ISSN: 0067-2904

An Evolutionary Algorithm with Gene Ontology-Aware Crossover Operator for Protein Complex Detection

Isra H. Abdulateef[1,2], Dhia A. Alzubaydi[1,3], Bara'a Ali Attea[2*]
[1]*Department of Computer Science, College of Science, Al-Mustansiriyah University, Baghdad, Iraq*
[2]*Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq*
[3]*Al-Rasheed University College, Baghdad, Iraq*

Received: 20/1/2022 Accepted: 3/8/2022 Published: 30/4/2023

Abstract

Evolutionary algorithms (EAs), as global search methods, are proved to be more robust than their counterpart local heuristics for detecting protein complexes in protein-protein interaction (PPI) networks. Typically, the source of robustness of these EAs comes from their components and parameters. These components are solution representation, selection, crossover, and mutation. Unfortunately, almost all EA based complex detection methods suggested in the literature were designed with only canonical or traditional components. Further, topological structure of the protein network is the main information that is used in the design of almost all such components. The main contribution of this paper is to formulate a more robust EA with more biological consistency. For this purpose, a new crossover operator is suggested where biological information in terms of both gene semantic similarity and protein functional similarity is fed into its design. To reflect the heuristic roles of both semantic and functional similarities, this paper introduces two gene ontology (GO) aware crossover operators. These are direct annotation-aware and inherited annotation-aware crossover operators. The first strategy is handled with the direct gene ontology annotation of the proteins, while the second strategy is handled with the directed acyclic graph (DAG) of each gene ontology term in the gene product. To conduct our experiments, the proposed EAs with GO-aware crossover operators are compared against the state-of-the-art heuristic, canonical EAs with the traditional crossover operator, and GO-based EAs. Simulation results are evaluated in terms of recall, precision, and F measure at both complex level and protein level. The results prove that the new EA design encourages a more reliable treatment of exploration and exploitation and, thus, improves the detection ability for more accurate protein complex structures.

Keywords: Evolutionary algorithm; gene ontology; protein complex; protein-protein interaction network; semantic similarity.

خوارزمية تطورية مع عامل خلط الحلول مدرك لعلم الجينات لأكتشاف المركبات البروتينية

أسراء هيثم عبد اللطيف[1و2]، ضياء الزبيدي[1و3]،براء علي عطية[2*]
[1]قسم الحاسبات ، كلية العلوم ، الجامعة المستنصرية ، بغداد ، العراق
[2]قسم الحاسبات ، كلية العلوم ، جامعة بغداد ، بغداد ، العراق
[3] كلية الرشيد الجامعة، بغداد، العراق

*Email: bara.a@sc.uobaghdad.edu.iq

الخلاصة:

أثبتت الخوارزميات التطورية (EAs) ، كطرق بحث عامة ، أنها أكثر قوة من أساليب البحث المحلية لاكتشاف المركبات البروتينية في شبكات تفاعل البروتين البروتين (PPI). عادةً ما يأتي مصدر قوة هذه الخوارزميات من مكوناتها. هذه المكونات هي تمثيل الحل، والاختيار، خلط الحلول، والطفرة. مع هذه الأهمية ولسوء الحظ ، أغلب طرق أكتشاف المركبات البروتينية والتي تم تصميمها في الأدبيات والقائمة على EA تقريبًا كانت مُقترحة الأدبيات باستعمال مكونات أساسية أو تقليدية فقط. علاوة على ذلك ، فإن البنية الطوبولوجية لشبكة البروتين هي المعلومات الرئيسية المستخدمة في تصميم جميع هذه المكونات تقريبًا. المساهمة الرئيسية لهذا البحث هي صياغة EA أكثر قوة مع مزيد من الاتساق البيولوجي. لهذا الغرض، تم اقتراح عامل كخلط حلول جديد حيث يتم تغذية المعلومات البيولوجية من حيث التشابه الدلالي الجيني والتشابه الوظيفي للبروتين في تصميمه. وعلى هذا الأساس، نقدم اثنين من مشغلي خلط الحلول المدركين لعلم الوجود الجيني (GO)، ألأول يتعامل مع أوجه التشابه الجيني بين البروتينات بشكل مباشر، بينما يتم التعامل مع الإستراتيجية الثانية باستعمال الرسم البياني غير الدوري الموجه (DAG). لإجراء تجاربنا، تتم مقارنة الخوارزمية المقترحة مع الخوارزميات ذات التوجيه المحلي و الخوارزميات التطورية المتعارف عليها مع مشغل خلط الحلول التقليدي و الخوارزميات التطورية المبنية على أساس علم الجينات. يتم تقييم المحاكاة من حيث الاسترجاع والدقة وقياس F على كل من المستوى المركب البروتيني وعلى مستوى البروتين. أثبتت النتائج أن تصميم EA الجديد يشجع على معالجة أكثر موثوقية للاستكشاف والاستغلال ، وبالتالي ، يحسن القدرة على الكشف بأكثر دقة عن الهياكل المركبة للبروتين.

1. Introduction

Execution of a genetic program, including those with harmful genes and encoded proteins for example COVID-19, has a very harmful effect. Actually, protein complexes and functional modules are formed as a physical aggregations and molecular interactions of different protein-protein interactions (PPIs), and protein-protein interaction networks (PPINs). Thus, identification of protein complexes (or functional modules) is a critical problem in biology systems in any living organism. Typically, detecting protein complexes from PPIN, and generally, the areas of identifying a priori unknown building blocks from complex networks is known as bi-clustering or co-clustering problem [1, 2]. It is defined as a natural division of a complex network which follows a general heterogeneous connections rule, known as *modules* or *communities* where a densely intra-connected module of nodes is also sparsely inter-connected with other modules [3]. Bi-clustering problem is recently reporting an increasing interest. Unfortunately, akin to many real-world optimization problems, the computational complexity of protein complex detection problem falls into the category of non-deterministic polynomial time hard (NP-hard) problems [4, 5].

