822**
50%	EA_Q	0.1539	0.3624	0.2152	0.0900	0.2116	0.1255
	EA_{QD}	0.1906	0.5546	0.2823	0.0833	0.2557	0.1249
	EA_{CS}	0.3359	0.4257	0.3744	0.1398	0.2072	0.1664
	$MOCD_1$	0.3250	0.4639	0.3232	0.1542	0.2628	0.1564
	$MOCD_2$	0.8051	0.5683	0.6575	0.4020	0.2915	0.3300
	$MOCD_{SNN}$	**0.8462**	**0.6447**	**0.6855**	**0.6007**	**0.4336**	**0.4723**

as stated in Eq. 11.

$$\min \text{Inter}(\mathcal{C}) = K \times \sum_{i=1}^{K} \left(\frac{\sum_{p \in C_i} d_{out_i}(p)}{n_i} \right) + |p \in C_i|_{d_{in_i}(p) < d_{out_i}(p)} \tag{11}$$

The Pizzuti method (EA_Q, EA_QD, and EA_CS) offers advantages such as simplicity and the ability to use different quality functions for fitness. However, it has disadvantages related to the limited exploration of multiple objectives and sensitivity to the choice of fitness function. Additionally, its limitations include a dependency on the validity of quality functions and potential applicability constraints to specific network types. On the other hand, Bandyopadhyay et al method ($MOCD_1$) incorporated both biological and topological characteristics into their MOO frameworks for the purpose of identifying protein complexes and, additionally, detecting associations with diseases. In both studies, two distinct objective functions were defined to represent topological properties, while another objective function was specifically designed to address certain biological properties.

C. GENOTYPE ENCODING AND PHENOTYPE DECODING

The chromosome or the individual solution I of a population $\mathbb{I}$ is defined as a collection of n genes in the PPI network. Each is defined both locus value and allele value. Locus i identifies a protein p_i in the network, while its allele value j corresponds to the neighbor j that has an actual interaction with protein p_i in the network. Hence, each gene represents a possible interacted protein pair. This genotype encoding does not produce infeasible solutions where disconnected node neighbors, noisy interactions can occur, $A[p_i, p_j] = 0 \; | i, j \in \{1, 2, \ldots, n\}$. By this genotype encoding, chromosome representation can be expressed as:

$$I : (I_1, I_2, \ldots, I_n) = I_i|_{\forall i, 1 <= i <= n} \tag{12}$$

where I_i refers to the set of all neighbor nodes with node i in the PPI network. The decoding function, γ, of an individual I outlines the possible intra and inter structures of a group of complexes formed by the genotype of this individual. This means $\gamma(I) : \mathcal{C} = \{C\}_{i=1}^{K}$. However, K could vary from one individual to another. The formula for the representation described by Eq. 12 implicitly determines the number of detected complexes, K, being encoded in each individual I.

After deciding the representation of the chromosome, the next step is to collect the structure of a population of individual solutions. The population can be represented as: $\mathbb{I} = (I_1, I_2, \ldots, I_\mu)$. It is important to point out that in any EA-based algorithm, most of the computation time is governed by the adopted model. Here, the time complexity depends on the total number of nodes and their connections. A protein interaction network with n proteins and, for the worst case,

TABLE 7. Robustness evaluation in terms of recall, precision, and F. False interactions are added to proteins of minimum number of interactions.

Noise	Algorithm	Yeast-D1 Recall	Yeast-D1 Precision	Yeast-D1 F	Yeast-D2 Recall	Yeast-D2 Precision	Yeast-D2 F
0%	EA_Q	0.5346	0.5371	0.5356	0.3162	0.2669	0.2893
	EA_{QD}	0.6034	0.5768	0.5896	0.3407	0.2810	0.3078
	EA_{CS}	0.6090	0.5272	0.5649	0.3453	0.2571	0.2947
	$MOCD_1$	0.7051	0.5515	0.6060	0.3853	0.2858	0.3223
	$MOCD_2$	0.8359	0.6023	0.6898	0.4453	0.3114	0.3507
	$MOCD_{SNN}$	**0.8423**	**0.6462**	**0.6907**	**0.4976**	**0.3706**	**0.4460**
10%	EA_Q	0.5049	0.5371	0.5201	0.3741	0.3396	0.3555
	EA_{QD}	0.5291	0.5598	0.5438	0.3222	0.3008	0.3108
	EA_{CS}	0.5846	0.5259	0.5531	0.3380	0.2693	0.2996
	$MOCD_1$	0.5679	0.5799	0.5571	0.3107	0.2963	0.2963
	$MOCD_2$	0.8192	0.6142	0.6887	0.4207	0.3179	0.3486
	$MOCD_{SNN}$	**0.8833**	**0.6945**	**0.7282**	**0.6900**	**0.4355**	**0.5095**
20%	EA_Q	0.3513	0.7216	0.4704	0.3250	0.3292	0.3264
	EA_{QD}	0.4406	0.5390	0.4843	0.2827	0.3025	0.2917
	EA_{CS}	0.4906	0.4678	0.4785	0.3087	0.2612	0.2828
	$MOCD_1$	0.4743	0.6031	0.5007	0.2601	0.3333	0.2774
	$MOCD_2$	0.7987	0.6156	0.6881	0.4133	0.3334	0.3489
	$MOCD_{SNN}$	**0.8744**	**0.7039**	**0.7156**	**0.6900**	**0.4551**	**0.5249**
30%	EA_Q	0.3233	0.4337	0.3699	0.2609	0.3018	0.2791
	EA_{QD}	0.3479	0.4973	0.4088	0.2262	0.2757	0.2481
	EA_{CS}	0.4291	0.4376	0.4329	0.2576	0.2408	0.2484
	$MOCD_1$	0.3864	0.5949	0.4256	0.2173	0.3544	0.2456
	$MOCD_2$	0.7590	0.6004	0.6548	0.3967	0.3219	0.3468
	$MOCD_{SNN}$	**0.8397**	**0.7136**	**0.7004**	**0.6473**	**0.4412**	**0.4920**
40%	EA_Q	0.2460	0.3853	0.2995	0.1940	0.2623	0.2221
	EA_{QD}	0.2756	0.4680	0.3461	0.1531	0.2341	0.1845
	EA_{CS}	0.3816	0.4081	0.3937	0.1971	0.2059	0.2009
	$MOCD_1$	0.3330	0.5573	0.3648	0.1779	0.3387	0.2016
	$MOCD_2$	0.7590	0.5960	0.6585	0.3967	0.3357	0.3451
	$MOCD_{SNN}$	**0.8436**	**0.6985**	**0.6968**	**0.6147**	**0.4456**	**0.4941**
50%	EA_Q	0.1773	0.3241	0.2286	0.1429	0.2256	0.1740
	EA_{QD}	0.2098	0.4511	0.2859	0.1191	0.2372	0.1576
	EA_{CS}	0.3115	0.3554	0.3314	0.1884	0.2145	0.2001
	$MOCD_1$	0.2916	0.5153	0.3199	0.1554	0.3306	0.1766
	$MOCD_2$	0.7423	0.5835	0.6318	0.3700	0.3352	0.3336
	$MOCD_{SNN}$	**0.8192**	**0.6831**	**0.6845**	**0.6013**	**0.4511**	**0.5059**

with $n-1$ different connections for each protein, yields a worst time computational complexity equal to $\mathcal{O}(n \times n)$. Thus, for a population of μ chromosomes, the estimated worst time complexity is $\mathcal{O}(\mu \times n^2)$.

In Figure 2 an illustrative example of a compact yeast PPI network is presented, featuring 20 proteins and a total of 81 interactions. The depiction highlights an individual solution, showcasing its genotype and phenotype representations. The revelation of diverse phenotypes is facilitated by analyzing the allele values within the genotype solution. The decoding process transforms the genotype into three distinct complexes. Specifically, the first complex comprises 6 proteins, the second complex consists of 8 proteins, and the third complex encompasses 6 proteins.

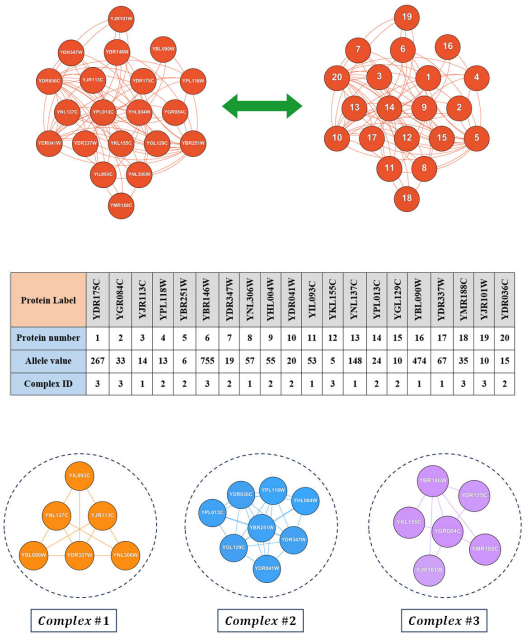


FIGURE 2. Small yeast PPI networks with 20 proteins and a total of 81 interactions are depicted, showcasing an individual solution with genotype and phenotype representations.

D. EVOLUTIONARY OPERATORS

In the adopted of EAs, iterative evolution ψ maintains two main data structures: a population of individuals $\mathbb{I}$ and a set of non-dominated solutions $\mathbb{PS}$. Formally, $\psi : \{\mathbb{I}, \mathbb{PS}\} \rightarrow \{\mathbb{I}', \mathbb{PS}'\}$ with $\psi(\mathbb{I}_t) = \mathbb{I}_{t+1}$, where $\mathbb{I}_t$ and $\mathbb{I}_{t+1}$ are the individuals at iteration t and $t+1$, respectively. However, that $\mathbb{PS}$ and $\mathbb{PS}'$ are the non-dominated solutions at iteration t and $t+1$, respectively. The population starts with an initial random set of solutions, I_0, and continues until a predetermined maximum number of iterations, denoted by the variable max_t, has been reached. A group of good-quality individuals are, then, selected and exposed to perturbation operators. A predetermined value for a chromosome-wise recombination probability, denoted by P_c, is followed to accomplish the uniform recombination. For this recombination, consider two

TABLE 8. Robustness evaluation in terms of recall, precision, and F. True interactions are randomly deleted from protein pairs.

Noise	Algorithm	Yeast-D1 Recall	Yeast-D1 Precision	Yeast-D1 F	Yeast-D2 Recall	Yeast-D2 Precision	Yeast-D2 F
0%	EA_Q	0.5346	0.5371	0.5356	0.3162	0.2669	0.2893
	EA_{QD}	0.6034	0.5768	0.5896	0.3407	0.2810	0.3078
	EA_{CS}	0.6090	0.5272	0.5649	0.3453	0.2571	0.2947
	$MOCD_1$	0.7051	0.5515	0.6060	0.3853	0.2858	0.3223
	$MOCD_2$	0.8359	0.6023	0.6898	0.4453	0.3114	0.3507
	$MOCD_{SNN}$	**0.8423**	**0.6462**	**0.6907**	**0.4976**	**0.3706**	**0.4460**
10%	EA_Q	0.5189	0.3969	0.4496	0.3181	0.2190	0.2591
	EA_{QD}	0.6150	0.5715	0.5922	0.3544	0.2785	0.3117
	EA_{CS}	0.6662	0.5572	0.6065	0.3462	0.2535	0.2925
	$MOCD_1$	0.6252	0.4682	0.5261	0.3187	0.2048	0.2463
	$MOCD_2$	0.8192	0.5823	0.6645	0.4380	0.3018	0.3394
	$MOCD_{SNN}$	**0.8923**	**0.6844**	**0.7357**	**0.7293**	**0.4237**	**0.5167**
20%	EA_Q	0.4195	0.2771	0.3336	0.2717	0.1632	0.2038
	EA_{QD}	0.5940	0.5774	0.5854	0.3404	0.2748	0.3040
	EA_{CS}	0.6222	0.5421	0.5792	0.3467	0.2543	0.2932
	$MOCD_1$	0.5579	0.3097	0.3934	0.2881	0.1429	0.1899
	$MOCD_2$	0.8038	0.5979	0.6685	0.4133	0.2818	0.3166
	$MOCD_{SNN}$	**0.8923**	**0.7022**	**0.7422**	**0.7107**	**0.4235**	**0.5112**
30%	EA_Q	0.3845	0.2308	0.2883	0.2228	0.1175	0.1537
	EA_{QD}	0.5829	0.5491	0.5652	0.3560	0.3065	0.3290
	EA_{CS}	0.6226	0.5075	0.5589	0.3622	0.2791	0.3150
	$MOCD_1$	0.4927	0.2349	0.3163	0.2601	0.1084	0.1527
	$MOCD_2$	0.7769	0.5771	0.6422	0.4453	0.3468	0.3670
	$MOCD_{SNN}$	**0.8769**	**0.6929**	**0.7377**	**0.7120**	**0.4361**	**0.5202**
40%	EA_Q	0.2974	0.1622	0.2098	0.1743	0.0842	0.1135
	EA_{QD}	0.6350	0.5893	0.6111	0.3842	0.3103	0.3432
	EA_{CS}	0.6774	0.5601	0.6130	0.3907	0.293	0.3347
	$MOCD_1$	0.3964	0.1645	0.2322	0.2247	0.0849	0.1231
	$MOCD_2$	0.7987	0.6199	0.6766	0.4473	0.3651	0.3733
	$MOCD_{SNN}$	**0.8885**	**0.7189**	**0.7687**	**0.7187**	**0.4503**	**0.5382**
50%	EA_Q	0.2042	0.0987	0.1330	0.1123	0.0508	0.0698
	EA_{QD}	0.6090	0.5348	0.5694	0.3951	0.3228	0.3551
	EA_{CS}	0.6615	0.5108	0.5763	0.4080	0.3000	0.3456
	$MOCD_1$	0.3130	0.1182	0.1715	0.1860	0.0687	0.1002
	$MOCD_2$	0.7513	0.5655	0.6307	0.4513	0.3761	0.3877
	$MOCD_{SNN}$	**0.8667**	**0.7144**	**0.7650**	**0.7287**	**0.4738**	**0.5551**

individuals, $I_1 : (I_{1,1}, I_{1,2}, \ldots, I_{1,n})$ and $I_2 : (I_{2,1}, I_{2,2}, \ldots, I_{2,n})$), as the two participating individuals. By combining the alleles of the two individuals, a child I' can be produced from them. Figure 3 illustrates the working mechanism of the uniform crossover, which is mathematically formalized as follows:

$$(\forall i \in \{1, 2, \ldots, n\}): \quad I'_i = \begin{cases} I_{1,i} & \text{if } r \leq 0.5 \\ I_{2,i} & \text{otherwise} \end{cases} \tag{13}$$

where $r \sim [0, 1]$ is a uniform random value.

