A Meta-Heuristic Approach to Predict Protein-Protein Interaction Network

Archana Chowdhury[1], Pratyusha Rakshit[2], Amit Konar[3]
Department of Electronics & Telecommunication Engineering
Jadavpur University
Kolkata, India
[1]chowdhuryarchana@gmail.com,[2]pratyushar1@gmail.com,
[3]konaramit@yahoo.co.in

Atulya K. Nagar
Department of Math and Computer Science,
Liverpool Hope University,
Liverpool, U.K
nagara@hope.ac.uk

Abstract— **Protein interactions are central to structural and functional organization of the cell. Understanding biological processes thus relies on a comprehensive knowledge of different types of protein–protein interactions (PPIs) and interaction mechanisms. This paper formulates the PPI prediction problem as a multi-objective optimization problem. The focus here is to jointly maximize i) the number of common neighbors of the proteins predicted to be interacting, ii) their functional similarity, and iii) the ratio between their individual accessible solvent area and that of the corresponding protein-protein complex. The above multi-objective optimization problem is solved using a fusion of the differential evolution for multi-objective optimization and the stochastic learning automata. Here the former is employed to globally explore the search space and the latter for the adaptive tuning of the control parameters of the algorithm. Experiments undertaken reveal that the proposed PPI prediction technique outperforms existing methods with respect to sensitivity, specificity, and F1 score.**

Keywords—***protein–protein interaction networks; annotation; accessible solvent area; stochastic learning automata; differential evolution for multi-objective optimization.***

I. INTRODUCTION

Proteins are involved in many essential processes within the cell such as metabolism, cell structure, immune response, and cell signaling [1]. Proteins rarely act in isolation; rather they function in the crowded medium of other molecules and proteins. Proteins interact with other proteins to form protein–protein interactions (PPI). PPIs are of interest in biology because they regulate roughly all cellular processes, including metabolic cycles, DNA transcription and replication, different signaling cascades and many additional processes. Collections of interactions among proteins form a complex interaction network in the cell. The more we know about proteins, the more we see them as parts of complex networks or pathways rather than isolated entities. The function of a protein can be viewed by its position within this cellular interaction network [2].

The importance of understanding the PPI interactions has prompted the development of various experimental methods used in measuring them**.** High-throughput functional genomics approaches are needed to bridge the gap between the raw sequence information and the relevant biochemical and medical information. Computational methods are required for discovering interactions that are not accessible to high throughput methods. A number of computational approaches for protein interaction discovery have been developed over recent years [3]. These methods differ in feature information used for protein interaction prediction. Many studies have demonstrated that knowing the tools and being familiar with the databases is important for new research in PPI analysis to be conducted [4].

In this paper, the prediction of possible interaction between two proteins is based on maximizing three scoring functions based on: 1) the number of common neighbor shared by the two proteins predicted to be interacting, 2) similarity of their functions, and 3) the ratio between their individual accessible solvent area and that of the corresponding protein-protein complex. The first criterion of the objective function is based on the topological feature of the proteins. The PPI network is characterized by several topological properties [5]. One of the topological features is the number of common neighbors shared by the proteins in the PPI network. It has been observed that if two proteins share a large number of common neighbors in the network then the two proteins are predicted to be interacting [6]. The second criterion of the objective function is based on *gene ontology* (GO) annotation of proteins. GO annotation has been identified as one of the strongest predictors for protein interactions [7]. The final criterion is based on *accessible solvent area* (ASA) of protein. The important properties of macromolecules like proteins, DNA and RNA are related to their interaction with the surrounding water molecules as they are designed by the natural evolutionary process to function in the environment of water solutions. Since the determination of the protein-solvent interaction is very difficult, several approaches have been proposed to replace these interactions with approximated potentials. One such approximation is solvation energy. The solvation energy of the atoms or atomic groups is proportional to the atomic surface exposed to the solvent [8], which is calculated using ASA. Sharing common neighbors between proteins may not always confirm the desired structural properties of the interacting proteins (suitable for formation of protein-protein complex). Similarly, different proteins may be found to possess a large number of shared neighbors but with rare functional similarity required to validate the real world interaction. Thus, it can be concluded that these three properties are mutually independent and hence need to be optimized simultaneously and individually to validate the predicted PPIs. This justifies the formulation of the PPI prediction problem in a multi-objective optimization framework.