Unlike heuristic, metaheuristics and evolutionary algorithms (EAs) are proved to be a sustainable alternative to solve NP-hard problems while accommodating their combinatorial explosion [6, 7]. For complex detection problem, Pizzuti and Rombo in 2014 [8] were the first to show that evolutionary based complex detection methods are more robust than other state-of-the-art heuristic-based complex detection methods. Unfortunately, almost the design of the main components of all these EA-based complex detection methods is either canonical or guided by a general topological characteristic of communities and modules. For example, Pizzuti and Rombo [8] expressed a canonical single objective EAs to detect protein complexes and showed the encouraging performance of EAs to outperform the counterpart heuristic methods. Another EA-based complex detection algorithms were proposed in [9] and [10]. However, in both algorithms a topological-based mutation operator is designed. The basic idea of the designed topology-aware mutation operator is to breakdown the coexistence of a pair of proteins according to their topological similarity. Their interactions can serve for either intra-delineation topology or inter-delineation topology. The design of an EA with a topology-based

component (e.g., mutation operator in [9] and [10]) is proved to harnesses the detection ability of several single and multi-objective topology-based optimization models (such as modularity, community fitness, community score, conductance, expansion, internal density, inter-score and intra-score, normalized cut, negative ration association, and ratio cut).
Unfortunately, the current effort in literature to design evolutionary-based complex detection methods with gene ontology (GO) aware components is still lagging behind. Only a few works in literature examine incorporation of GO semantic similarity into design of evolutionary algorithms. Recently, the authors in [11] adopted the EA of Pizzuti and Rombo [8] with their single-objective models and the topological-based migration operator of Attea and Q. Z. Abdullah [9]. However, they reflect similarity values of the Gene Ontology Consistency (GOC) metric to let the migration operator to find the elected complexes for the mutated proteins.

The key contribution of this paper is to design an evolutionary-based complex detection algorithm with GO-aware crossover operator. The proposed crossover operator is viewed with two different manifestations of GO-based heuristics. The remaining of this paper is organized as follows. Section 2 presents a brief introduction to the graph and ontology means of PPI networks. This is followed by Section 3 while introducing the proposed EA with GO-aware crossover operator. Two formulations are suggested for the proposed GO-aware crossover operator. The results and discussions are provided in Section 4, demonstrate that it is curious enough to develop an EA with only non-ontology-based complex detection algorithms. Finally, conclusions and future directions are provided in Section 5.

2. Preliminary concept

A protein-protein interaction network (PPIN) is generally formulated as a finite heterogeneous network $\mathcal{N}(n, \mathbb{E})$ of a set of n proteins (i.e. $\mathbb{P} = \{P_1, P_2, \dots, P_n\}$) and a set of m interactions connecting pairs of proteins (i.e. $\mathbb{E} = \{E_1, E_2, \dots, E_m\}$). Thus, $\mathcal{N}$ is mathematically expressed as a graph $\mathcal{G}$ of a set of n vertices and m edges. For a protein or vertex P_i, its degree, d_i, is defined as the number of interactions incorporating P_i. Further, the data representation of the graph $\mathcal{G}$ is usually denoted as a square, symmetric and binary matrix called adjacency matrix $\mathrm{A} = [a_{i,j}]^{n \times n}$, where protein pair P_i and P_j are adjacent, and thus $a_{i,j} = 1$ when there is an interaction between P_i and P_j, otherwise, $a_{i,j} = 0$.

Note that, the adjacency matrix A contains all possible decompositions of the network $\mathcal{N}$ into different number of square sub-matrices with different sizes. For complex detection problem, the main challenge is that the number of complexes, K, used to partition a PPI network is unknown. However, a protein P_i in a complex C can be quantified by the degree of its intra-connections with other proteins within C and inter-connections with proteins in other complexes [8] [9].

Gene Ontology (GO) is the most common biology-focused and animated controlled vocabulary (CV) devoted to the functional annotation of proteins (i.e. gene products) in a cellular context and a species independent manner [12]. In CV, each GO term, t, is assigned a unique alphanumeric code (e.g., 'YMR091C', 'YLR033W', 'YMR033W', 'YPR023C', 'YDR073W', 'YFL049W', 'YGR275W', 'YJL002C', 'YMR149W', 'YNL078W', and 'YML112W') and is used to annotate genes and gene products (i.e. proteins). GO is divided into three sub-ontologies. These are Molecular Function (MF), Biological Process (BP), and Cellular Component (CC). Then, the statement of a connection between a type of gene product and the types designated by terms in the GO is called a GO annotation (GOA) [13, 14]. In other words, gene products are annotated with GO terms, either directly or through inheritance (true path rule), since annotation to a given term implies annotation to all of its ancestors.

Each sub-ontology is represented by a network or an independent directed acyclic graph (DAG), where individual GO terms that describe components of a gene product function are nodes in the DAG and connected by directional edges [15]. These directional edges are most commonly of the types 'is_a' and 'part of', where the 'is_a' denotes a simple class–subclass relationship and 'part of' denotes a part–whole relationship. Also, in DAG, each node may have more than one parent as well as zero, one, or more children. For example, consider the DAG in Figure 1. The DAG is for a GO term of *BP* sub-ontology (septum digestion after cytokinesis: 0000920). It represents the DAG for septum digestion after cytokinesis GO: 0000920 BP term. Other terms (nodes in the DAG) represent functional feature description, while the directional edges form relations between the terms.