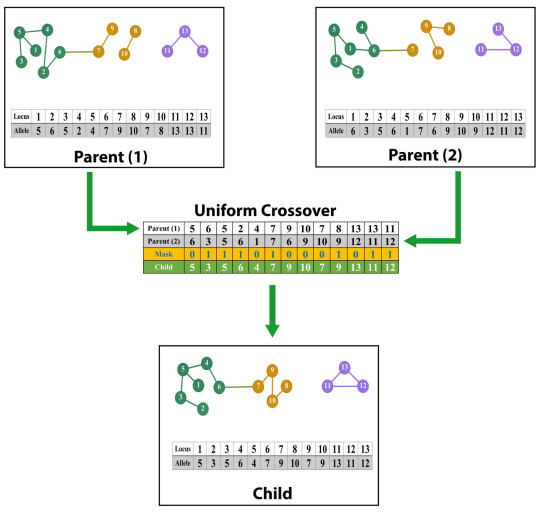


FIGURE 3. Two parents with their graph structures and genotypes. The child is generated by a uniform crossover.

E. HEURISTIC MUTATION OPERATOR

The typical purpose of the mutation operator is to make a tiny shift in the solution's behavior toward neighboring regions. Random change mutations, for example, can be used with probability P_m as a canonical non-heuristic mutation operator. However, in order to enhance the efficiency of any EA and ensure its reliability, the design must include the incorporation of domain knowledge. This concept is adopted with various formulations to improve the performance of EAs in different research topics [10], [15], [25].

1) PROTEIN COMPLEX ATTRACTION AND REPULSION

In this paper, we adopted the perturbation operator (Θ_h) proposed by Attea et al. [4] called (protein complex attraction and repulsion). This operator is an extension of the operator originally proposed in [22]. The extension is specifically designed to address the complex detection problem in PPI networks. The core idea of Θ_h is to fulfill the topological properties at both complex and protein levels. At the complex level, this operator attempts to limit the boundary of each complex with proteins that have dense linkages and to strive

TABLE 9. Robustness evaluation in terms of recall, precision, and F. True interactions are deleted from proteins of maximum number of interactions.

Noise	Algorithm	Yeast-D1 Recall	Yeast-D1 Precision	Yeast-D1 F	Yeast-D2 Recall	Yeast-D2 Precision	Yeast-D2 F
0%	EA_Q	0.5346	0.5371	0.5356	0.3162	0.2669	0.2893
	EA_{QD}	0.6034	0.5768	0.5896	0.3407	0.2810	0.3078
	EA_{CS}	0.6090	0.5272	0.5649	0.3453	0.2571	0.2947
	$MOCD_1$	0.7051	0.5515	0.6060	0.3853	0.2858	0.3223
	$MOCD_2$	0.8359	0.6023	0.6898	0.4453	0.3114	0.3507
	$MOCD_{SNN}$	**0.8423**	**0.6462**	**0.6907**	**0.4976**	**0.3706**	**0.4460**
10%	EA_Q	0.5828	0.5625	0.5721	0.4132	0.3307	0.3671
	EA_{QD}	0.6017	0.5706	0.5855	0.3467	0.2862	0.3133
	EA_{CS}	0.6517	0.5549	0.5992	0.3538	0.263	0.3015
	$MOCD_1$	0.6572	0.5751	0.6031	0.354	0.2884	0.3134
	$MOCD_2$	0.8308	0.6104	0.6848	0.4420	0.3099	0.3493
	$MOCD_{SNN}$	**0.8846**	**0.6742**	**0.7257**	**0.7327**	**0.4200**	**0.5101**
20%	EA_Q	0.6085	0.5723	0.5896	0.4198	0.3352	0.3725
	EA_{QD}	0.6325	0.5877	0.6091	0.3491	0.278	0.3093
	EA_{CS}	0.6774	0.5572	0.6112	0.3729	0.2665	0.3106
	$MOCD_1$	0.6717	0.5778	0.6111	0.3572	0.2872	0.3138
	$MOCD_2$	0.8359	0.6049	0.6802	0.4573	0.3136	0.3590
	$MOCD_{SNN}$	**0.8910**	**0.6900**	**0.7399**	**0.7200**	**0.4130**	**0.5077**
30%	EA_Q	0.6186	0.5728	0.5945	0.4247	0.3334	0.3733
	EA_{QD}	0.6423	0.5855	0.6124	0.3596	0.2814	0.3155
	EA_{CS}	0.6756	0.5538	0.6084	0.3838	0.2657	0.3139
	$MOCD_1$	0.68	0.5708	0.6106	0.3648	0.2881	0.3176
	$MOCD_2$	0.8359	0.5873	0.6716	0.4513	0.3059	0.3555
	$MOCD_{SNN}$	**0.8833**	**0.6884**	**0.7448**	**0.7240**	**0.4107**	**0.5111**
40%	EA_Q	0.6297	0.5714	0.5988	0.4282	0.3326	0.3742
	EA_{QD}	0.6479	0.5739	0.6085	0.3642	0.2788	0.3157
	EA_{CS}	0.6936	0.5511	0.614	0.3929	0.2672	0.318
	$MOCD_1$	0.685	0.5665	0.6099	0.3686	0.286	0.3179
	$MOCD_2$	0.8308	0.5840	0.6672	0.4540	0.3062	0.3512
	$MOCD_{SNN}$	**0.8910**	**0.6766**	**0.7382**	**0.7360**	**0.4031**	**0.5036**
50%	EA_Q	0.6364	0.5678	0.5999	0.4263	0.3272	0.3699
	EA_{QD}	0.6645	0.5786	0.6184	0.3691	0.278	0.317
	EA_{CS}	0.6953	0.5497	0.6137	0.3916	0.264	0.3153
	$MOCD_1$	0.6842	0.5597	0.6055	0.3677	0.2806	0.3139
	$MOCD_2$	0.8179	0.5572	0.6526	0.4580	0.3118	0.3505
	$MOCD_{SNN}$	**0.8859**	**0.6679**	**0.7355**	**0.7267**	**0.4088**	**0.5101**

toward reducing the number of scattered connections that occur between complexes. At protein level, on the other hand, the adopted operator works on re-assigning proteins to complexes such that more intra-relationships are grouped together.

2) STRONG NEIGHBOR-NODE MIGRATION OPERATOR

In this paper, we propose a heuristic perturbation operator, referred to as the strong neighbor-node migration (*SNN*) operator, to satisfy the topological attributes inherent to the protein level. The primary focus of the proposed operator is to deal with proteins being classified as weaker entities (defined by having fewer internal connections compared to external connections) within a given complex. These proteins are subsequently reassigned to one of the complexes of adjacent protein. Consider an individual $I = \{I_1, I_2, \ldots, I_N\}$ with K complexes in its complex structure $\mathcal{C}$. For the proposed *SNN* operator, each protein is examined using the parameter P_m. In the context of this framework, the preservation of the structural integrity of the complex associated with P_i is imperative, contingent upon its classification as a 'strong node'. This classification is predicated on the fulfillment of the conditions outlined in Eq. 14. In cases where this condition is not met, the complex of node P_i will assume the characteristics of one of the adjacent nodes' complexes. In a formal context, when considering a set of proteins denoted as $P_i|1 \leq i \leq n$, if protein P_i within its current complex C_k fails to meet the criteria specified in Eq. 14, the complex of the neighboring proteins P_j is assigned to protein P_i. An outline of the proposed *SNN* heuristic mutation operator is depicted in algorithm 4.

$$Intra(P_i, C) > Inter(P_i, C)|P_i \in C \tag{14}$$

Algorithm 4 General Framework of the Proposed Heuristic Mutation Operator

```
Initialize the complex structure: C ← γ(I)
Initialize protein counter: i = 1
while i ≤ N do
    Set C as the complex of protein P_i
    if rand <= p_m then
        /* The probability of SNN is satisfied */
        Calculate Intra(P_i, C) = (d_in_Pi, C)
        Calculate Inter(P_i, C) = (d_out_Pi, C)
        if Intra(P_i, C) <= Inter(P_i, C) then
            /* Protein P_i is weak in complex C */
            Find P_i = argmax_Intra(P_i, C)
            Set allele value of gene i : I_i ← i
        end if
    end if
    Increment protein counter: i = i + 1
    Return C ← γ(I);
end while
```

TABLE 10. Robustness evaluation in terms of recall, precision, and F. True interactions are deleted from proteins of minimum number of interactions.

Noise	Algorithm	Yeast-D1 Recall	Yeast-D1 Precision	Yeast-D1 F	Yeast-D2 Recall	Yeast-D2 Precision	Yeast-D2 F
0%	EA_Q	0.5346	0.5371	0.5356	0.3162	0.2669	0.2893
	EA_{QD}	0.6034	0.5768	0.5896	0.3407	0.2810	0.3078
	EA_{CS}	0.6090	0.5272	0.5649	0.3453	0.2571	0.2947
	$MOCD_1$	0.7051	0.5515	0.6060	0.3853	0.2858	0.3223
	$MOCD_2$	0.8359	0.6023	0.6898	0.4453	0.3114	0.3507
	$MOCD_{SNN}$	**0.8423**	**0.6462**	**0.6907**	**0.4976**	**0.3706**	**0.4460**
10%	EA_Q	0.4692	0.3194	0.3799	0.2721	0.1613	0.2024
	EA_{QD}	0.5799	0.5541	0.5663	0.3276	0.2865	0.3055
	EA_{CS}	0.6444	0.5307	0.5819	0.3544	0.2649	0.303
	$MOCD_1$	0.5951	0.4259	0.488	0.3053	0.1661	0.2129
	$MOCD_2$	0.8051	0.5938	0.6755	0.4220	0.3123	0.3491
	$MOCD_{SNN}$	**0.9000**	**0.6756**	**0.7272**	**0.7313**	**0.4416**	**0.5290**
20%	EA_Q	0.4581	0.2999	0.3623	0.2724	0.1537	0.1964
	EA_{QD}	0.5923	0.5404	0.565	0.3502	0.2781	0.3098
	EA_{CS}	0.6581	0.5232	0.5827	0.3622	0.2586	0.3016
	$MOCD_1$	0.5737	0.3984	0.462	0.2941	0.1615	0.2063
	$MOCD_2$	0.8372	0.5892	0.677	0.4227	0.3063	0.3425
	$MOCD_{SNN}$	**0.9115**	**0.6709**	**0.7261**	**0.7447**	**0.4357**	**0.5311**
30%	EA_Q	0.4695	0.3035	0.3685	0.2695	0.1507	0.1932
	EA_{QD}	0.5803	0.5084	0.5418	0.3544	0.2753	0.3098
	EA_{CS}	0.6385	0.4849	0.5508	0.366	0.2494	0.2965
	$MOCD_1$	0.5763	0.3729	0.4453	0.2761	0.1461	0.189
	$MOCD_2$	0.7936	0.5448	0.6310	0.4240	0.3070	0.3419
	$MOCD_{SNN}$	**0.9026**	**0.6654**	**0.7265**	**0.7560**	**0.4282**	**0.5268**
40%	EA_Q	0.4324	0.2589	0.3238	0.2597	0.1284	0.1717
	EA_{QD}	0.55	0.422	0.4773	0.3709	0.2573	0.3038
	EA_{CS}	0.5872	0.395	0.472	0.3751	0.2354	0.2892
	$MOCD_1$	0.5264	0.2998	0.3763	0.2822	0.1282	0.1752
	$MOCD_2$	0.7244	0.4379	0.5380	0.4080	0.2705	0.3121
	$MOCD_{SNN}$	**0.9077**	**0.6691**	**0.7327**	**0.7620**	**0.4032**	**0.5097**
50%	EA_Q	0.2699	0.1341	0.1791	0.1852	0.0907	0.1218
	EA_{QD}	0.4957	0.3692	0.4231	0.3362	0.2743	0.3019
	EA_{CS}	0.5316	0.3535	0.4245	0.3489	0.2577	0.2964
	$MOCD_1$	0.384	0.1544	0.2194	0.248	0.0991	0.1414
	$MOCD_2$	0.6103	0.3930	0.4706	0.3920	0.3221	0.3374
	$MOCD_{SNN}$	**0.8872**	**0.7007**	**0.7455**	**0.7293**	**0.4613**	**0.5491**

To elucidate the significant role of this operator in enhancing both exploration and exploitation facets within our approach, we provide a detailed exposition:

- Exploration: Our innovative heuristic operator serves as a catalyst for exploration by seamlessly directing solutions towards promising domains within the expansive search space. Specifically, the concept of ''strong neighbor-node migration'' empowers the algorithm to traverse diverse solution landscapes, accentuating the consideration of robust interactions between nodes. This pivotal attribute enriches the exploration phase, leading to a more nuanced and comprehensive examination of potential solutions tailored to the intricate detection challenges inherent in Protein-Protein Interaction (PPI) networks.
- Exploitation: In addition to fostering exploration, our heuristic operator adeptly facilitates exploitation by steering the evolutionary trajectory towards regions in the search space where high-quality solutions are likely to be concentrated. Through a deliberate focus on potent neighbor-node interactions, the algorithm adeptly exploits areas that hold promise, enabling iterative refinement and enhancement of solutions. This strategic approach ensures a continual convergence of the algorithm towards optimal or near-optimal solutions, solidifying its efficacy over time.