The PPI problem refers to determining the possible interacting protein pairs and is in general a combinatorial optimization problem, which is usually *NP* hard [9]. To obtain approximate solutions to the problem in a finite time for *NP* hard problems, usually heuristics are used. In this paper, a novel meta-heuristic algorithm is proposed that jointly performs the global exploration and the local refinement in the optimization process, by using the differential evolution for multi-objective optimization (DEMO) [10] and the stochastic learning automata (SLA) [11].

Exploration and exploitation are used to keep track of the efficiency of an evolutionary multi-objective optimization algorithm in locating the global optima. In reality, the control parameter (scale factor) of DEMO needs to be adaptively tuned during the search process to effectively counter balance the exploration and exploitation capabilities, which in turn helps in balancing the run-time complexity and the computational accuracy. SLA is used here to achieve it.

This paper is an approach to improve the work proposed in [12] significantly. In [12], the authors used firefly algorithm with non-dominated sorting (FANS), whereas the present version examines the scope of DEMO with SLA. The PPI network in this paper has been encoded as a one-dimensional vector of size one more than the possible number of interactions possible in a PPI network. The elements of the vector are the weights of the connection between the proteins and the last element of the vector contains the threshold value based on which the connection is established between the proteins. The inclusion of number of common neighbors of proteins and accessible solvent area in the objective formulation and the learning of the parameters of DEMO by using SLA is also done. All these above mentioned point results in significant improvement in performance as indicated by three useful metrics, namely *specificity, sensitivity,* and *F*1 *score*.

The rest of the paper is divided into five sections. Section II gives a brief idea about the formulation of the PPI prediction problem and explains the criteria used. In section III, DEMO-SLA algorithm is proposed. Section IV presents the experimental settings and the results. Section V concludes the paper.

II. Problem Formulation

This section attempts to formulate the PPI prediction as a multi-objective optimization problem. The characteristic features used to develop the objective functions are illustrated here. These objective functions on simultaneous maximization returns the desired network.

A. Formation of a Protein-Protein Interaction Network

A PPI network with *N* proteins may have a maximum of $N\times(N-1)/2$ interactions, if self-interactions are ignored. The observation has motivated us to represent the PPI network by a vector $\vec{Z}$ of dimension $1\times D$ where

$$D = N\times(N-1)/2+1 . \qquad (1)$$

The *m*-th element of $\vec{Z}$, denoted by $Z_m \in [0, 1]$, for *m*=[1, 2, …, *D*–1], symbolizes the predicted weight of interaction $w_{i,j}$ between proteins p_i and p_j, for *i*= [1, 2, …, *N*–1] and *j*= [*i*+1, *i*+2, …, *N*], such that

$$m = N\times(i-1) - \frac{i\times(i+1)}{2} + j . \qquad (2)$$

The *D*-th element, $Z_D \in [0, 1]$, denotes the threshold value, *Th*. The proteins p_i and p_j are predicted to be interacting if $w_{i,j} \geq Th$. An example of $\vec{Z}$ for representing the interaction weights of five proteins (*N*=5) in a PPI networkis given in Fig. 1. The first ten elements of $\vec{Z}$ are used to decode the weights of possible interaction between proteins. The eleventh component of $\vec{Z}$ denotes the threshold *Th*.