Two types of semantic similarity can be obtained from n proteins annotated with N GO terms. These are term semantic similarity (SS) and functional similarity (FS). Term semantic similarity (SS) quantifies the specificity of terms and the closeness or relatedness and difference between terms within an ontology. Thus, from N terms, a semantic-based square similarity matrix $S = [SS_{i,j}]^{N \times N}$ is obtained, where $SS_{i,j}$ quantifies the semantic similarity between GO term t_i and GO term t_j. In protein-level annotation, two sets of GO terms within a specific category (i.e. MF, BP, or CC) are required to assess the functional similarity (FS) between two proteins. The functional similarity quantifies the functional similarity between pair of proteins based on their GO terms. Similarly, for n proteins, a functional-based square similarity matrix $F = [FS_{i,j}]^{n \times n}$ can be constructed, where $FS_{i,j}$ quantifies the functional similarity between protein P_i and protein P_j.

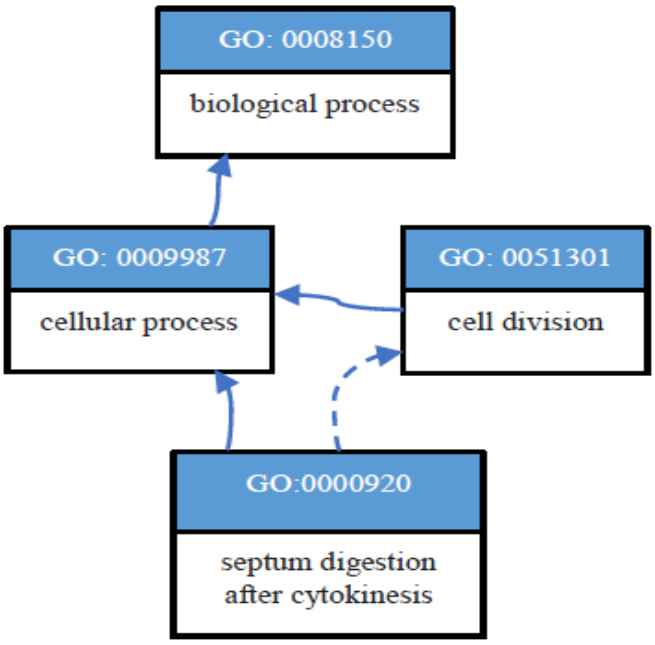

Figure 1: DAG for a GO term of *BP* sub-ontology (septum digestion after cytokinesis: 0000920).

Broadly speaking, functional similarity, FS, can be divided into two approaches: pairwise and group-wise [15, 16]. In pairwise approaches, FS between two proteins, P_1 and P_2, with their annotating terms, T_{P_1} and T_{P_2}, respectively, is evaluated by combining the semantic similarity, SS, of the pairwise terms in T_{P_1} and T_{P_2}. Pairwise combination approach could use all pairs or

best pairs methods. A global FS is then statistically obtained (with average, sum, maximum, or minimum threshold). Some of the well-known FS closeness measures are average inter-set similarity, maximum similarity, and average of maximum similarity.

Group-wise similarity measures, on the other hand, can be further classified into set-based, graph-based, and vector-based. Set-based methods consider only direct annotations for gene products while holding off the impact of the shared ancestry between GO terms. In set-based methods, only traditional cardinality-based similarity measures such as Jaccard and Dice measures are used. Several graph matching methods and set similarity measures such as term overlap and normalized term overlap are developed for graph-based approaches. In vector-based approaches, proteins are described as binary or weighted vectors of GO terms. For example, inverse document frequency is associated as a weight for each GO term counting the number of occurrence of this GO term in the whole corpus of gene products [17].

3. The proposed EA with GO-aware crossover operator

Any Evolutionary Algorithm (EA) is simply defined as a search mechanism to find the most applicable solution from a set of all possible solutions for the problem at hand. An EA searches for good solutions while iteratively evaluates a population of individual solutions, and performs three main evolutionary operators (i.e. selection, crossover, and mutation). The canonical design of these operators (particularly crossover and mutation) can be used as general operators for almost all types of optimization problems. However, for a particular problem, the problem-specific design of such evolutionary operators would then determine the characteristic of the adopted EA and would improve its performance. In this section, the definition of an EA with all its components is defined while the formulation of all its components (including the crossover operator) is relaxed for the purpose of complex detection problem in PPI networks.

3.1 Canonical EA for complex detection problem

In the adopted EA, the locus-based representation of Handl and Knowles [14] [19] is adopted. A population, $\mathbb{I} = \{I_1, I_2, \dots, I_{pop-size}\}$, of $pop - size$ individual solutions out of the whole search space size is first identified and then initialized. An individual or chromosome solution I from $\mathbb{I}$ is defined as a complete solution being encoded with a finite set of n genes. It is worth to mention here that one should distinguish between the term "gene" used as the smallest sub-solution of a chromosome solution in EA terminology and the term "gene" used as a set of GO terms to semantically represent a protein or gene product. Each gene in I is simply the smallest sub-solution from the solution I and is defined by its location or index (usually known as locus) and its content or value (usually known as allele). Thus, for complex detection problem, an individual solution $I_{i|\ 1\leq i\leq pop-size}$ is formulated as $I_i = \{I_{i,1}, I_{i,2}, \dots, I_{i,n}\}$, where each sub-solution (i.e., gene) $I_{i,j}$ is expressed as protein to protein complex-neighbor sub-solution. $I_{i,j}$ is defined by its locus (protein P_i) and its allele (protein P_j) where P_i and P_j should have an interaction (i.e. $a_{i,j} = 1$). This will result in locating proteins P_i and P_j within the same complex C_k.