In this paper, we introduce a methodical and detailed framework for evaluating the detection of protein complexes, as illustrated in Figure 4. This framework is structured into multiple critical stages, each playing a vital role in the precise detection and assessment of protein complexes. By systematically addressing each stage, the methodology ensures a comprehensive approach to the analysis of protein complexes.

IV. RESULTS AND DISCUSSION

In this paper, we propose a heuristic perturbation operator for the complex detection problem in PPI networks. Thus, we must determine whether these suggestions make sense by evaluating the quality gain of the generated complexes as compared to the most recent models. In this section, the performance evaluations are presented into two successive competition phases. In the first phase, we will study the effectiveness of the proposed heuristic operator injected with multi-objective model against the single and multi-objective EA models proposed by the state-of-the-art. In the second phase, we will evaluate the robustness of the proposed heuristic operator and the adopted single and multi-objective EAs in unraveling protein complexes from PPI networks that contain noisy interactions or suffer from missing true interactions. Three models with single objective EAs proposed by Pizzuti et al. [47] are evaluated. These models

TABLE 11. Robustness evaluation in terms of $recall_N$, $precision_N$, and F_N. False interactions are randomly added to protein pairs.

Noise	Algorithm	Yeast-D1			Yeast-D2		
		$Recall_N$	$Precision_N$	F_N	$Recall_N$	$Precision_N$	F_N
0%	EA_Q	0.446	0.3099	0.3657	0.1813	0.1728	0.1766
	EA_{QD}	0.535	0.3745	0.4406	0.2008	0.187	0.1934
	EA_{CS}	0.5619	0.394	0.4632	0.2043	0.1976	0.2006
	$MOCD_1$	0.6227	0.4263	0.5057	0.2296	0.2044	0.2137
	$MOCD_2$	0.8409	0.5648	0.6738	0.2661	0.2418	0.2509
	$MOCD_{SNN}$	**0.8561**	**0.5834**	**0.6914**	**0.2835**	**0.3015**	**0.2860**
10%	EA_Q	0.3826	0.3826	0.3826	0.1909	0.3809	0.2537
	EA_{QD}	0.4233	0.2936	0.3467	0.264	0.1445	0.1463
	EA_{CS}	0.5031	0.3436	0.4083	0.1704	0.1606	0.165
	$MOCD_1$	0.507	0.3044	0.3794	0.1695	0.1419	0.1538
	$MOCD_2$	0.8238	0.5578	0.6639	0.2515	0.2322	0.2399
	$MOCD_{SNN}$	**0.8315**	**0.5597**	**0.6811**	**0.2732**	**0.2977**	**0.2858**
20%	EA_Q	0.285	0.285	0.285	0.147	0.2977	0.196
	EA_{QD}	0.3362	0.2247	0.2694	0.2412	0.0987	0.0986
	EA_{CS}	0.4161	0.2777	0.3331	0.1331	0.1284	0.1301
	$MOCD_1$	0.4449	0.2626	0.329	0.1481	0.1227	0.1334
	$MOCD_2$	0.8257	0.5483	0.6569	0.24	0.219	0.226
	$MOCD_{SNN}$	**0.8319**	**0.5498**	**0.6613**	**0.2715**	**0.2515**	**0.2731**
30%	EA_Q	0.2156	0.2156	0.2156	0.1027	0.2143	0.138
	EA_{QD}	0.2425	0.1584	0.1916	0.2208	0.0703	0.0685
	EA_{CS}	0.335	0.2172	0.2635	0.1022	0.0967	0.0989
	$MOCD_1$	0.3926	0.2308	0.2895	0.1263	0.103	0.1126
	$MOCD_2$	0.819	0.5442	0.6529	0.2395	0.2162	0.2265
	$MOCD_{SNN}$	**0.8288**	**0.5491**	**0.6534**	**0.2476**	**0.2381**	**0.2301**
40%	EA_Q	0.1429	0.1429	0.1429	0.0728	0.1521	0.0974
	EA_{QD}	0.1635	0.104	0.1271	0.1938	0.0509	0.0507
	EA_{CS}	0.279	0.1801	0.2189	0.0759	0.0789	0.0771
	$MOCD_1$	0.3241	0.1915	0.2396	0.1034	0.0839	0.0917
	$MOCD_2$	0.8084	0.5399	0.6467	0.2394	0.2126	0.2244
	$MOCD_{SNN}$	**0.8195**	**0.5468**	**0.6510**	**0.2411**	**0.2372**	**0.2301**
50%	EA_Q	0.0972	0.0972	0.0972	0.0473	0.1052	0.0646
	EA_{QD}	0.1126	0.0698	0.0862	0.1691	0.0347	0.0338
	EA_{CS}	0.236	0.1492	0.1828	0.0677	0.0633	0.0647
	$MOCD_1$	0.2823	0.1691	0.2106	0.09	0.0741	0.0804
	$MOCD_2$	0.7632	0.5032	0.6051	0.2352	0.2034	0.2175
	$MOCD_{SNN}$	**0.7831**	**0.5185**	**0.6137**	**0.2489**	**0.2322**	**0.2198**

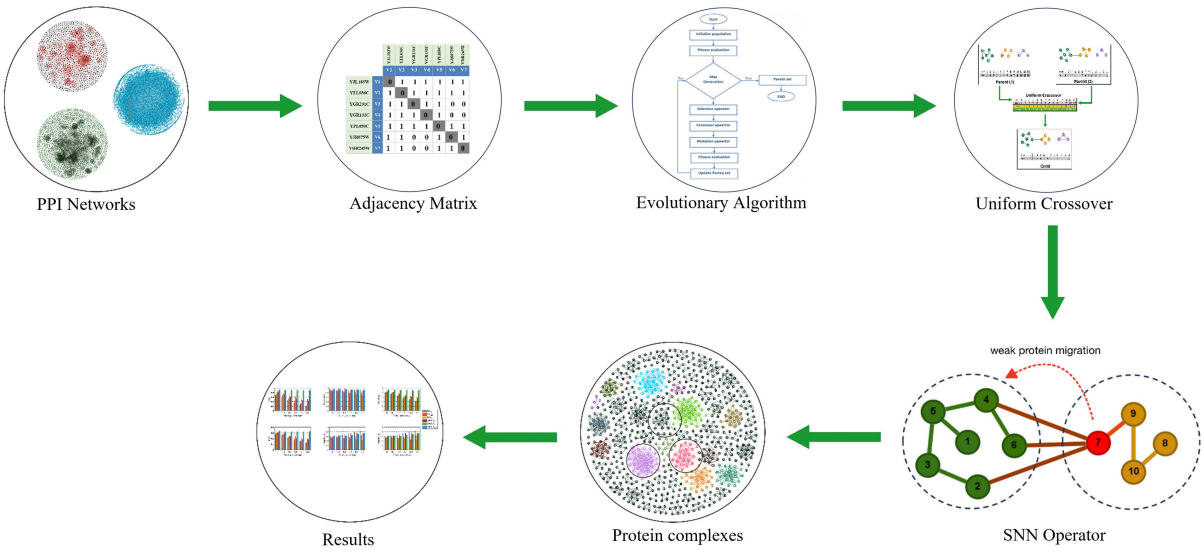


FIGURE 4. A Comprehensive Framework for Evaluating Protein Complex Detection: An Overview of Key Stages and Processes.

are called hereinafter as EA_Q, EA_{QD}, and EA_{CS}. Further, the multi-objective EA-based model proposed by [6] and formulated in Eq. 15 and Eq. 16 is also adopted using the framework of MOEA/D (referred as $MOCD_1$). The second multi-objective EA-based model (formulated in Eq. 9 and Eq. 11) and the heuristic operator proposed by [4] are adopted using the framework of NSGA-II (referred as $MOCD_2$). The third multi-objective EA-based model based on our heuristic operator is used the framework of MOEA/D (referred as $MOCD_{SNN}$).

$$\max Con(\mathcal{C}) = \sum_{i=1}^{K} \sum_{v \in C_i} \frac{in_i(v)}{|l_v|} \tag{15}$$

$$\max CC(\mathcal{C}) = \sum_{i=1}^{K} CC_i \tag{16}$$

TABLE 12. Robustness evaluation in terms of $recall_N$, $precision_N$, and F_N. False interactions are added to proteins of maximum number of interactions.

Noise	Algorithm	Yeast-D1			Yeast-D2		
		$Recall_N$	$Precision_N$	F_N	$Recall_N$	$Precision_N$	F_N
0%	EA_Q	0.446	0.3099	0.3657	0.1813	0.1728	0.1766
	EA_{QD}	0.535	0.3745	0.4406	0.2008	0.187	0.1934
	EA_{CS}	0.5619	0.394	0.4632	0.2043	0.1976	0.2006
	$MOCD_1$	0.6227	0.4263	0.5057	0.2296	0.2044	0.2137
	$MOCD_2$	0.8409	0.5648	0.6738	0.2661	0.2418	0.2509
	$MOCD_{SNN}$	**0.8561**	**0.5834**	**0.6914**	**0.2835**	**0.3015**	**0.2860**
10%	EA_Q	0.4029	0.4029	0.4029	0.1931	0.3866	0.2569
	EA_{QD}	0.468	0.3266	0.3847	0.2644	0.1625	0.1632
	EA_{CS}	0.525	0.3577	0.4254	0.186	0.1728	0.1789
	$MOCD_1$	0.5376	0.3272	0.4059	0.1825	0.1539	0.1664
	$MOCD_2$	0.8304	0.5602	0.6674	0.2581	0.2352	0.2444
	$MOCD_{SNN}$	**0.8415**	**0.5840**	**0.6687**	**0.2776**	**0.2455**	**0.2605**
20%	EA_Q	0.2858	0.2858	0.2858	0.112	0.2351	0.151
	EA_{QD}	0.3474	0.2322	0.2783	0.2311	0.1087	0.1086
	EA_{CS}	0.4358	0.2924	0.35	0.1441	0.1398	0.1416
	$MOCD_1$	0.4535	0.2678	0.3356	0.141	0.1186	0.1283
	$MOCD_2$	0.8147	0.5413	0.6495	0.2468	0.2177	0.23
	$MOCD_{SNN}$	**0.8152**	**0.5526**	**0.6637**	**0.2497**	**0.2368**	**0.2395**
30%	EA_Q	0.1933	0.1933	0.1933	0.0677	0.1508	0.0929
	EA_{QD}	0.2523	0.1616	0.197	0.1876	0.0694	0.0706
	EA_{CS}	0.3593	0.234	0.2834	0.1026	0.1045	0.1034
	$MOCD_1$	0.3753	0.2221	0.2779	0.1069	0.0907	0.0975
	$MOCD_2$	0.8033	0.5389	0.6443	0.2403	0.2132	0.2254
	$MOCD_{SNN}$	**0.8100**	**0.5401**	**0.6538**	**0.2492**	**0.2467**	**0.2340**
40%	EA_Q	0.1305	0.1305	0.1305	0.0469	0.1056	0.0646
	EA_{QD}	0.1756	0.1097	0.135	0.1558	0.0454	0.0471
	EA_{CS}	0.303	0.1944	0.2369	0.0842	0.0815	0.0826
	$MOCD_1$	0.3071	0.1826	0.2281	0.0859	0.075	0.0796
	$MOCD_2$	0.7474	0.5024	0.5999	0.2394	0.2112	0.2238
	$MOCD_{SNN}$	**0.802**	**0.5255**	**0.6191**	**0.2404**	**0.2322**	**0.2335**
50%	EA_Q	0.0932	0.0932	0.0932	0.032	0.0734	0.0443
	EA_{QD}	0.1215	0.0738	0.0918	0.1287	0.0317	0.0322
	EA_{CS}	0.2589	0.1643	0.201	0.0601	0.0572	0.0583
	$MOCD_1$	0.2744	0.1659	0.206	0.0782	0.0696	0.0733
	$MOCD_2$	0.7522	0.5115	0.6081	0.2334	0.2031	0.2165
	$MOCD_{SNN}$	**0.7812**	**0.5221**	**0.6237**	**0.2451**	**0.2257**	**0.2232**

where CC_i is the reciprocal shortest-path distance averaged over all vertices in cluster C_i.