1	2	3	4	5	6	7	8	9	10	11
0.34	0.68	0.29	0.43	0.73	0.62	0.78	0.38	0.19	0.24	0.58

Fig. 1 Example of vector representation of a PPI network with 4 proteins

The weight of interaction $w_{i,j}$ between proteins p_i and p_j (for *i*= [1, 2, …, *N*–1] and *j*= [*i*+1, *i*+2, …, *N*]) can be identified by decoding $\vec{Z}$ using (2). For example, for proteins p_1 and p_3 (i.e., *i*=1 and *j*=3), the value of *m* is 2, as obtained from (2). It implies that the weight of interaction $w_{1,3}$ between proteins p_1 and p_3 can be decoded from the 2nd component of $\vec{Z}$, giving $w_{1,3}=Z_2=0.68$. The decoded vector now can be represented as follows.

1	2	3	4	5	6	7	8	9	10	11
0.34	0.68	0.29	0.43	0.73	0.62	0.78	0.38	0.19	0.24	0.58
$w_{1,2}$	$w_{1,3}$	$w_{1,4}$	$w_{1,5}$	$w_{2,3}$	$w_{2,4}$	$w_{2,5}$	$w_{3,4}$	$w_{3,5}$	$w_{4,5}$	*Th*

Fig. 2 Example of decoding vector representing a PPI network with 4 proteins

After decoding the weights, they are individually compared to $Z_D= Z_{11}=Th$ to identify the predicted non-interacting protein pairs.It is evident from Fig. 2 that proteins p_1 and p_3 are predicted to be interacting as $w_{1,3}(=0.68) > Th(=0.58)$, while entailing non-interaction between proteins p_3 and p_4 with $w_{3,4}(=0.38) < Th(=0.58)$. After decoding $\vec{Z}$ and interpreting the interaction weights, we obtain the following PPI network as shown in Fig. 3.

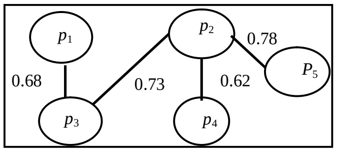


Fig. 3 Example of a PPI network with 5 proteins

B. Predicting Protein-Protein Interaction Using Neighborhood Topology

Interaction between a pair of proteins is possibly proportional to the size of their common neighborhood [13]. The size of the common neighborhood of protein pair p_i and p_j can be determined by identifying the number of proteins p_l in the network, which are directly interacting with both p_i and p_j. The weight of interaction $w_{i,j}$ between proteins p_i and p_j is captured by the number of proteins in their common neighborhood. Protein p_l is predicted to be interacting with p_i

and p_j if the corresponding weights of interaction $w_{i,l}$ and $w_{j,l}$ are both greater than the threshold value *Th*. Apparently,

$$w_{i,j} \propto |n_{i,j}| / N \quad (3)$$

where $n_{i,j}$ is the set of all proteins p_l in the network with $w_{i,l}$ (or $w_{l,i}$)>*Th* and $w_{j,l}$ (or $w_{l,j}$)>*Th*. The accuracy in predicting the interacting protein pairs p_i and p_j in a PPI network can be judged by measuring the similarity between the predicted weight of interaction $w_{i,j}$ and their common neighborhood ratio $|n_{i,j}|/N$. The above requirement can be accomplished by maximizing (4).

$$s_n(p_i, p_j) = 1 / [(w_{i,j} - |n_{i,j}| / N)^2 + \varepsilon] \quad (4)$$

Here ε is a small positive constant. By maximizing (5) we may accurately predict the interaction weights between proteins in a network.

$$J_1 = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} s_n(p_i, p_j) \quad (5)$$

C. Predicting Protein-Protein Interaction Using Functional Characteristics

Proteins that are located at the same cellular compartment and possess related molecular functions as well as are involved in similar biological processes, are considered as functionally similar. The similarity *sim*(*f*, *f* $^{/}$) between any two functions *f* and *f* $^{/}$ can be evaluated by a similarity score between the respective GO terms annotating functions *f* and *f* $^{/}$. Let *O* and *O*$^{/}$ signify the GO term annotating protein functions *f* and *f* $^{/}$ respectively. Hence,

$$sim(f, f') = sim(O, O') \quad (6)$$

There are evidences [14] where semantic similarity measure between two proteins is used to infer their possible interaction. The semantic similarity between two proteins is defined as the average similarity of all GO terms annotating the functions of two proteins. The similarity score between *O* and O$^{/}$ has been adopted from [14] and is given as follows.