The decoding function δ of individual I will determine the number of the complexes and outline their structure, i.e. $\delta(I)$: $\mathcal{C} = \{C\}_{i=1}^{K}$. By its nature, the locus-based representation can automatically determine the number of complexes, K, being encoded in each individual I. Consider locus i is assigned with allele j. This means that protein i and protein j will be in the same complex C. Then, decoding an individual solution I will figure out one complete solution, $\mathcal{C} = \{C_1, C_2, \dots, C_K\}$, of a set of K complexes from the search space. Recall that the number of complexes, K, can differ from one solution $\mathcal{C}_i$ for chromosome I_i to another solution $\mathcal{C}_j$ for

chromosome I_j, where $1 \leq i, j \leq pop - size$. As an illustrative example, consider Figure 2, where a PPI network of 990 proteins and 4687 different interactions is encoded into four different chromosome solutions. These solutions were decoded into their corresponding phenotype solutions with different number of complexes, K. Further, two complexes in each solution are also enlarged in the figure to clarify the intra- and inter-connections. One protein (#79) with its intra- and inter-connections in two candidate solutions is also enlarged in the figure.

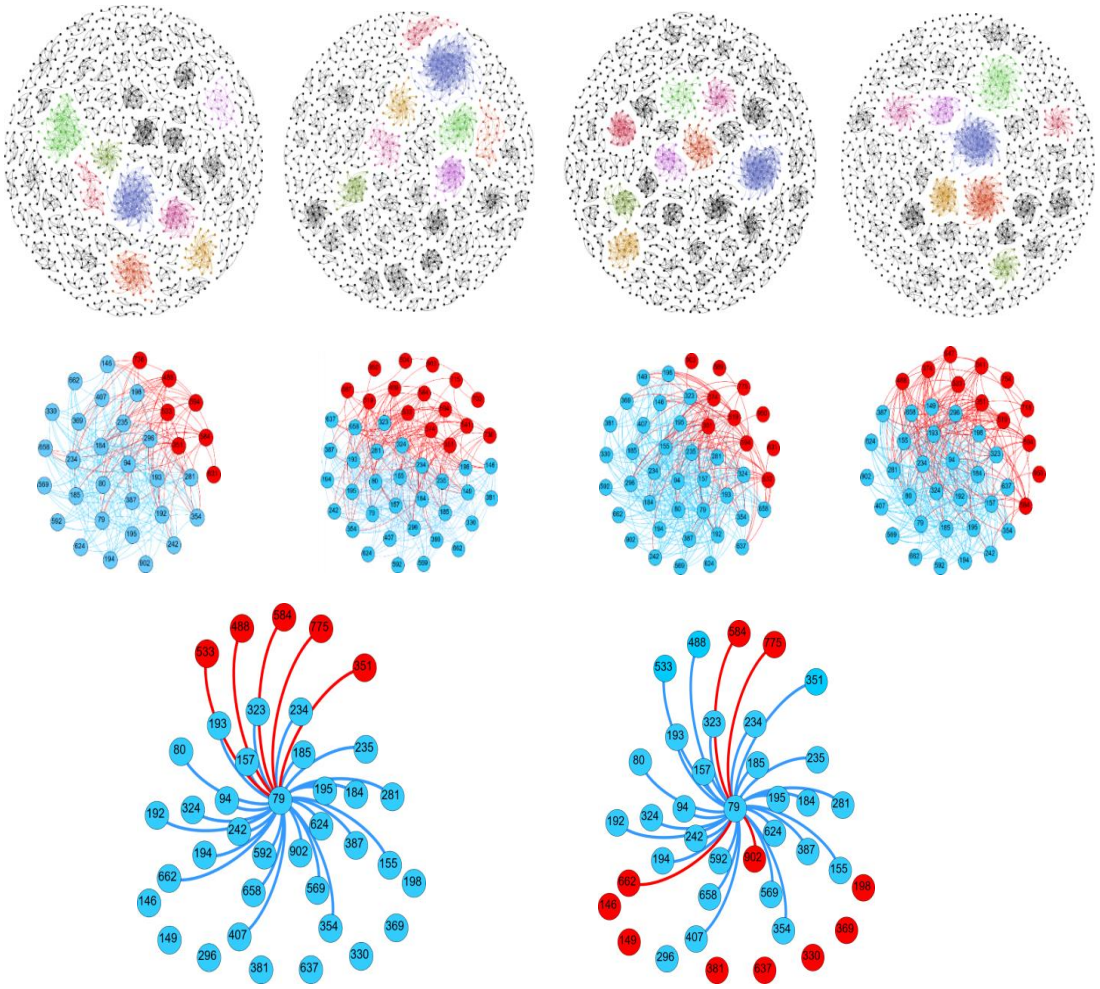

Figure 2: An illustrative example clarifying the partitioning of Yeast PPI network with 990. 1) Top: the phenotypic representation of four different chromosome solutions in a form of network partitions with varying number of complexes (K). Middle: the enlargement of two complexes with blue and red colors. Bottom: the enlargement of the intra-connections (blue) and inter-connections (red) of one protein (protein number 79)

For a PPI network $\mathcal{N}$ of n proteins and m interactions, first, the population $\mathbb{I}$ is initialized randomly, such that in each locus and in each chromosome (i.e., $I_{i,j|\ 1 \leq i \leq pop-size\ \wedge 1 \leq j \leq n}$), the allele is randomly initialized, such that P_i has an actual interaction with protein P_j.

Once the population is created, their individuals are evaluated according to the complex detection problem. The general characteristic a complex structure follows a complex community or module. Newman-Girvan modularity (Q) [20] for a candidate complex solution $\mathcal{C}$ with K complexes is defined as:

$$Q(\mathcal{C}) = \sum_{k=1}^{K} \left[\frac{\overline{m}(C_k)}{m} - \left(\frac{\sum_{P_i \in C_k} m_i}{2m} \right)^2 \right] \quad (1)$$

where m (as defined previously) is the total number of interactions in $\mathcal{N}$, $\overline{m}(C_k)$ is the number of intra-connection within complex C_k, and m_i is the number of interactions (degree) of protein P_i. Thus, the main characteristic of Q is its implicit definition for a single intra-complex score rather than a single intra- and inter-complex score.