A. PPI NETWORK DATASETS

To evaluate the performance of the EA models, it is necessary to rely on datasets that were adopted by the previous researchers in their study. There are three different datasets were applied in this study. Two of them are *yeast Saccharomyces cerevisiae*, and the third dataset is the human proteins. Gavin et al. [19] prepared the first dataset called Yeast-D1 and [53] filtered it. There are $n = 990$ proteins and $m = 4687$ interactions in the filtered version. The second dataset, Yeast-D2 [53], contains $n = 1443$ proteins and $m = 6993$ interactions. The third dataset, human protein dataset contains $n = 9589$ protein and $m = 39240$ interactions. Figure 5 depicts these three PPI networks.

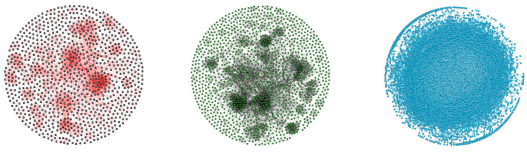


FIGURE 5. Three different PPI networks. Yeast Saccharomyces cerevisiae and human datasets are shown from left to right.

Three benchmark datasets for protein complexes (Cmplx-D1, Cmplx-D2, and human) are used in the evaluation. These are hand-curated from the Munich Information Center for Protein Sequence (MIPS) catalog [34], and human from the Human Protein Reference Database (HPRD) [43]. Cmplx-D1 contains 81 complexes, each complex ranging in size from 6 to 38 proteins. Cmplx-D2 contains 162 complexes, each complex ranging in size from 4 to 266 proteins. HPRD complex contain 1318 complexes, each complex ranging in size from 1 to 31 proteins.

Figure (6) depicts Yeast-D1 containing 990 different proteins with 4687 interactions (top left). This network is decomposed into 81 complexes of different sizes (top right). One of these complexes is selected and enlarged (bottom right). The selected complex contains 21 proteins with their internal connections. Further, protein #49 ('YBR198C') from this complex is also depicted (bottom left) to clarify its details within the complex. It contains internal links (represented in green color) and three other inter-connections (represented in red color).

B. EVALUATION MEASURES

1) COMPLEX-LEVEL EVALUATION

There are a variety of metrics that can be used to evaluate the efficacy of the predicted complexes. For the complex level, the predicted complexes $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ obtained by the examined model were compared to benchmark gold complexes $\mathcal{C}^* = \{C_1^*, C_2^*, \ldots, C_{K^*}^*\}$ from the MIPS. In terms

TABLE 13. Robustness evaluation in terms of $recall_N$, $precision_N$, and F_N. False interactions are added to proteins of minimum number of interactions.

Noise	Algorithm	Yeast-D1			Yeast-D2		
		$Recall_N$	$Precision_N$	F_N	$Recall_N$	$Precision_N$	F_N
0%	EA_Q	0.446	0.3099	0.3657	0.1813	0.1728	0.1766
	EA_{QD}	0.535	0.3745	0.4406	0.2008	0.187	0.1934
	EA_{CS}	0.5619	0.394	0.4632	0.2043	0.1976	0.2006
	$MOCD_1$	0.6227	0.4263	0.5057	0.2296	0.2044	0.2137
	$MOCD_2$	0.8409	0.5648	0.6738	0.2661	0.2418	0.2509
	$MOCD_{SNN}$	**0.8561**	**0.5834**	**0.6914**	**0.2835**	**0.3015**	**0.2860**
10%	EA_Q	0.4202	0.4202	0.4202	0.2302	0.4545	0.3051
	EA_{QD}	0.4667	0.3272	0.3847	0.2836	0.1782	0.1833
	EA_{CS}	0.5484	0.3727	0.4438	0.2021	0.1918	0.1967
	$MOCD_1$	0.5304	0.324	0.4013	0.1936	0.1632	0.1765
	$MOCD_2$	0.8282	0.5469	0.6583	0.2529	0.2207	0.235
	$MOCD_{SNN}$	**0.8451**	**0.5531**	**0.6735**	**0.2601**	**0.2814**	**0.2509**
20%	EA_Q	0.3373	0.3373	0.3373	0.1963	0.3919	0.2609
	EA_{QD}	0.3826	0.2609	0.3103	0.2701	0.1525	0.158
	EA_{CS}	0.4682	0.3133	0.3753	0.1865	0.1718	0.1786
	$MOCD_1$	0.45	0.2627	0.3306	0.1653	0.13	0.1446
	$MOCD_2$	0.802	0.5291	0.6371	0.2526	0.2124	0.2296
	$MOCD_{SNN}$	**0.8310**	**0.5311**	**0.6422**	**0.2608**	**0.2399**	**0.2366**
30%	EA_Q	0.2722	0.2722	0.2722	0.1542	0.3134	0.2059
	EA_{QD}	0.3118	0.206	0.2481	0.2357	0.1109	0.1154
	EA_{CS}	0.4207	0.2747	0.3323	0.1562	0.1433	0.1489
	$MOCD_1$	0.3772	0.2163	0.2738	0.1415	0.108	0.1215
	$MOCD_2$	0.7532	0.4914	0.5941	0.2423	0.2037	0.2201
	$MOCD_{SNN}$	**0.7704**	**0.5003**	**0.6306**	**0.2503**	**0.2085**	**0.2333**
40%	EA_Q	0.1988	0.1988	0.1988	0.1109	0.2298	0.1486
	EA_{QD}	0.2283	0.148	0.1796	0.2115	0.0807	0.082
	EA_{CS}	0.3716	0.2405	0.292	0.1235	0.1148	0.1183
	$MOCD_1$	0.3278	0.1894	0.239	0.1188	0.0914	0.1021
	$MOCD_2$	0.7558	0.4977	0.599	0.2437	0.2037	0.2204
	$MOCD_{SNN}$	**0.7650**	**0.5010**	**0.6227**	**0.2594**	**0.2155**	**0.2314**
50%	EA_Q	0.1317	0.1317	0.1317	0.0751	0.1573	0.1008
	EA_{QD}	0.1709	0.1066	0.1313	0.1896	0.0587	0.0596
	EA_{CS}	0.3107	0.201	0.2441	0.1206	0.1057	0.112
	$MOCD_1$	0.2848	0.1657	0.2088	0.1033	0.0796	0.0889
	$MOCD_2$	0.7274	0.4811	0.5788	0.2321	0.1956	0.211
	$MOCD_{SNN}$	**0.7412**	**0.5028**	**0.5894**	**0.2435**	**0.2143**	**0.2300**

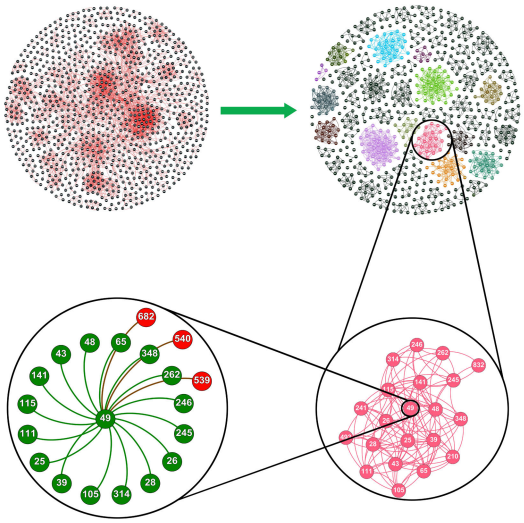


FIGURE 6. Yeast-D1 and protein #49 with its intra-connections (green edges) and inter-connections (red edges).

of proteins, an overlapping score (OS) indicates the degree to which a predicted complex C_i in the solution $\mathcal{C}$ overlaps with a benchmark complex C_j^* (Eq. 17). If both complexes (C_i and C_j^*) have an overlapping score (OS) equal to or greater than a given threshold, σOS, then the predicted complex C_i is said to match the benchmark complex C_j^*.

$$OS(C_i, C_j^*) = \frac{|C_i \cap C_j^*|^2}{|C_i \cup C_j^*|} \tag{17}$$

$$match(C_i, C_j^*) = \begin{cases} 1, & \text{if } OS(C_i, C_j^*) \geq \sigma OS \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

The terms of recall, precision, and cumulative F are determined according to the matching (stated in Eq. 18). The quality of a prediction is measured by its recall, which is the fraction of benchmark complexes that are successfully matched to any predicted complex. On the other hand, precision refers to the fraction of predicted complexes that are identical to a given benchmark complex. The F is a harmonic mean of both recall and precision, as demonstrated in Eq. 20.

$$Recall = \frac{|C_i^* : C_i^* \in \mathcal{C}^* \wedge \exists C_j \in \mathcal{C} \rightarrow match(C_i^*, C_j)|}{K^*} \tag{19}$$

$$Precision = \frac{|C_i : C_i \in \mathcal{C} \wedge \exists C_j^* \in \mathcal{C}^* \rightarrow match(C_i, C_j^*)|}{K}$$

$$F = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{20}$$

2) PROTEIN-LEVEL EVALUATION

In the same manner as recall and precision measure the overall quality of the prediction at the complex level,

TABLE 14. Robustness evaluation in terms of $recall_N$, $precision_N$, and F_N. True interactions are randomly deleted from protein pairs.

Noise	Algorithm	Yeast-D1			Yeast-D2		
		$Recall_N$	$Precision_N$	F_N	$Recall_N$	$Precision_N$	F_N
0%	EA_Q	0.446	0.3099	0.3657	0.1813	0.1728	0.1766
	EA_{QD}	0.535	0.3745	0.4406	0.2008	0.187	0.1934
	EA_{CS}	0.5619	0.394	0.4632	0.2043	0.1976	0.2006
	$MOCD_1$	0.6227	0.4263	0.5057	0.2296	0.2044	0.2137
	$MOCD_2$	0.8409	0.5648	0.6738	0.2661	0.2418	0.2509
	$MOCD_{SNN}$	**0.8561**	**0.5834**	**0.6914**	**0.2835**	**0.3015**	**0.2860**
10%	EA_Q	0.3873	0.3882	0.3877	0.1514	0.318	0.2044
	EA_{QD}	0.5356	0.3772	0.4426	0.2939	0.1911	0.2004
	EA_{CS}	0.6284	0.4398	0.5174	0.2067	0.1987	0.2024
	$MOCD_1$	0.5306	0.3195	0.398	0.1662	0.1469	0.1553
	$MOCD_2$	0.8218	0.561	0.6659	0.2612	0.2366	0.2479
	$MOCD_{SNN}$	**0.8298**	**0.5729**	**0.6837**	**0.2731**	**0.2408**	**0.2513**
20%	EA_Q	0.2717	0.276	0.2738	0.1046	0.2438	0.1462
	EA_{QD}	0.5316	0.388	0.4486	0.2935	0.1911	0.197
	EA_{CS}	0.5837	0.4142	0.4845	0.2053	0.2015	0.2032
	$MOCD_1$	0.3951	0.2573	0.3109	0.1194	0.116	0.1173
	$MOCD_2$	0.8183	0.5754	0.675	0.2464	0.2291	0.2365
	$MOCD_{SNN}$	**0.8206**	**0.5776**	**0.6800**	**0.2507**	**0.2333**	**0.2406**
30%	EA_Q	0.2368	0.2456	0.2411	0.0736	0.172	0.103
	EA_{QD}	0.5243	0.3916	0.4483	0.2979	0.2034	0.2073
	EA_{CS}	0.6029	0.4512	0.5161	0.2103	0.2089	0.2094
	$MOCD_1$	0.3073	0.2126	0.251	0.0873	0.0931	0.0899
	$MOCD_2$	0.7706	0.5517	0.6415	0.2605	0.2477	0.2533
	$MOCD_{SNN}$	**0.7811**	**0.5684**	**0.6499**	**0.2695**	**0.2536**	**0.2645**
40%	EA_Q	0.1676	0.1766	0.172	0.0523	0.1281	0.0742
	EA_{QD}	0.5553	0.4356	0.4882	0.305	0.2167	0.2159
	EA_{CS}	0.6333	0.494	0.5551	0.2141	0.2284	0.2209
	$MOCD_1$	0.2105	0.1577	0.1802	0.0629	0.073	0.0675
	$MOCD_2$	0.7706	0.5736	0.6573	0.2547	0.2553	0.2526
	$MOCD_{SNN}$	**0.7790**	**0.5844**	**0.6791**	**0.2602**	**0.2617**	**0.2670**
50%	EA_Q	0.1149	0.1276	0.1209	0.0296	0.0797	0.0431
	EA_{QD}	0.5384	0.4521	0.4915	0.3075	0.2353	0.2276
	EA_{CS}	0.6153	0.519	0.563	0.218	0.2416	0.2291
	$MOCD_1$	0.1487	0.121	0.1334	0.0455	0.0592	0.0514
	$MOCD_2$	0.6976	0.5534	0.6159	0.2492	0.2616	0.2516
	$MOCD_{SNN}$	**0.7334**	**0.5666**	**0.6234**	**0.2504**	**0.2600**	**0.2594**

$Recall_N$ and $Precision_N$ can evaluate the quality of the prediction, but at the protein level [53]. Based on these measurements, F_N resembles at the protein level.