$$sim(O,O') = \max_{g \in CA(O,O')} \left(\left(2 \times \log p_g - (\log p_O + \log p_{O'}) \right) \times \left(1 - p_g \right) \right) \quad (7)$$

where *CA*(*O*, O$^{/}$) represents the set of all common ancestor GO terms of *O* and O$^{/}$. Here p_t represents the probability of occurrence of GO term *t* in a specific corpus which is normally estimated by its frequency of annotation. Frequency of annotation is usually calculated from the number of child nodes a GO term has in the GO tree structure [26].

Maximization of the functional similarity between any two predicted interacting proteins p_i and p_j in the network, given as

$$J_2 = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \max\left(w_{i,j} \times s_f(p_i, p_j), \left(1 - w_{i,j}\right) \times \left(1 - s_f(p_i, p_j)\right) \right), \quad (8)$$

will yield a better and more accurate prediction of interacting protein partners. Expression (8) provides a high value of J_2 if both $w_{i,j}$ (the predicted interaction weight between two proteins p_i and p_j) and s_f (p_i, p_j) (their functional similarity) are comparable to each other. If p_i and p_j are predicted to be interacting with a high value of $w_{i,j}$ and if they truly have a high functional similarity it will make $w_{i,j} \times s_f(p_i, p_j)$ high. Similarly, if the protein pair is predicted to be non- interacting with a low value of $w_{i,j}$ and if they really have low functional similarity, it will make $[(1-w_{i,j}) \times (1- s_f(p_i, p_j)]$ high. These in turn maximize J_2. A wrong prediction of $w_{i,j}$ will reduce the value of J_2.

D. Predicting Protein-Protein Interactions Using Accessible Solvent Area

The concept of ASA was introduced by Lee and Richards to quantitatively describe the extent to which atoms on the protein surface can form contacts with water [15]. ASA is identified by probing the van der Waals surface of a protein molecule with a probe atom. The probe atom is generally taken as the solvent molecule or water molecule. If the maximum van der Waals radius of any atom in the molecule is *Rv* and the radius of the probe atom is *Rp*, then for any given atom say *X*, only atoms with centers within 2(*Rv*+ *Rp*) of *X* can be involved in creating ASA. ASA is defined as the locus of the center of probe sphere when it rolls on the van der Waals surface of the molecule without penetrating any atom. Fig. 4 represents the pictorial view of three protein atoms being probed by a probe molecule. The locus of the center of the probe molecule, marked as blue dashed line, represents the accessible solvent area.

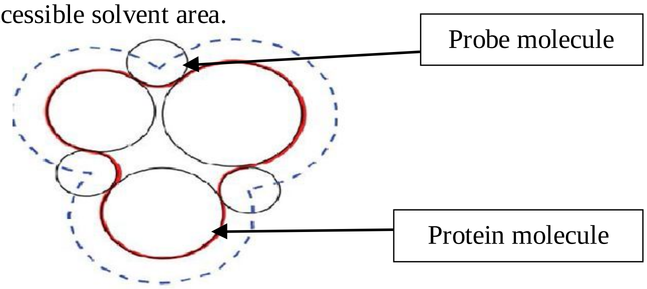


Fig. 4. Pictorial representation of accessible surface area

The reduction of unfavorable interactions in PPI occurring between water and non-polar atoms is termed as hydrophobic effect. The hydrophobic residues in proteins are incapable of forming hydrogen bonds in the aqueous solution. In general, the binding of two proteins is associated with an increase in entropy ($T\Delta S$) and a decrease in enthalpy (ΔH) of the protein-protein complex relative to individual proteins. The combined effect of such changes in the free energy given by ΔG can be represented as

$$\Delta G = \Delta H - T\Delta S \quad (9)$$

which is the displacement of water molecules from the hydrophobic surfaces of interacting proteins. These water molecules get released to the surrounding aqueous environment, where they form hydrogen bonds with other water molecules.