Then, a set of good set of parent solutions is selected using binary tournament selection and used to evolve by perturbation (i.e., crossover, $\Psi_\times$, and mutation, Ψ_m, operators) further solutions to create better child individuals. The combined sequence of evaluation, selection, crossover, and mutation is then applied for a maximum number of generations, gen_{max}, and the best individual solution I_{best} (with its decoded complex structure $\mathcal{C}_{best}$) reached in gen_{max} is finally adopted as the required solution to the problem.

The canonical definition of uniform crossover, $\Psi_\times$, can be expressed as follows: Consider two chromosomes $I_1: (I_{1,1}, I_{1,2}, \dots, I_{1,n})$ and $I_2: (I_{2,1}, I_{2,2}, \dots, I_{2,n})$ to be the two parents participating in the crossover. With a specified crossover probability, $p_\times$, a child $I': (I'_1, I'_2, \dots, I'_n)$ can be generated from the two parents by mixing their alleles, uniformly (i.e. with equal chance), at each gene. This can be formally defined by: $\Psi_\times: (I_1, I_2, p_\times) \rightarrow I'$
$\forall i | \ 1 \leq i \leq n$:

$$I'_i = \begin{cases} I_{1,i} & if\ r \leq 0.5 \\ I_{2,i} & otherwise \end{cases} \quad (2)$$

where $r \sim [0,1]$ is a uniform random number. Generally, crossover probability, $p_\times$, is set high, e.g., $p_\times = 0.6$.

For mutation operator, Ψ_m, a small variation could occur to the generated child I_i after crossover. The canonical mutation operator imitates the traditional allele-aware mutation operator to change, with typically a small mutation probability, p_m, the allele value of a selected locus $I_{i,j}$ to another neighborhood protein. Again, the new allele value should represent one of the proteins that have interactions with protein P_j This can formally be specified as:
$\Psi_m: (I_{i| \ 1 \leq i \leq pop-size}, p_m) \rightarrow I_i{}'$
$\forall j | 1 \leq j \leq n \ \wedge r \leq p_m$:
$I'_{i,j} = j' | (i, j') \in \mathbb{E}$
(3)
where $r \sim [0,1]$ is a uniform random number.

3.2 An EA with the proposed GO-aware crossover operator

Designing an EA with appropriate operators that are tailored specially for complex detection problem is essential and can harness performance of the algorithm. Unfortunately, little effort is found in literature for designing EAs with GO-aware operators. The canonical crossover operator formulated in Eq. 2 is used in almost all EA-based complex detection algorithms. Actually, it is a variation operator working on the genotype representation, completely overpassing the semantic code. It uniformly inherits the topological information from two individual parents. However, to let this uniform crossover to respect the semantic and, thus, the functional information of the encoded parents, one can re-define the uniform crossover as a GO-based crossover, $\Psi_{GO-\times}$, as follows:
$\Psi_{GO-\times}: (I_1, I_2, p_\times) \rightarrow I'$

$\forall i | \ 1 \leq i \leq n$:

$$I_i' = \begin{cases} I_{1,i} & if\ FS_{P_i,I_{1,i}} > FS_{P_i,I_{2,i}} \\ I_{2,i} & otherwise \end{cases} \quad (4)$$

where FS is the functional similarity between two proteins.

In the proposed GO-aware crossover operator, two types for finding FS are used. These are:
1- Direct annotation scheme, which directly annotates each protein based on its direct GO terms.
2- Direct and inherited annotation scheme, which annotates each protein based on its direct GO terms and their ancestors in the DAGs.
For the first type of annotation, Jaccard similarity (Eq. 5) and Dice similarity (Eq. 6) are used to compute the functional similarity between two sets of GO terms, T_i and T_j, of, respectively, two proteins, P_i and P_j [16].

$$FS_{P_i,P_j} = \frac{|T_i \cap T_j|}{|T_i \cup T_j|} \quad (5)$$

$$FS_{P_i,P_j} = \frac{2 \times |T_i \cap T_j|}{|T_i| + |T_j|} \quad (6)$$

The second type, however, Wang et al. [21] method is used to compute semantic similarity between two GO terms. A GO term t is assigned with a semantic value $S(t)$ from the aggregation of the semantic contribution, $SC(DAG(t))$ of all its ancestors passing their best weighted paths to t. The best weighted path of each ancestor is the path with the maximum product of the weights on its edges (they set 0.8 and 0.6 for 'is a' and 'part of', respectively). Then, the semantic similarity ($SS_{t1,t2}$) between two GO terms, $t_1$and t_2, is defined as the ratio of the semantic contributions of all common terms (i.e. intersecting terms) in the DAGs of $t_1$and t_2 to the semantic values of $t_1$and t_2. For the functional similarity, on the other hand, four different types are adopted to calculate FS. These are: best match average similarity (BMA), average of best match similarity (ABM), maximum similarity (Max), and average inter-set similarity (Avg) in Eq. 7 – Eq. 10 [16]. The overall component of the proposed EA with the proposed heuristic model is then presented in Algorithm 1.

$$FS_{P_i,P_j} = \frac{max_{\forall t1 \in T_i} \sum_{\forall t2 \in T_j} SS_{t1,t2} + max_{\forall t2 \in T_j} \sum_{\forall ti \in T_i} SS_{t1,t2}}{|T_i| + |T_j|} \quad (7)$$

$$FS_{P_i,P_j} = \frac{\sum_{\forall t1 \in T_i} \max\left(SS_{t1,t2|t2 \in T_j}\right) + \sum_{\forall t2 \in T_j} \max\left(SS_{t1,t2|t1 \in T_i}\right)}{|T_i| + |T_j|} \quad (8)$$

$$FS_{P_i,P_j} = \max\left(SS_{t1,t2}\right) | t1 \in T_i, t2 \in T_j \quad (9)$$

$$FS_{P_i,P_j} = \frac{\sum_{\forall t1 \in T_i, t2 \in T_j} SS_{t1,t2}}{|T_i| \times |T_j|} \quad (10)$$