$$Recall_N = \frac{\sum_{i=1}^{K^*} |\max_{C_j \in \mathcal{C}} match(C_i^*, C_j)|}{\sum_{i=1}^{K^*} |C_i^*|} \tag{21}$$

$$Precision_N = \frac{\sum_{i=1}^{K} |\max_{C_j^* \in \mathcal{C}^*} match(C_i, C_j^*)|}{\sum_{i=1}^{K} |C_i|} \tag{22}$$

$$F_N = \frac{2 \times Recall_N \times Precision_N}{Recall_N + Precision_N} \tag{23}$$

Two additional measures based on the intersection between true complexes $\mathcal{C}$ and detected complexes $\mathcal{C}^*$ can be evaluated directly without reference to the overlapping score (σOS). These measures are *sensitivity* and positive predictive value *PPV*.

$$sensitivity = \frac{\sum_{i=1}^{K} \max_{j=1}^{K^*} T_{i,j}}{\sum_{i=1}^{K^*} n_i} \tag{24}$$

$$PPV = \frac{\sum_{j=1}^{K} \max_{i=1}^{K^*} T_{i,j}}{\sum_{j=1}^{K} \sum_{i=1}^{K^*} T_{i,j}} \tag{25}$$

where $T_{i,j}$ in both Eq. 24 and Eq. 25 represents the matched proteins between the true complexes and the predicted complexes. Geometric accuracy represents the trade-off between *sensitivity* and *PPV* in Eq. 26.

$$accuracy = \sqrt{sensitivity \times PPV} \tag{26}$$

C. ALGORITHM PARAMETER SETTINGS

All single and multi-objective EA models used in this paper are set up with the following parameter settings: population size (μ) is set to 100, and to stop the evolution process, the maximum number of generations is set to 100. We set the following control parameters: probability of uniform crossover, ($P_c = 0.8$), probability of mutation operator, ($P_m = 0.2$), and maximum number of run (MaxRun = 30). Extensive experiments on all EA models were evaluated over yeast PPI networks and their noisy versions. Noisy PPI networks were generated using different percentage of noise. In each percentage, 10 different artificial PPI networks were generated. Each network is tested with 30 different runs and the average of the 10 networks over 300 runs is reported in each test case.

D. ROBUSTNESS AGAINST NEGATIVE CONTROL

The reliability of a PPI network is affected by a major problem: the high noise rate in high-throughput experiments. However, the spurious inter-complex interactions refer to the addition of false positives, whereas removal of interactions refer to missing protein interactions in a PPI network. Brohee et al., and Pizzuti et al., [14], [47] have investigated the

TABLE 15. Robustness evaluation in terms of $recall_N$, $precision_N$, and F_N. True interactions are deleted from proteins of maximum number of interactions.

Noise	Algorithm	Yeast-D1			Yeast-D2		
		$Recall_N$	$Precision_N$	F_N	$Recall_N$	$Precision_N$	F_N
0%	EA_Q	0.446	0.3099	0.3657	0.1813	0.1728	0.1766
	EA_{QD}	0.535	0.3745	0.4406	0.2008	0.187	0.1934
	EA_{CS}	0.5619	0.394	0.4632	0.2043	0.1976	0.2006
	$MOCD_1$	0.6227	0.4263	0.5057	0.2296	0.2044	0.2137
	$MOCD_2$	0.8409	0.5648	0.6738	0.2661	0.2418	0.2509
	$MOCD_{SNN}$	**0.8561**	**0.5834**	**0.6914**	**0.2835**	**0.3015**	**0.2860**
10%	EA_Q	0.5001	0.5001	0.5001	0.2559	0.4899	0.3358
	EA_{QD}	0.5331	0.3685	0.4358	0.2952	0.1918	0.1988
	EA_{CS}	0.618	0.4174	0.4983	0.211	0.2	0.2052
	$MOCD_1$	0.6134	0.3796	0.4683	0.2145	0.1842	0.1977
	$MOCD_2$	0.8405	0.5587	0.6699	0.2588	0.2354	0.2451
	$MOCD_{SNN}$	**0.8661**	**0.5731**	**0.6801**	**0.2773**	**0.2461**	**0.2575**
20%	EA_Q	0.5323	0.5323	0.5323	0.2554	0.4959	0.3368
	EA_{QD}	0.5756	0.3937	0.4676	0.2943	0.1875	0.1978
	EA_{CS}	0.629	0.4277	0.5091	0.2165	0.2035	0.2096
	$MOCD_1$	0.6253	0.3852	0.476	0.2122	0.1856	0.1976
	$MOCD_2$	0.8294	0.5527	0.6611	0.2616	0.2435	0.2508
	$MOCD_{SNN}$	**0.8345**	**0.5635**	**0.6784**	**0.2887**	**0.2613**	**0.2602**
30%	EA_Q	0.5437	0.5437	0.5437	0.2544	0.4982	0.3364
	EA_{QD}	0.5822	0.4004	0.4745	0.296	0.1894	0.1977
	EA_{CS}	0.6262	0.4261	0.5071	0.22	0.2083	0.2139
	$MOCD_1$	0.6325	0.3908	0.4824	0.2154	0.1868	0.1995
	$MOCD_2$	0.823	0.5602	0.6666	0.2574	0.244	0.2478
	$MOCD_{SNN}$	**0.8332**	**0.5618**	**0.6713**	**0.2598**	**0.2513**	**0.2482**
40%	EA_Q	0.553	0.553	0.553	0.2502	0.4959	0.3322
	EA_{QD}	0.5788	0.4007	0.4735	0.2973	0.1932	0.2001
	EA_{CS}	0.6172	0.4236	0.5024	0.2204	0.2096	0.2148
	$MOCD_1$	0.6291	0.3869	0.4784	0.2152	0.1855	0.1988
	$MOCD_2$	0.8024	0.5277	0.6344	0.2594	0.2472	0.2512
	$MOCD_{SNN}$	**0.8227**	**0.5311**	**0.6507**	**0.2617**	**0.2500**	**0.2577**
50%	EA_Q	0.5636	0.5636	0.5636	0.2448	0.489	0.3258
	EA_{QD}	0.5823	0.4056	0.4781	0.2992	0.193	0.1987
	EA_{CS}	0.6242	0.4283	0.508	0.2155	0.2057	0.2104
	$MOCD_1$	0.6235	0.3823	0.4732	0.2128	0.1855	0.1978
	$MOCD_2$	0.7793	0.5108	0.613	0.2568	0.2423	0.2479
	$MOCD_{SNN}$	**0.7882**	**0.5266**	**0.6360**	**0.2579**	**0.2517**	**0.2561**

potential of their algorithms to assess the consistency and effectiveness of the algorithms in detecting protein complexes even in the presence of noise in PPI networks. However, the addition or deletion were performed randomly. In this paper, the adopted EAs, (EA_Q, EA_{QD}, EA_{CS}, $MOCD_1$, and $MOCD_2$) are tested on PPI networks, artificially generated as:

- Adding interactions to random protein pairs
- Adding interactions to proteins that have the most number of interactions (highest interaction score)
- Adding interactions to proteins that have the fewest number of interactions (lowest interaction score)
- Removing interactions from random protein pairs
- Removing interactions from proteins that have the most number of interactions
- Removing interactions from proteins that have the fewest number of interactions

In each type of noise, an increasing percentage of interactions (10%, 20%, 30%, 40% and 50%) are added to or deleted from Yeast-D1 and Yeast-D2. In each percentage, 10 different networks are generated. Interactions are added/deleted to/from randomly selected proteins, to/from proteins with the highest number of interactions, or to/from proteins with fewest number of interactions. Table 1 reports the statistics of adding spurious interactions. Table 2 reports the statistics of deleting true interactions from the original PPI dataset. In both tables, m refers to the number of interactions, ($|n|_{d=1}$) refers to the number of proteins that have only one interaction, and (d_{Avg}) refers to the average number of interactions per protein.

Figure 7 depicts the Saccharomyces cerevisiae yeast PPI network, which is represented by the following, the original Yeast-D1 PPI network shown at the top, which contains $n = 990$ proteins with $m = 4687$ interactions, as the proteins that contain one interaction is $|n|_{d=1} = 28$ proteins, with an average of this network is $d_{avg} = 9.4687$ interactions for each protein. In the bottom left, it represents the random addition

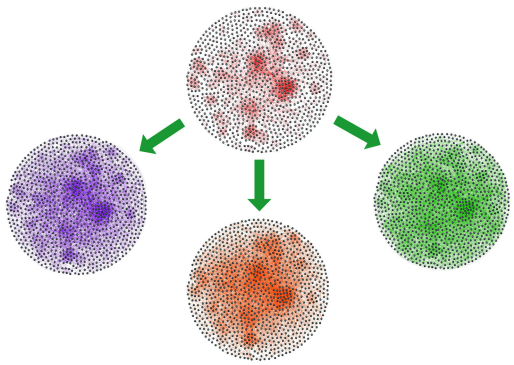


FIGURE 7. Yeast-D1 (top) and three artificial PPI networks generated by adding false interactions to Yeast-D1.

TABLE 16. Robustness evaluation in terms of $recall_N$, $precision_N$, and F_N. True interactions are deleted from proteins of minimum number of interactions.

Noise	Algorithm	Yeast-D1			Yeast-D2		
		$Recall_N$	$Precision_N$	F_N	$Recall_N$	$Precision_N$	F_N
0%	EA_Q	0.446	0.3099	0.3657	0.1813	0.1728	0.1766
	EA_{QD}	0.535	0.3745	0.4406	0.2008	0.187	0.1934
	EA_{CS}	0.5619	0.394	0.4632	0.2043	0.1976	0.2006
	$MOCD_1$	0.6227	0.4263	0.5057	0.2296	0.2044	0.2137
	$MOCD_2$	0.8409	0.5648	0.6738	0.2661	0.2418	0.2509
	$MOCD_{SNN}$	**0.8561**	**0.5834**	**0.6914**	**0.2835**	**0.3015**	**0.2860**
10%	EA_Q	0.3124	0.3169	0.3146	0.1071	0.2533	0.1505
	EA_{QD}	0.5164	0.3661	0.4284	0.2897	0.1871	0.1917
	EA_{CS}	0.626	0.4362	0.5141	0.2079	0.1995	0.2034
	$MOCD_1$	0.4761	0.2945	0.363	0.1347	0.1305	0.1322
	$MOCD_2$	0.8101	0.5283	0.6386	0.2553	0.232	0.2405
	$MOCD_{SNN}$	**0.8244**	**0.5337**	**0.6418**	**0.2664**	**0.2376**	**0.2581**
20%	EA_Q	0.2972	0.3015	0.2993	0.1071	0.2493	0.1497
	EA_{QD}	0.5065	0.3585	0.4199	0.294	0.197	0.2031
	EA_{CS}	0.6159	0.4357	0.5103	0.2118	0.2055	0.2084
	$MOCD_1$	0.4369	0.271	0.3336	0.1254	0.1191	0.1217
	$MOCD_2$	0.8191	0.5492	0.656	0.2523	0.2331	0.2405
	$MOCD_{SNN}$	**0.8247**	**0.5600**	**0.6618**	**0.2579**	**0.2418**	**0.2482**
30%	EA_Q	0.2993	0.3037	0.3015	0.1066	0.2475	0.1489
	EA_{QD}	0.4933	0.3526	0.4112	0.2933	0.1962	0.2022
	EA_{CS}	0.5795	0.4141	0.483	0.2136	0.206	0.2097
	$MOCD_1$	0.4298	0.2635	0.3259	0.1141	0.1095	0.1113
	$MOCD_2$	0.7615	0.5152	0.6136	0.2532	0.236	0.243
	$MOCD_{SNN}$	**0.7724**	**0.5213**	**0.6201**	**0.5617**	**0.2394**	**0.2447**
40%	EA_Q	0.2689	0.2728	0.2708	0.0972	0.2157	0.1339
	EA_{QD}	0.4587	0.3315	0.3848	0.2957	0.1964	0.202
	EA_{CS}	0.5298	0.3812	0.4433	0.2144	0.204	0.2091
	$MOCD_1$	0.3607	0.2311	0.2811	0.1119	0.109	0.1099
	$MOCD_2$	0.6679	0.4707	0.5513	0.2408	0.2262	0.2315
	$MOCD_{SNN}$	**0.7215**	**0.4865**	**0.5694**	**0.2537**	**0.2374**	**0.2385**
50%	EA_Q	0.1541	0.165	0.1594	0.0503	0.1316	0.0727
	EA_{QD}	0.4054	0.3204	0.3579	0.29	0.1893	0.1808
	EA_{CS}	0.4775	0.3743	0.4196	0.1831	0.1988	0.1906
	$MOCD_1$	0.2146	0.1603	0.1833	0.0698	0.0853	0.0765
	$MOCD_2$	0.5796	0.4268	0.4909	0.2182	0.2264	0.2203
	$MOCD_{SNN}$	**0.6750**	**0.4676**	**0.5004**	**0.2248**	**0.2371**	**0.2261**

of false interactions to protein pairs. The total number of interactions is increased to $m = 5859$ after adding the false interaction, and the protein with one interaction is reduced to $|n|_{d=1} = 3$ proteins, with an average of this spurious PPI network being $d_{avg} = 11.8364$ interactions per protein. In the bottom center, it represents the addition of false interactions to the protein with the highest degree of interactions. The total number of interactions is increased to be $m = 6520$, and the protein with one interaction is reduced to be $|n|_{d=1} = 5$ proteins, with an average of this spurious PPI network is $d_{avg} = 13.1727$ interactions per protein. In the right corner, it represents the addition of false interactions to the protein with least degree of interactions. The total number of interactions is increased to be $m = 6651$, and since the addition is to the proteins with the fewest interactions, hence, $|n|_{d=1} = 0$, the average of this spurious PPI network is $d_{avg} = 13.4364$. On the other hand, Figure 8 depicts the Saccharomyces cerevisiae yeast PPI network, which is represented by the original Yeast-D1 PPI network shown at the top, which contains $n = 990$ proteins with $m = 4687$ interactions, as the proteins that contain one interaction is $|n|_{d=1} = 28$ proteins, with an average of this network is $d_{avg} = 9.4687$ interactions for each protein. In the bottom left, it represents the random deletion of true interactions from protein pairs. The total number of interactions is decreased to $m = 2344$ after removing