The binding sites of two interacting proteins must be desolvated upon binding. When two proteins are bound to each other, the side chain or the main chain non-polar functional groups, present in their binding sites, become partially or completely restrained and construct intermolecular interaction. These non-polar molecules (functional groups) stay together to minimize water-exposed ASA. The hydrophobic effect is entropy driven process in PPI, which seeks to decrease the ΔG of the protein-protein complex by minimizing the surface interface between hydrophobic residues and water. Hence, in PPI, the extent of penetration of two proteins in their binding sites can be evaluated by assessing the ASA of the protein-protein complex.

Considering the above requirements, the strength of interaction between proteins p_i and p_j based on their ASA reduction profile upon binding is given by

$$s_d(p_i, p_j) = \frac{ASA(p_i) + ASA(p_j)}{ASA(p_{i\_j})} \tag{10}$$

where $ASA(p_i)$ and $ASA(p_{i\_j})$ respectively denote the ASAs of protein p_i and the complex formed by interaction between proteins p_i and p_j. It is evident, more the extension in reduction of the ASA of the complex ($ASA(p_{i\_j})$) with respect to their individual ASAs, more is the strength of the possible binding between proteins p_i and p_j.

The maximization of the similarity between two predicted interacting proteins based on their reduced ASA after binding, given as

$$J_3 = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} w_{i,j} \times s_d(p_i, p_j) \tag{11}$$

is expected to capture the predicted interacting protein pairs. As in case of J_2, here also prediction of high (or low) interacting weights $w_{i,j}$ for proteins p_i and p_j with more (or less) reduction in the ASA of the complex after their binding with respect to their individual ASAs, ensures a high value of J_3.

III. Algorithm

The objective functions J_1 to J_3 in section II usually take multi-modal objective surfaces with local discontinuities at finite points in the search space of PPI network. Traditional calculus based approach thus is inconvenient to optimize the above-mentioned individual objective functions. We think of using meta-heuristic algorithms to optimize individual objectives. Although there exist plenty of algorithms to address the optimization problem, very few of them can capture the global optima as the objective surfaces have multiple rough local optima [16]. We here propose a solution by using a meta-heuristic algorithm to serve as the optimization task, while simultaneously learning the objective surfaces in the search space of PPI network. Hence, we plan to adapt the parameters of the meta-heuristic search algorithm according to the objective function estimates of a candidate solution (i.e., we learn the objective surface at the location of the trial solution). The knowledge of the parameter selection for each candidate solution in the optimization algorithm is obtained from the previously acquired knowledge about the parametric choice. The differential evolution for multi-objective optimization (DEMO) [10] here has been used as the basic optimization algorithm and the stochastic learning automata (SLA) [11] as the parameter (scale factor of DEMO)-tuning algorithm. The synergistic effect of both improves the solutions for the given PPI prediction problem.

A. Differential Evolution for Multi-objective Optimization

An overview of the main steps of the traditional differential evolution for multi-objective optimization (DEMO) algorithm for simultaneous maximization of *K* objectives is presented next.

(a) Initialization: DEMO commences from an initial population P_t of *NP*, *D*-dimensional target vectors (candidate solutions) $\vec{Z}_i(t) = \{z_{i,1}(t), z_{i,2}(t), ..., z_{i,D}(t)\}$ for *i*= [1, *NP*] at generation *t*=0, randomly initialized within the search bound $[\vec{Z}^{\min}, \vec{Z}^{\max}]$ where $\vec{Z}^{\min} = \{z_1^{\min}, z_2^{\min}, ..., z_D^{\min}\}$ and $\vec{Z}^{\max} = \{z_1^{\max}, z_2^{\max}, ..., z_D^{\max}\}$. The crossover rate *CR* is initialized in [0, 1]. The *k*-th objective function $J_k(\vec{Z}_i(0))$ is evaluated for the target vector $\vec{Z}_i(0)$ for *k*= [1, *K*] and *i*=[1, *NP*].