Algorithm 1: EA with GO-aware crossover operator

Input: 1) PPI network: $\mathcal{N}(n, E)$, GO database
2) population size: $pop - size$ and maximum number of generations gen_{max}

3) EA operators and their probabilities: s, $\Psi_{\times}$, Ψ_m, $p_{\times}$, p_m

Output: Best individual solution I_{best}

initialize $\mathbb{I} \leftarrow \{I_1, I_2, \dots, I_{pop-size}\}$;
$gen \leftarrow 0$;
evaluate Q for each individual in the population $\{Q_1, Q_2, \dots, Q_{pop-size}\}$;
while $(gen \leq gen_{max})$ **do**
  for $i \leftarrow 1$ to $pop - size$ **do**
    $I_{i,1}(t) \leftarrow select(\mathbb{I}_i(gen))$; // select parent 1 for I_i
    $I_{i,2}(t) \leftarrow select(\mathbb{I}_i(gen))$; // select parent 2 for I_i
    $I_i(gen) \leftarrow \Psi_{GO\times}(I_{i,1}(gen), I_{i,2}(gen), p_{\times})$;
    $I_i(gen) \leftarrow \Psi_m(I_i(gen), p_m)$;
    evaluate $Q(I_i(gen))$;
  end for
  $gen \leftarrow gen + 1$;
end while
return $I_{best} \in \mathbb{I}(gen)$ with maximum Q;

4. Results and discussions

A yeast Saccharomyces cerevisiae PPI network is used in the performance evaluation. The filtered version of this network contains $m = 4687$ interactions for $n = 990$ proteins. The GO terms assigned to the proteins are taken from the Saccharomyces Genome Database (SGD). The 990 proteins are annotated with GO terms for a total of 1245 BP, 452 CC, and 541 MF. To validate the quality of the predicted complexes, a benchmark gold standard complex set drawn from the Munich Information Center for Protein Sequence (MIPS) catalog is used in the experiments. This benchmark standard complex set contains 859 proteins partitioned into 81 protein complexes. The GO terms assigned to the proteins and their DAGs were downloaded from the Saccharomyces Genome Database (SGD) in URL: http://genome-www.stanford.edu/Saccharomyces/ in the period June 2021 - April 2022. The common measures of recall, precision, and cumulative F measure at both complex and protein levels are used in the evaluation. Figures 2 and 3 clarify the performance of the tested algorithms for average of 30 different runs. Note that Jaccard similarity score between a benchmark complex $C*$ and a predicted complex C (which is defined as the ratio of mutual proteins in $C*$ and C to the size of the set that contains all proteins of $C*$ and C) should be at least equals an overlapping score $OS(C, C*) = \frac{|C \cap C*|^2}{|C| \cdot |C*|}$.

Then C is said to be a true predicted complex. In other words, OS scores how effectively a predicted complex matched a complex from the benchmark set of complexes. In the experiments, OS is set from 0.1 to 0.8 in increment steps of 0.05.

For the performance comparison of EAs with the proposed GO-aware crossover operators, the results depicted in Figures 3 and 4 were obtained. The EAs with the

two variants of the proposed GO-aware crossover operators outperform the canonical EA (with the simple topological-based crossover operator) in almost all evaluation metrics and at both complex level and protein level due to the additional improvement capabilities introduced by the functional information. The hybridization capabilities of the GO-based crossover operator and other topological or canonical based operators, in the results, is proved to sufficiently locate more accurate complex partitions, in terms of functional structures, from the search space and to regulate the average size of complexes under more desirable functional structures.

Although, EA with the canonical operators seemingly better in terms of Precision (Figure 3) at the complex level, this behavior is not completely valid. To be more precise, the complex needs to be examined at a closer level, i.e., at the protein level. Then, the results in Figure 4 reveal that the introduction of GO information has no difficulties to achieve high values in all evaluation metrics at the protein level.

Further, the results in Table 1 report additional results which interpret the benefit that can come from designing GO-aware EA operators. The results in Table 1 compare the performance of the proposed GO-aware EA against other state-of-the-art non EAs, EAs with topological-aware operators proposed in [9], and a recent EA with different single-objective models and GO-based mutation operator proposed in [11]. Here, all EA-based methods are set to the following, more or less, commonly used settings. Population size, $pop - size$, is set to 100. The maximum number of generations used to stop the evolutionary process is set to 100 (i.e. 10000 function evaluations). Control parameters for the main evolutionary operators are set to the following: the probability of the uniform crossover, pc = 0.8, the probability of the mutation operator, pm = 0.2, and the probability of the GO-based mutation operator is also set to pm = 0.2. Further, for the proposed GO-aware EA, the results are reported using Jaccard semantic similarity. The first and second best results in each evaluation metric are indicated with bold.