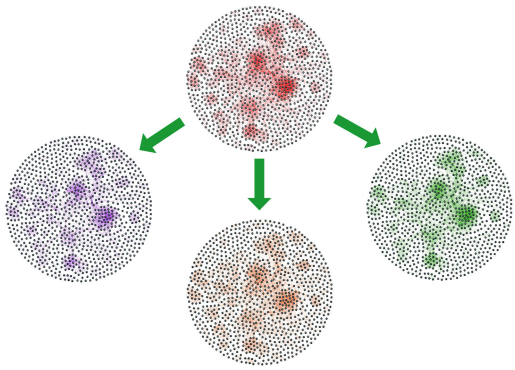


FIGURE 8. Yeast-D1 (top) and three artificial PPI networks generated by deleting true interactions from Yeast-D1.

the true interaction, and the protein with one interaction is increased to $|n|_d = 1 = 90$ proteins, with an average of this spurious PPI network being $d_avg = 4.7354$ interactions per protein. In the bottom center, it represents the deletion of true interactions from the protein with highest degree of interactions. The total number of interactions is decreased to be $m = 3519$, and the protein with one interaction is increased to be $|n|_{d=1} = 29$ proteins, with an average of this spurious PPI network is $d_{avg} = 7.1091$ interactions per protein. In the right corner, it represents the deletion of true interactions from

TABLE 17. Robustness evaluation in terms of recall, precision, and F. False interactions are randomly added to protein pairs.

Noise	Algorithm	Yeast-D1 Recall	Yeast-D1 Precision	Yeast-D1 F	Yeast-D2 Recall	Yeast-D2 Precision	Yeast-D2 F
0%	EA_Q	0.1855	0.5371	0.5356	0.3162	0.2669	0.2893
	EA_{QD}	0.6034	0.5768	0.5896	0.3407	0.2810	0.3078
	EA_{CS}	0.6090	0.5272	0.5649	0.3453	0.2571	0.2947
	$MOCD_1$	0.7051	0.5515	0.6060	0.3853	0.2858	0.3223
	$MOCD_2$	0.8359	0.6023	0.6898	0.4453	0.3114	0.3507
	$MOCD_{SNN}$	**0.8423**	**0.6462**	**0.6907**	**0.4976**	**0.3706**	**0.4460**
10%	EA_Q	0.4545	0.5196	0.4844	0.3130	0.3201	0.3159
	EA_{QD}	0.4872	0.5367	0.5104	0.2491	0.2555	0.2519
	EA_{CS}	0.5462	0.4965	0.5196	0.2769	0.2333	0.2529
	$MOCD_1$	0.5407	0.5728	0.5362	0.2742	0.2828	0.2690
	$MOCD_2$	0.8231	0.6000	0.6835	0.4227	0.3079	0.3445
	$MOCD_{SNN}$	**0.8821**	**0.7077**	**0.7138**	**0.6960**	**0.4257**	**0.4857**
20%	EA_Q	0.3427	0.4534	0.3897	0.2388	0.2889	0.2607
	EA_{QD}	0.3863	0.5069	0.4378	0.1682	0.2133	0.1872
	EA_{CS}	0.4585	0.4440	0.4508	0.2053	0.1985	0.2014
	$MOCD_1$	0.4680	0.5652	0.4812	0.2362	0.2814	0.2423
	$MOCD_2$	0.8154	0.6060	0.6870	0.3927	0.2997	0.3284
	$MOCD_{SNN}$	**0.8756**	**0.7012**	**0.7043**	**0.6473**	**0.4174**	**0.4812**
30%	EA_Q	0.2713	.04054	0.3242	0.1801	0.2552	0.2102
	EA_{QD}	0.2962	0.4566	0.3585	0.1333	0.2192	0.1654
	EA_{CS}	0.3726	0.4036	0.3867	0.1724	0.1920	0.1811
	$MOCD_1$	0.4154	0.5643	0.4348	0.2012	0.2909	0.2161
	$MOCD_2$	0.8128	0.6160	0.6896	0.3867	0.3006	0.3261
	$MOCD_{SNN}$	**0.8667**	**0.6731**	**0.6915**	**0.6380**	**0.4513**	**0.4796**
40%	EA_Q	0.1890	0.3217	0.2375	0.1310	0.2134	0.1612
	EA_{QD}	0.2060	0.4014	0.2715	0.1084	0.2234	0.1453
	EA_{CS}	0.3145	0.3593	0.3350	0.1413	0.1812	0.1586
	$MOCD_1$	0.3437	0.5218	0.3588	0.1610	0.2966	0.1767
	$MOCD_2$	0.8038	0.6012	0.6731	0.3853	0.211	0.3275
	$MOCD_{SNN}$	**0.8654**	**0.6425**	**0.6942**	**0.6120**	**0.4167**	**0.4581**
50%	EA_Q	0.1387	0.2669	0.1819	0.0942	0.1804	0.1229
	EA_{QD}	0.1376	0.3338	0.1942	0.0649	0.1679	0.0932
	EA_{CS}	0.2509	0.3097	0.2767	0.1076	0.1442	0.1224
	$MOCD_1$	0.3005	0.4735	0.3067	0.1406	0.2803	0.1494
	$MOCD_2$	0.7795	0.6009	0.6654	0.3727	0.3641	0.3185
	$MOCD_{SNN}$	**0.8513**	**0.6439**	**0.6835**	**0.5667**	**0.4032**	**0.4370**

the protein with least degree of interactions. The total number of interactions is decreased to be $m = 3476$, and since the addition is to the proteins with the fewest interactions, hence, $|n|_{d=1} = 207$, the average of this spurious PPI network is $d_{avg} = 7.0222$ interactions per protein. It is worth to note that all of these additions and deletions are with extremely high noise percentages of 50%.

The results of the following tables report performance comparison at both complex and protein levels for both Yeast-D1 and Yeast-D2 and their synthesized noisy networks. The overlapping score is fixed to $\sigma OS = 0.5$. The best results in each table are highlighted in bold.

Table 3 report recall, precision, and F measures for Yeast-D1, Yeast-D2, and HPRD datasets. The results point out that multi-objective based on proposed heuristic operator, $MOCD_{SNN}$ beats all single and multi-objective state-of-the art models. Furthermore, Table 4 reports the Wilcoxon signed ranked test for the results reported in the Table 3. The test is calculated with significance level $\alpha = 0.05$. The results are reported as $(p-value)$, where $p-value$ means the probability that a success of the proposed multi-objective based on heuristic operator $MOCD_{SNN}$ over the counterpart state-of-the-art models. The results are given in bold if $p-value \leq \alpha$ (means satisfying significance requirement).

Table 17, Table 6, and Table 7 reports the evaluation of robustness (in terms of *Recall*, *Precision*, and F) for both Yeast-D1 and Yeast-D2 and their synthesized noisy networks, where noise is added to the original networks. Table 8, Table 9, and Table 10, on the other hand, report the evaluation of robustness (in terms of *Recall*, *Precision*, and F) for Yeast-D1 and Yeast-D2 and their synthesized noisy networks, where true interactions are deleted from the networks.

For protein level, Table 11, Table 12, and Table 13 report the evaluation of robustness (in terms of $Recall_N$, $Precision_N$, and F_N) for both Yeast-D1 and Yeast-D2 and their synthesized noisy networks, where false interactions are added to the networks. On the other hand, Table 14, Table 15, and Table 16 report the evaluation of robustness (in terms of $Recall_N$, $Precision_N$, and F_N) for the two networks and their synthesized noisy networks where true interactions are deleted with different percentage (10%, 20%, 30%, 40% and 50%).

The figures presented in this study (Figure 9, Figure 10, Figure 11) showcase the impact of three distinct types of noise on two yeast PPI networks, Yeast-D1 and Yeast-D2. Each graph consists of two rows, where the first row details PPV, sensitivity, and accuracy statistics based on equations (25, 24, and 26) for Yeast-D1. The second row mirrors the same statistics but for Yeast-D2. These graphs serve to illustrate the robustness of our proposed method ($MOCD_{SNN}$) against the introduction of spurious interactions into the original network, with proportions varying from 0% to 50% as indicated on the x-axis. For each proportion, we systematically compare the performance of

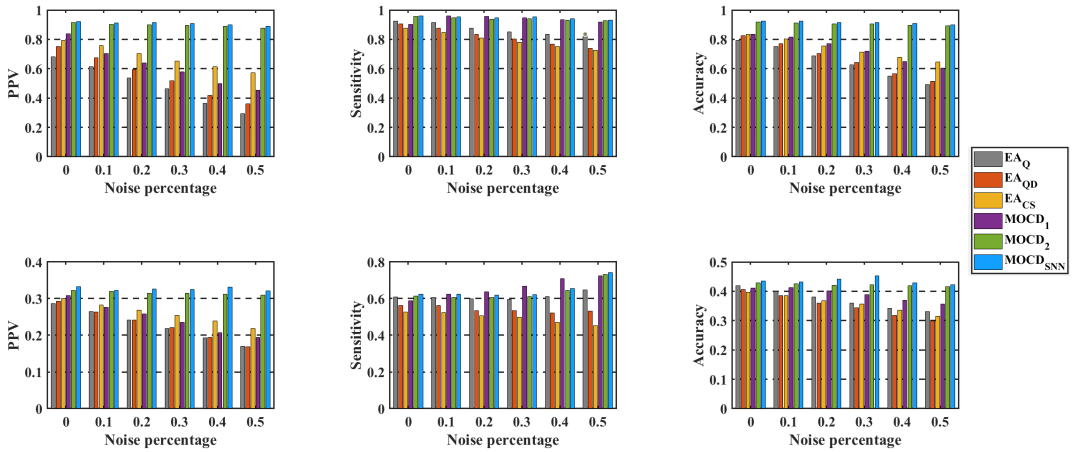


FIGURE 9. An evaluation of Yeast-D1 (top) and Yeast-D2 (bottom). False interactions are randomly added.

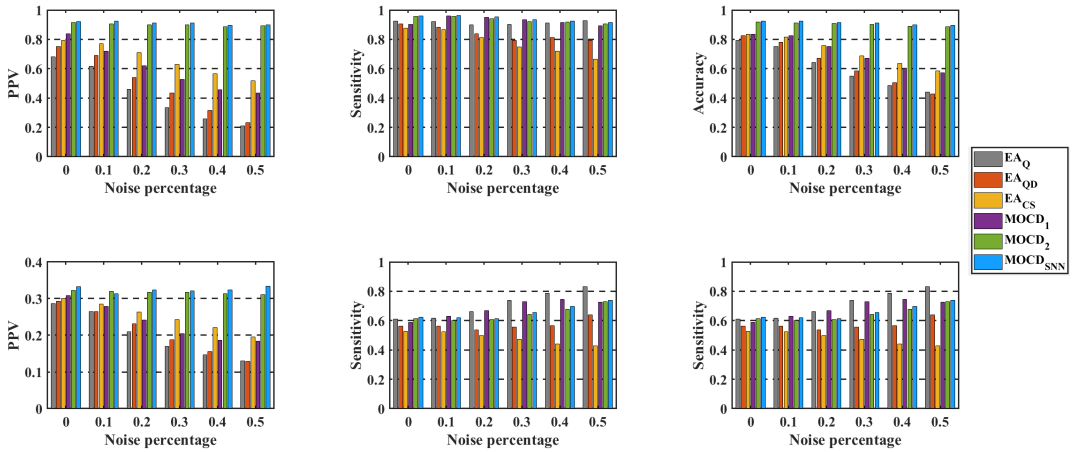


FIGURE 10. An evaluation of Yeast-D1 (top) and Yeast-D2 (bottom). False interactions are added to highly-connected proteins.