(b) Mutation: A donor vector $\vec{V}_i(t)$ is created corresponding to each target vector $\vec{Z}_i(t)$, for *i*=[1, *NP*], following DE/rand/1 mutation scheme

$$\vec{V}_i(t) = \vec{Z}_{r1}(t) + F \times (\vec{Z}_{r2}(t) - \vec{Z}_{r3}(t)) \tag{12}$$

where $\vec{Z}_{r1}(t), \vec{Z}_{r2}(t)$ and $\vec{Z}_{r3}(t)$ are randomly selected solutions from P_t such that $i \neq r1 \neq r2 \neq r3$. *F* symbolizes the scaling factor in [0, 2].

(c) Crossover: The binomial crossover is concerned with generating a trial vector $\vec{U}_i(t)$ for each pair of a donor vector $\vec{V}_i(t)$ and the respective target vector $\vec{Z}_i(t)$ by the following operation

$$u_{i,j}(t) = \begin{cases} v_{i,j}(t) \text{ if } \mathrm{rand}_{i,j} \le CR \text{ or } j = j_{rand} \\ z_{i,j}(t) \text{ otherwise} \end{cases} \tag{13}$$

for *j*=[1, *D*] where $j_{rand} \in [1, D]$ is a randomly chosen index. The *k*-th objective function $J_k(\vec{U}_i(t))$ is evaluated for the trial vector $\vec{U}_i(t)$ for *k*= [1, *K*] and *i*= [1, *NP*].

(d) Population Update: To ensure faster convergence to the true Pareto front while simultaneous maximization of *N* objectives, the following non-dominated sorting based selection strategy is adopted in DEMO. The trial vector $\vec{U}_i(t)$ replaces the corresponding target vector $\vec{Z}_i(t)$ if $\vec{U}_i(t)$ dominates $\vec{Z}_i(t)$. However, when $\vec{U}_i(t)$ and $\vec{Z}_i(t)$ are non-dominated, $\vec{U}_i(t)$ is included in the current population P_t. Otherwise, $\vec{U}_i(t)$ is discarded. Repeating this process for *i*=[1, *NP*] yields a population of solution vectors with size $|P_t| \in [$ *NP*, 2*NP*].

(e) Update of Next Generation Population using Non-dominated Sorting and Crowding Distance Metric: The resulting population P_t is then sorted into a number of Pareto fronts pf_1, pf_2, pf_3, and so on, according to non-domination [17]. The population P_{t+1} for the next generation is formed by identifying the non-dominated sets of solutions from P_t (of size in [*NP*, 2*NP*]) according to the ascending order of their

Pareto ranking starting from pf_1. The members of the front pf_l, which can be partially passed on to P_{t+1}, are sorted in descending order of crowding distance [17]. To ensure diversity in population, the solutions with the highest crowding distances are included in P_{t+1} until its size becomes *NP*.

(f) Convergence: After each evolution step, we repeat from step (b) until the terminating condition for convergence is satisfied.

B. Stochastic Learning Algorithm

Stochastic learning automata (SLA) fall under the class of reinforcement learning [11]. In reinforcement learning, the agent performs an action causing a state transition in the environment and receives a reward (or penalty) for the action in an attempt to reach a definite goal. The task of the agent here is to learn a control policy to select an action (from a set of possible actions) at a given state *s* to improve the probability of correct response because of interaction with its environment.