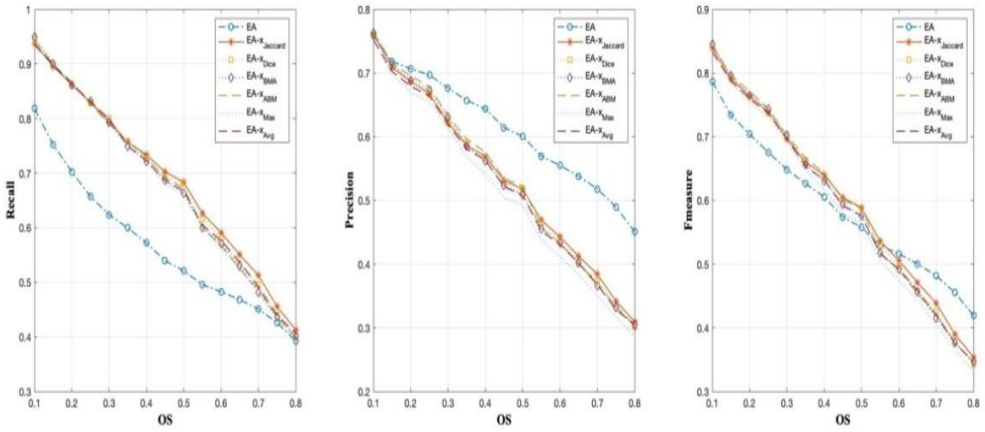

Figure 3: Performance comparison in terms of complex level Recall (left), Precision (middle), and F measure (right) for average of 30 different runs of canonical EA with topology aware crossover (denoted as EA) against the proposed EAs with GO aware crossover operators: with Jaccard similarity (EA-$X_{Jaccard}$), with Dice similarity (EA-X_{Dice}), with best match average similarity (EA-X_{BMA}), with average of best match similarity (EA-X_{ABM}), with maximum similarity (EA-X_{Max}), and with average similarity (EA-X_{Avg}).

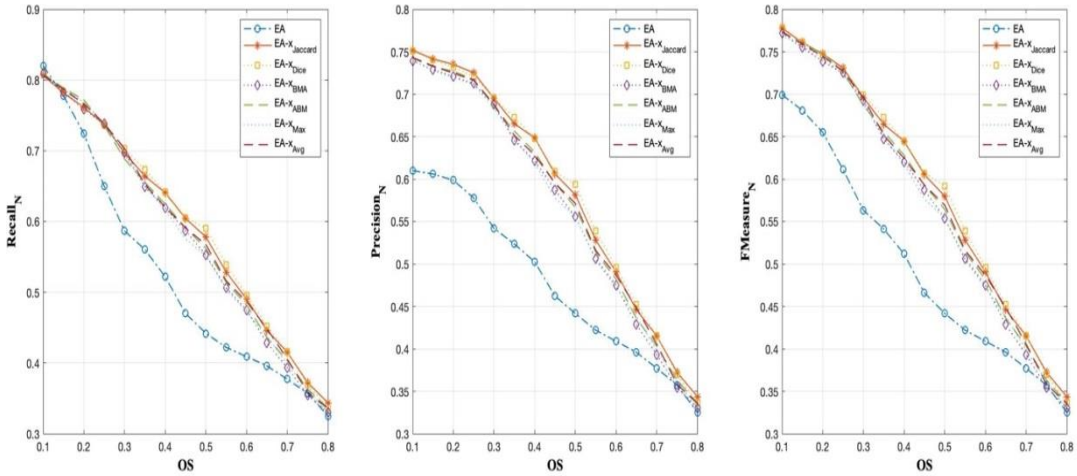

Figure 4: Performance comparison in terms of protein level Recall (left), Precision (middle), and F measure (right) for average of 30 different runs of canonical EA with topology aware crossover (denoted as EA) against the proposed EAs with GO aware crossover operators: with Jaccard similarity (EA-$X_{Jaccard}$), with Dice similarity (EA-X_{Dice}), with best match average similarity (EA-X_{BMA}), with average of best match similarity (EA-X_{ABM}), with maximum similarity (EA-X_{Max}), and with average similarity (EA-X_{Avg}).

The results reflect that the investment in the functional domain reinforces the proposed EA with GO-aware crossover to outperform the counterpart algorithms.

In other words, this indicates the implicit ability of the proposed GO-based crossover operator to maintain an adequate partitioning of the proteins into more accurate complex structures. More interestingly, the proposed GO-based EA with modularity-based model outperforms (in terms of Recall and F measure) the counterpart GO-based EAs proposed in [11], even, with other single-objective models.

Table 1: Comparison of performance in terms of complex level Recall, Precision, and F measure with overlapping score equals to 0.2. Here, 1st group: some well-known non EAs, 2nd group: topological-based based EAs proposed in [9], 3rd group: GO-based EAs proposed in [11], and the proposed EA with GO-based crossover operator.

Algorithm		Metric			
Class	**name**	**Year [Ref.]**	***recall***	***precision***	***F Measure***
Non EA	**RNSC**	2004 [22]	**0.8490**	0.2650	0.4039
	MCODE	2003 [23]	0.6700	0.6250	0.6467
	MCL	2002 [24]	0.8230	0.5390	0.6514
	LC	2010 [25]	0.4950	0.0410	0.0757
	OCG	2012 [26]	0.8380	0.6150	0.7094
	RanCoC	2012 [27]	0.0000	0.0000	0.0000
	CPM	2005 [28]	0.5850	0.6170	0.6006
	ELC	2013 [29]	0.5910	0.6479	0.6181
	NDOCD	2016 [30]	0.7830	0.7000	0.7392
Topological-based based EA	**EA-Q**	2014 [8]	0.8103	0.6453	0.7181
	EA-CO		0.7462	0.6793	0.7108
	EA-EX		0.8474	0.6073	0.7073
	EA-CR		0.7538	0.6742	0.7110
	EA-NC		0.7513	0.6839	0.7157
	EA-ID		0.8282	0.6139	0.7048
	EA-CS		0.8564	0.6196	0.7188
GO-based EA	**EA-Q**	2020 [11]	0.7910	0.7473	0.7682
	EA-CO		0.7923	**0.7583**	0.7744
	EA-EX		0.7577	0.7567	0.7569
	EA-CR		0.7821	0.7567	0.7687
	EA-NC		0.7885	**0.7643**	**0.7760**
	EA-ID		0.8256	0.6940	0.7533
EA with GO-aware crossover	**EA-Q_x**		**0.8610**	0.6890	**0.8650**

5. Conclusions

Although the need for GO heuristic operators to improve the performance of evolutionary-based complex detection algorithms is unquestionable, this trend of researching is seldom examined in the literature. In this paper, an EA with a GO-

aware crossover operator is proposed in an attempt reach more reliable results. From the results, future research can be recommended while underscoring new design methodology. These new research directions would help to fill up the relatively empty map between the biological domain and the topological domain for protein complexes. For example, semantic similarity measures can be studied and applied to other biological ontologies, such as Sequence Ontology, Microarray and Gene Expression Data Ontology. For example, the availability of time course gene expression profile enables us to uncover the dynamics of molecular networks and improve the detection of protein complexes.