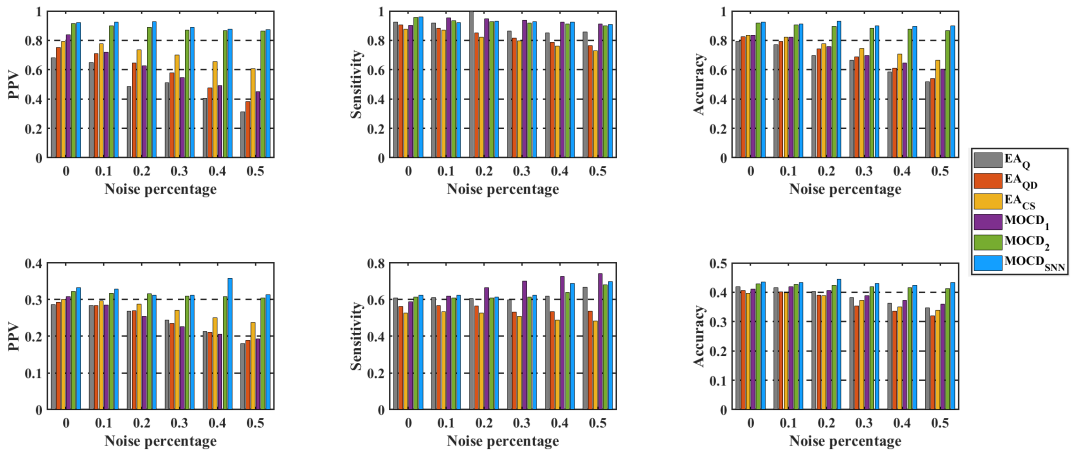


FIGURE 11. An evaluation of Yeast-D1 (top) and Yeast-D2 (bottom). False interactions are added to minimally-connected proteins.

state-of-the-art methods alongside our proposed method, represented by the distinctive blue bar. Significantly, our proposed method ($MOCD_{SNN}$) consistently outperforms all previous approaches, both in the realm of single-objective (EA_Q, EA_{QD}, EA_{CS}) and multi-objective ($MOCD_1$, $MOCD_2$) evaluations.

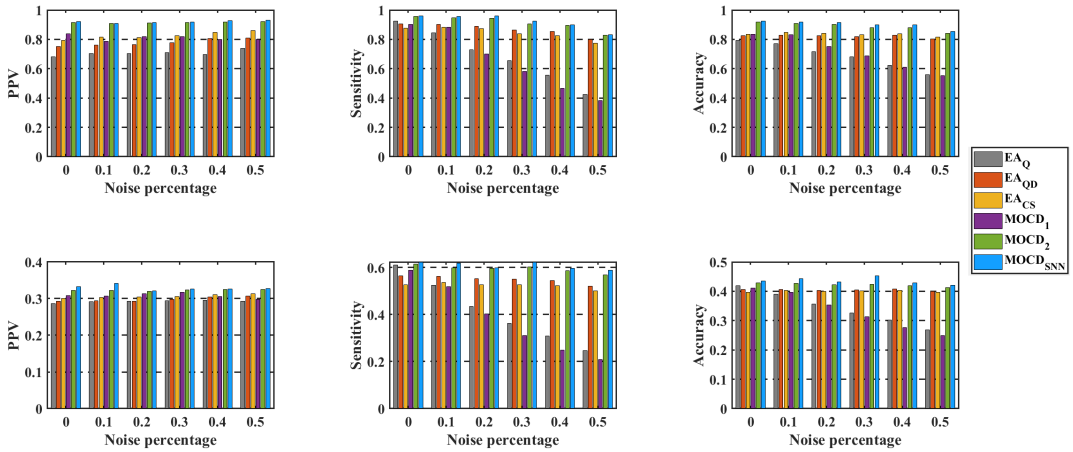


FIGURE 12. An evaluation of Yeast-D1 (top) and Yeast-D2 (bottom). True interactions are randomly deleted.

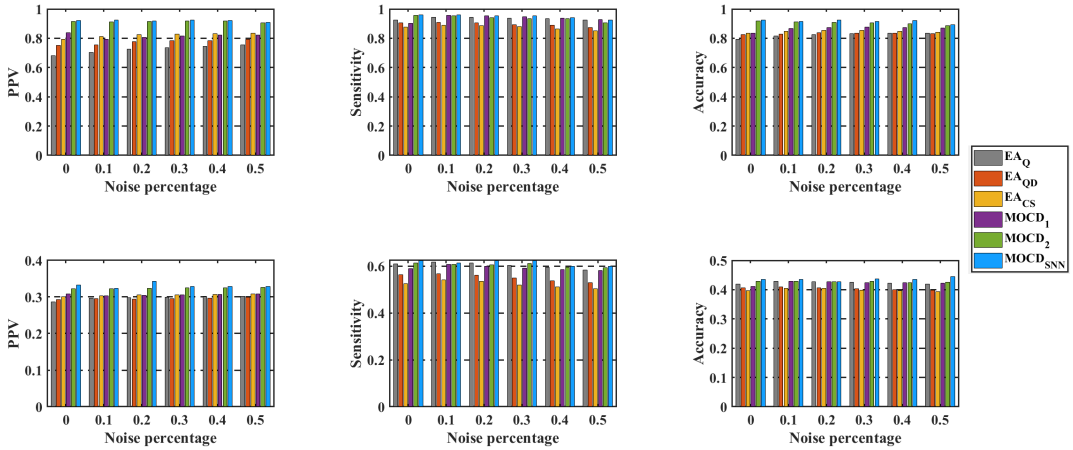


FIGURE 13. An evaluation of Yeast-D1 (top) and Yeast-D2 (bottom). True interactions are deleted from highly-connected proteins.

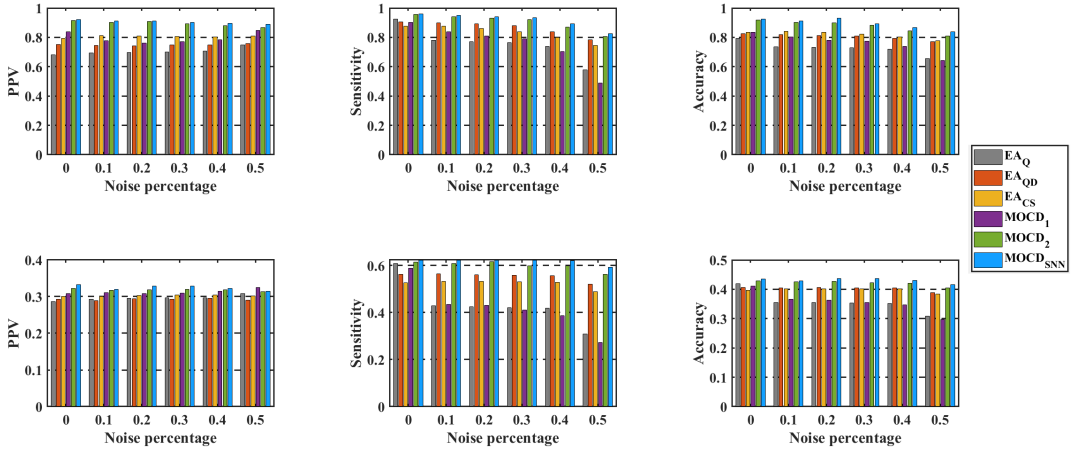


FIGURE 14. An evaluation of Yeast-D1 (top) and Yeast-D2 (bottom). True interactions are deleted from minimally-connected proteins.

In contrast, the figures presented in another context (Figure 12, Figure 13, Figure 14) delve into the repercussions of three types of noise introduced into two yeast PPI networks, Yeast-D1 and Yeast-D2. Similar to the previous scenario, each graph features two rows. The first row delineates PPV, sensitivity, and accuracy matching statistics

for Yeast-D1, while the second row mirrors these statistics for Yeast-D2. These graphs illuminate the robustness of our proposed method ($MOCD_{SNN}$) against the deletion of true interactions from the original network, with proportions ranging from 0% to 50% as indicated on the x-axis. As with the previous experiment, we meticulously compare the performance of established methods against our proposed method, symbolized by the consistent blue bar. Remarkably, our proposed method ($MOCD_{SNN}$) continues to outshine all previous approaches, spanning both single-objective (EA_Q, EA_{QD}, EA_{CS}) and multi-objective ($MOCD_1$, $MOCD_2$).

The overall results reveal the ability of the multi-objective model based on the proposed heuristic operator to outperform the detection ability of the state-of-the-art single and multi-objective models. Further, the definition of the complex detection problem as a multi-objective model also affects the performance of the MOEA. In the reported results, we found that $MOCD_{SNN}$ is even more robust than the counterpart MOEA, i.e. $MOCD_1$, and $MOCD_2$ in the presence of noisy interactions or in the absence of true interactions. This is mainly due to the formula used in the definition of $MOCD_{SNN}$ model (Eq. 9 and Eq. 11) and to the effectiveness of the proposed heuristic mutation operator.

E. COMPLEX DETECTION PERFORMANCE: SNN-BASED EVOLUTIONARY ALGORITHMS AGAINST STATE-OF-THE-ART METHODS

The *SNN* operator, designed to enhance the solution quality during the evolution of an EA tailored for the complex detection in PPI networks. The superiority of our proposed method can be attributed to its ability to address a critical aspect that has been relatively overlooked in existing state-of-the-art methods: the robustness of evolutionary algorithms when dealing with noisy or missing interactions in PPI networks. While existing methods often focus on partitioning PPI networks based on graph properties or biological semantics, our *SNN* operator specifically targets the topological attributes inherent to the protein level.

To elaborate, the *SNN* operator focuses on proteins classified as weaker entities within a given complex, defined by having fewer internal connections compared to external connections. These weaker proteins are crucial to the structural integrity of the complex, and our operator aims to enhance their placement within the evolving algorithm. By reassessing the classification of proteins based on the ''strong node'' criteria, as outlined in Eq. 14, our method ensures the preservation of the structural integrity of complexes. In cases where a protein fails to meet the criteria of a strong node, the *SNN* operator dynamically reassigns the complex of the neighboring proteins to the weak protein. This adaptability enables our method to handle scenarios where traditional methods might falter, ensuring a more accurate representation of the underlying biological reality in the presence of noise or missing data. Therefore, our proposed method's outperform of other state-of-the-art methods lies in its unique ability to adapt and improve the classification of weaker proteins within PPI networks, ultimately leading to more accurate and biologically relevant complex detection.

F. COMPUTATIONAL PARALLELIZATION

Due to their inherently evolutionary nature, EAs can readily accommodate parallelization across multiple generations. In our specific implementation, we have observed a significant performance enhancement, demonstrating a speedup proportional to the total number of available workers as opposed to a scenario with only a single worker. For configurations employing up to 88 cores, the speedup is computed as the ratio of the time (in seconds) required for the EA-based complex detection approach to complete its run when executed serially without any workers, to the time taken when executed with w cores (where $1 \leq w \leq 88$). It is imperative to note that our experimentation and analysis were conducted in the MATLAB R2022a environment, utilizing an Intel Xeon CPU E5-2699 v4 (2 sockets CPU's, each with 44 cores) with a base speed of 2.2GHz and 64 GB of RAM. Figure 15 illustrates the computational hardware employed for algorithm implementation.

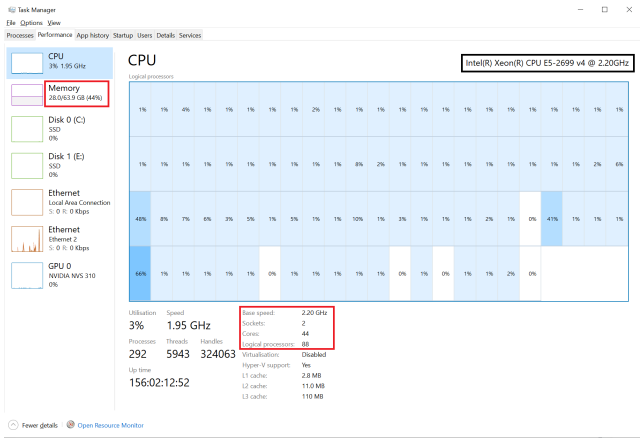


FIGURE 15. An illustration of the computational hardware utilized in implementing the algorithms.

V. CONCLUSION

In this paper, we proposed a novel heuristic operator and employed single and multi-objective EAs to assess the robustness of three single-objective models and two multi-objective models in the context of complex detection. The evaluation was conducted on three widely recognized datasets, comprising Saccharomyces cerevisiae (yeast) and human PPI networks, along with three benchmark sets of complexes. Additionally, we explored the impact of network perturbations, introducing various levels of noise to the original PPI network. Our experimental findings distinctly highlight the superiority of the multi-objective model, demonstrating a heightened level of prediction accuracy compared to other models. However, beyond the technical outcomes, it is imperative to discuss the practical implications of our work.

A. PRACTICAL IMPLICATIONS

1) BIOLOGICAL RELEVANCE

The observed success of the multi-objective model suggests its potential for practical application in understanding the intricacies of protein complexes. Future research should consider integrating biological information, such as gene ontology, to refine the objective function and heuristic operator. This could lead to the development of more biologically relevant and accurate models for protein complex detection.

2) BIOMEDICAL RESEARCH AND DRUG DISCOVERY

The accurate identification of protein complexes is crucial in biomedical research, especially in the context of diseases. Our findings provide a foundation for enhancing the efficiency of complex detection algorithms, thereby contributing to the identification of potential drug targets and understanding disease mechanisms.

3) NETWORK PERTURBATION STUDIES

The exploration of network perturbations and noise levels adds a practical dimension to our work. Understanding how these algorithms perform under varying conditions is instrumental in developing robust and adaptable models that can handle real-world biological data, which often contains inherent noise and uncertainties.