References

[1] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner, "On modularity clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 2, pp. 172-188, 2008.

[2] S. Schaeffer , "Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27–64, 2007.

[3] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 026113, 2004.

[4] H. R. Lewis, R. M. ΠGarey, D.S. Johnson , "*Computers and intractability. a guide to the theory of np-completeness*,", W. H. Freeman and Company, San Francisco1979, x+ 338 pp. The Journal of Symbolic Logic. Jun 1983.

[5] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti- Spaccamela, and M. Protasi, "*Complexity and approximation: Combinatorial optimization problems and their approximability properties*," Springer Science & Business Media, 2012.

[6] E.-G. Talbi, " Metaheuristics: from design to implementation," John Wiley & Sons, 2009, vol. 74.

[7] 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 of Things Journal*, vol.6, no. 4, pp. 7225-7231, 2019.

[8] 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, 2014.

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

[10] 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," *Applied Soft Computing*, vol. 73, pp. 1004–1025, 2018.

[11] D. A. Abduljabbar, S. Z. M. Hashim, R. Sallehuddin, "An enhanced evolutionary algorithm for detecting complexes in protein interaction networks with heuristic biological operator," in: *International Conference on Soft Computing and Data Mining*, Springer. pp. 334–345, 2020[Online], DOI: 10.1007/978-3-030-36056-6_32.

[12] G. O. Consortium, "The gene ontology in 2010: extensions and refinements," *Nucleic acids research*, vol. 38, no. suppl 1, pp. D331–D335, 2010.

[13] D. Barrell, E. Dimmer, R. P. Huntley, D. Binns, C. O'Donovan, and R. Apweiler, "The goa database in 2009—an integrated gene ontology annotation resource," *Nucleic acids research*, vol. 37, no. suppl 1, pp. D396–D403, 2009.

[14] E. Camon, M. Magrane, D. Barrell, V. Lee, E. Dimmer, J. Maslen, D. Binns, N. Harte, R. Lopez, and R. Apweiler, "The gene ontology annotation (goa) database: sharing knowledge in uniprot with gene ontology," *Nucleic acids research*, vol. 32, no. suppl 1, pp. D262–D266, 2004.

[15] P. V. Ogren, K. B. Cohen, G. K. Acquaah-Mensah, J. Eberlein, and L. Hunter, "The compositional structure of gene ontology terms," in *Biocomputing*, pp. 214–225, 2003[Online], https://doi.org/10.1142%2F9789812704856_0021.

[16] C. Pesquita, "*Semantic similarity in the gene ontology*," in The gene ontology handbook. Humana Press, New York, NY, 2017, pp. 161–173.

[17] P. Mihail, J.M. Keller, J.A. Mitchell, "Fuzzy measures on the gene ontology for gene product similarity," *IEEE/ACM Transactions on computational biology and bioinformatics* ,vol.3, no.3 pp.263–274, 2006.

[18] P. H. Guzzi, M. Mina, C. Guerra, and M. Cannataro, "Semantic similarity analysis of protein data: assessment with biological features and issues," *Briefings in bioinformatics*, vol. 13, no. 5, pp. 569–585, 2012.

[19] J. Handl, J. Knowles, "An Evolutionary Approach to Multiobjective Clustering*," Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 1, pp. 56–76, 2007.

[20] M. E. Newman, M. Girvan, "Finding and Evaluating Community Structure in Networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[21] J. Z. Wang, Z. Du, R. Payattakool, P. S. Yu, C.-F. Chen, "A new method to measure the semantic similarity of GO terms," *Bioinformatics*, vol. 23, no. 10, pp. 1274–1281, 2007.

[22] A. D. King, N. Pržulj, I. Jurisica, "Protein complex prediction via cost-based clustering," *Bioinformatics*, 20(17), 3013–3020, (2004).

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

[24] S. Ray, A. Hossain, U. Maulik, "Disease associated protein complex detection: A multi-objective evolutionary approach," *In Microelectronics, Computing and Communications (MicroCom), International Conference on IEEE*, January- 2016. (pp. 1-6).

[25] Y.Y. Ahn, J. P. Bagrow, S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol.466, no.7307, pp.761–764, 2010.

[26] E. Becker, B. Robisson, C.E. Chapple, A. Guénoche, C. Brun, "Multifunctional proteins revealed by overlapping clustering in protein interaction network," *Bioinformatics*, vol.28, no. 1, pp. 84–90, 2012.

[27] C. Pizzuti, S.E. Rombo, "A coclustering approach for mining large protein-protein interaction networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol.9, no.3, pp. 717-730, 2012.

[28] G. Palla, I. Derényi, I. Farkas, T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol.435, no.7043, pp. 814-818, 2005.

[29] L. Huang, G. Wang, Y. Wang, E. Blanzieri, C. Su, "Link clustering with extended link similarity and EQ evaluation division," *Plos One* vol.8, no.6, 2013.

[30] Z. Ding, X. Zhang, D. Sun, B. Luo, "Overlapping Community Detection based on Network Decomposition," *Scientific Reports*, vol.6, no.1, pp24115, 2016.