B. FUTURE DIRECTIONS

Encouraging further research to extend our work by incorporating biological information and designing objective functions based on this data will not only enhance the practical relevance of our findings but also open avenues for interdisciplinary collaboration between computer science and biology. In conclusion, our study not only advances the field of complex detection algorithms but also presents opportunities for impactful applications in biological and biomedical research, providing a bridge between computational methods and real-world biological phenomena.

REFERENCES

[1] A. H. Abdulateef, B. A. Attea, A. N. Rashid, and M. Al-Ani, "A new evolutionary algorithm with locally assisted heuristic for complex detection in protein interaction networks," *Appl. Soft Comput.*, vol. 73, pp. 1004–1025, Dec. 2018.

[2] V. Arnau, S. Mars, and I. Marín, "Iterative cluster analysis of protein interaction data," *Bioinformatics*, vol. 21, no. 3, pp. 364–378, Feb. 2005.

[3] N. Atias and R. Sharan, "Comparative analysis of protein networks: Hard problems, practical solutions," *Commun. ACM*, vol. 55, no. 5, pp. 88–97, May 2012.

[4] B. A. Attea and Q. Z. Abdullah, "Improving the performance of evolutionary-based complex detection models in protein–protein interaction networks," *Soft Comput.*, vol. 22, no. 11, pp. 3721–3744, Jun. 2018.

[5] G. D. Bader and C. W. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC Bioinf.*, vol. 4, no. 1, pp. 1–27, Jan. 2003.

[6] S. Bandyopadhyay, S. Ray, A. Mukhopadhyay, and U. Maulik, "A multiobjective approach for identifying protein complexes and studying their association in multiple disorders," *Algorithms Mol. Biol.*, vol. 10, no. 1, pp. 1–15, Dec. 2015.

[7] B. A. Attea and H. S. Khoder, "A new multi-objective evolutionary framework for community mining in dynamic social networks," *Swarm Evol. Comput.*, vol. 31, pp. 90–109, Dec. 2016.

[8] A.-L. Barabási and Z. N. Oltvai, "Network biology: Understanding the cell's functional organization," *Nature Rev. Genet.*, vol. 5, no. 2, pp. 101–113, Feb. 2004.

[9] B. A. Attea, A. D. Abbood, A. A. Hasan, C. Pizzuti, M. Al-Ani, S. Özdemir, and R. D. Al-Dabbagh, "A review of heuristics and metaheuristics for community detection in complex networks: Current usage, emerging development and future directions," *Swarm Evol. Comput.*, vol. 63, Jun. 2021, Art. no. 100885.

[10] B. A. Attea, H. M. Rada, M. N. Abbas, and S. Özdemir, "A new evolutionary multi-objective community mining algorithm for signed networks," *Appl. Soft Comput.*, vol. 85, Dec. 2019, Art. no. 105817.

[11] M. B. M'barek, A. Borgi, S. B. Hamida, and M. Rukoz, "Genetic algorithm to detect different sizes? Communities from protein–protein interaction networks," in *Proc. 14th Int. Conf. Softw. Technol.*, Jul. 2019, pp. 359–370.

[12] J. R. Bock and D. A. Gough, "Predicting protein–protein interactions from primary structure," *Bioinformatics*, vol. 17, no. 5, pp. 455–460, May 2001.

[13] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner, "On modularity clustering," *IEEE Trans. Knowl. data Eng.*, vol. 20, no. 2, pp. 172–188, Dec. 2007.

[14] S. Brohée and J. van Helden, "Evaluation of clustering algorithms for protein–protein interaction networks," *BMC Bioinf.*, vol. 7, no. 1, pp. 1–19, Dec. 2006.

[15] H. M. Burhan, B. A. Attea, A. D. Abbood, M. N. Abbas, and M. Al-Ani, "Evolutionary multi-objective set cover problem for task allocation in the Internet of Things," *Appl. Soft Comput.*, vol. 102, Apr. 2021, Art. no. 107097.

[16] M. Chen, K. Kuzmin, and B. K. Szymanski, "Community detection via maximization of modularity and its variants," *IEEE Trans. Computat. Social Syst.*, vol. 1, no. 1, pp. 46–65, Mar. 2014.

[17] Y.-R. Cho, W. Hwang, M. Ramanathan, and A. Zhang, "Semantic integration to identify overlapping functional modules in protein interaction networks," *BMC Bioinf.*, vol. 8, no. 1, pp. 1–13, Dec. 2007.

[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[19] A.-C. Gavin et al., "Proteome survey reveals modularity of the yeast cell machinery," *Nature*, vol. 440, no. 7084, pp. 631–636, Mar. 2006.

[20] E. Georgii, S. Dietmann, T. Uno, P. Pagel, and K. Tsuda, "Enumeration of condition-dependent dense modules in protein interaction networks," *Bioinformatics*, vol. 25, no. 7, pp. 933–940, Apr. 2009.

[21] M. Haque, R. Sarmah, and D. K. Bhattacharyya, "A common neighbor based technique to detect protein complexes in PPI networks," *J. Genetic Eng. Biotechnol.*, vol. 16, no. 1, pp. 227–238, Jun. 2018.

[22] B. A. Attea, W. A. Hariz, and M. F. Abdulhalim, "Improving the performance of evolutionary multi-objective co-clustering models for community detection in complex social networks," *Swarm Evol. Comput.*, vol. 26, pp. 137–156, Feb. 2016.

[23] S. Hashemifar, B. Neyshabur, A. A. Khan, and J. Xu, "Predicting protein–protein interactions through sequence-based deep learning," *Bioinformatics*, vol. 34, no. 17, pp. i802–i810, Sep. 2018.

[24] P. Jancura, E. Mavridou, E. Carrillo-de Santa Pau, and E. Marchiori, "A methodology for detecting the orthology signal in a PPI network at a functional complex level," *BMC Bioinf.*, vol. 13, no. S10, pp. 1–13, Jun. 2012.

[25] E. A. Khalil, S. Ozdemir, and B. A. Attea, "A new task allocation protocol for extending stability and operational periods in Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7225–7231, Aug. 2019.

[26] A. D. King, N. Pržulj, and I. Jurisica, "Protein complex prediction via cost-based clustering," *Bioinformatics*, vol. 20, no. 17, pp. 3013–3020, Nov. 2004.

[27] Z. A. King, J. Lu, A. Dräger, P. Miller, S. Federowicz, J. A. Lerman, A. Ebrahim, B. O. Palsson, and N. E. Lewis, "BiGG models: A platform for integrating, standardizing and sharing genome-scale models," *Nucleic Acids Res.*, vol. 44, no. D1, pp. D515–D522, Jan. 2016.

[28] N. Krogan et al., "Global landscape of protein complexes in the yeast Saccharomyces cerevisiae," *Nature*, vol. 440, no. 7084, pp. 637–643, Mar. 2006.

[29] X. L. Li, C. S. Foo, and S. K. Ng, "Discovering protein complexes in dense reliable neighborhoods of protein interaction networks," *Comput. Syst. Bioinf.*, vol. 6, pp. 157–168, Jan. 2007.

[30] J. X. Liu, J. C. Zeng, Y. W. Xue, and Y. Wang, "Quantitative function for community detection," *Adv. Mater. Res.*, vols. 433–440, pp. 6441–6446, Jan. 2012.
[31] I. Manipur, M. Giordano, M. Piccirillo, S. Parashuraman, and L. Maddalena, "Community detection in protein–protein interaction networks and applications," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 20, no. 1, pp. 217–237, Jan. 2023.
[32] X. Meng, X. Peng, F.-X. Wu, and M. Li, "Detecting protein complex based on hierarchical compressing network embedding," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Nov. 2019, pp. 215–218.
[33] X. Meng, J. Xiang, R. Zheng, F.-X. Wu, and M. Li, "DPCMNE: Detecting protein complexes from protein–protein interaction networks via multi-level network embedding," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 19, no. 3, pp. 1592–1602, May 2022.
[34] H. W. Mewes, "MIPS: A database for genomes and protein sequences," *Nucleic Acids Res.*, vol. 28, no. 1, pp. 37–40, Jan. 2000.
[35] T. Nepusz, H. Yu, and A. Paccanaro, "Detecting overlapping protein complexes in protein–protein interaction networks," *Nature Methods*, vol. 9, no. 5, pp. 471–472, May 2012.
[36] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, Feb. 2004, Art. no. 026113.
[37] S. Omranian, A. Angeleska, and Z. Nikoloski, "PC2P: Parameter-free network-based prediction of protein complexes," *Bioinformatics*, vol. 37, no. 1, pp. 73–81, Apr. 2021.
[38] S. Omranian and Z. Nikoloski, "CUBCO+: Prediction of protein complexes based on min-cut network partitioning into biclique spanned subgraphs," *Appl. Netw. Sci.*, vol. 7, no. 1, p. 71, Oct. 2022.
[39] M. Oti, "Predicting disease genes using protein–protein interactions," *J. Med. Genet.*, vol. 43, no. 8, pp. 691–698, Aug. 2006.
[40] S. Patra and A. Mohapatra, "Protein complex prediction in interaction network based on network motif," *Comput. Biol. Chem.*, vol. 89, Dec. 2020, Art. no. 107399.
[41] P. Pe and A. Zhang, "A two-step approach for clustering proteins based on protein interaction profile," in *Proc. 5th IEEE Symp. Bioinf. Bioeng. (BIBE)*, Oct. 2005, pp. 201–209.
[42] M. Pellegrini, M. Baglioni, and F. Geraci, "Protein complex prediction for large protein protein interaction networks with the core&peel method," *BMC Bioinf.*, vol. 17, no. S12, pp. 37–58, Oct. 2016.
[43] S. Peri, "Human protein reference database as a discovery resource for proteomics," *Nucleic Acids Res.*, vol. 32, pp. D497–D501, Jan. 2004.
[44] C. Pizzuti, "Community detection in social networks with genetic algorithms," in *Proc. 10th Annu. Conf. Genetic Evol. Comput.*, Jul. 2008, pp. 1137–1138.
[45] C. Pizzuti, "Ga-Net: A genetic algorithm for community detection in social networks," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Springer, Sep. 2008, pp. 1081–1090.
[46] C. Pizzuti, "Computational intelligence for community detection in complex networks and bio-medical applications," Ph.D. dissertation, 2014. [Online]. Available: http://hdl.handle.net/2066/129790
[47] C. Pizzuti and S. E. Rombo, "Algorithms and tools for protein–protein interaction networks clustering, with a special focus on population-based stochastic methods," *Bioinformatics*, vol. 30, no. 10, pp. 1343–1352, May 2014.
[48] S. E. Schaeffer, "Graph clustering," *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 27–64, 2007.
[49] R. Sharan, I. Ulitsky, and R. Shamir, "Network-based prediction of protein function," *Mol. Syst. Biol.*, vol. 3, no. 1, p. 88, 2007.
[50] X. Shen, L. Yi, X. Jiang, T. He, J. Yang, W. Xie, P. Hu, and X. Hu, "Identifying protein complex by integrating characteristic of core-attachment into dynamic PPI network," *PLoS ONE*, vol. 12, no. 10, Oct. 2017, Art. no. e0186134.
[51] C. Von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork, "Comparative assessment of large-scale data sets of protein–protein interactions," *Nature*, vol. 417, no. 6887, pp. 399–403, May 2002.
[52] S. Wang, R. Wu, J. Lu, Y. Jiang, T. Huang, and Y. D. Cai, "Protein–protein interaction networks as miners of biological discovery," *Proteomics*, vol. 22, nos. 15-16, Aug. 2022, Art. no. 2100190.
[53] N. Zaki, J. Berengueres, and D. Efimov, "Detection of protein complexes using a protein ranking algorithm," *Proteins, Struct. Function, Bioinf.*, vol. 80, no. 10, pp. 2459–2468, Oct. 2012.
[54] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

MUSTAFA N. ABBAS received the B.S. and M.S. degrees from the Department of Computer Science, University of Baghdad, Baghdad, Iraq, in 2014 and 2019, respectively. He is currently pursuing the Ph.D. degree in computer science with Otto-von-Guericke-University Magdeburg, Germany. His research interests include computational intelligence, multi-objective evolutionary algorithms, bioinformatics, and wireless sensor networks.

BARA'A A. ATTEA received the B.S. and M.S. degrees in computer science from the University of Baghdad, Baghdad, Iraq, in 1993 and 1996, respectively, and the Ph.D. degree in computer science from the University of Technology, Baghdad, in 2002. From 2011 to 2013, she was a Visiting Researcher with Gazi University, Ankara, Turkey. She is currently a Professor with the Department of Computer Science, University of Baghdad. Her main research interests include computational intelligence, multi-objective evolutionary algorithms, bioinformatics, and applications of bio-inspired algorithms in solving real-world problems, such as complex social network analysis and wireless sensor networks.

DAVID BRONESKE received the bachelor's, master's, and Ph.D. degrees in computer science from Otto-von-Guericke-University Magdeburg. He is currently the Head of the Department for Infrastructure and Methods, German Centre for Higher Education Research and Science Studies (DZHW), Hannover. His research interests include main-memory database systems, interdisciplinary data management, and the application of artificial intelligence in various domains.

GUNTER SAAKE received the Ph.D. degree from the Technical University of Braunschweig, in 1988. He is currently the Head of the Research Group Databases and Software Engineering, Otto-von-Guericke-University Magdeburg. He is also a Full Professor of computer science. His research interests include database integration, tailor-made data management, database management on new hardware, and feature-oriented software product lines.