Let,

$S=\{s_1, s_2, \ldots, s_m\}$ be a set of *m* states of an agent in a given environment,

$A=\{a_1, a_2, \ldots, a_n\}$ be a set of *n* actions that the agent can select in each state $s_i \in S$,

$p_{i,j}$ be the action probability governing the choice of the action for transition to a new state s_k by executing action a_j at state s_i,

$x_{i,j}$ be the response the agent acquires (from the environment) by the execution of an action a_j at state s_i. According to P-model [11] of environment, $x_{i,j}=0$ represents non-penalty response or reward, while $x_{i,j}=1$ signifies the penalty response.

SLA initiates with equal probabilities for each action and hence considers no prior assumption about the optimal action. SLA begins with a randomly selected initial state $s_i \in S$. One action $a_j \in A$ is selected at random, the response of the environment $x_{i,j}$ is observed, based on which the action probabilities $p_{i,l}$, for $l=[1, n]$, at state s_i are changed. The agent moves to a new state s_k due to execution of action a_j. Now the next state s_k is considered as the initial state and a new action is selected according to the updated action probabilities and the procedure is repeated.

It is now evident that the most crucial factor that influences the achievement of the desired performance of SLA is the reinforcement-learning scheme for updating of action probabilities. The linear reinforcement-learning scheme $L_{R\text{-}P}$ for updating action probabilities is described below.

For a non-penalty/reward response $x_{i,j}=0$ obtained at state s_i by executing action a_j

$$\begin{aligned} p_{i,j} &\leftarrow p_{i,j} + \lambda \times (1-p_{i,j}) \\ p_{i,l} &\leftarrow p_{i,l} - \lambda \times p_{i,l}, \quad \text{for } l=[1,n], l \neq j \end{aligned} \tag{14}$$

Contrarily, if a penalty response $x_{i,j}=0$ is obtained at state s_i by executing action a_j

$$\begin{aligned} p_{i,j} &\leftarrow p_{i,j} - \delta \times (1-p_{i,j}) - \frac{\delta}{n} \times (n-1) + (n-1) \\ p_{i,l} &\leftarrow p_{i,l} + \delta \times p_{i,l} + \frac{\delta}{n} - 1, \quad \text{for } l=[1,n], l \neq j \end{aligned} \tag{15}$$

The parameter $\lambda \in [0, 1]$ is associated with reward response while the parameter $\delta \in [0,1]$ is associated with the penalty response. Here $L_{R\text{-}P}$ model is employed with $\lambda=\delta$.

C. Stochastic Learning Algorithm Induced DEMO

(a) Initialization: DEMO-SLA starts with a population of *NP*, *D*-dimensional target vectors representing the candidate solutions within the prescribed bound $[\vec{Z}^{\min}, \vec{Z}^{\max}]$ at t=0. The objective function values $J_k(\vec{Z}_i(0))$ are evaluated for $k=[1, K]$ and $i=[1, NP]$. The entries (action probabilities) for the state transition table are initialized with equal and small values, for example 0.1. This is in accordance with the principle of unavailability of a priori information about the behavior of the environment and assuming all selection of actions (scale factor) to be equally likely at a particular stage.

(b) Ranking of Members and State Assignment: The target vectors $\vec{Z}_i(0)$, for $i=[1, NP]$, are then assigned states in the state transition matrix following two steps. First, the population members are grouped under subsequent Pareto fronts using the non-dominating criteria. Then the non-dominated target vectors in individual Pareto fronts are again sorted in the descending order of the crowding distance. Let $\vec{Z}_i(0)$ be placed in the Pareto front pf_p with a rank of *q* based on the crowding disance measurements of the residents in pf_p. Then the state of $\vec{Z}_i(0)$ in the state transition matrix is identified by evaluating its rank $r_i(0) \in [1, NP]$, given as

$$r_i(0) = \sum_{u=1}^{p-1} \left| pf_u \right| + q \tag{16}$$

Here $|pf_u|$ represents the number of target vectors in the *u*-th Pareto front. Apparently, the first term of (16) represents the total number of solutions (appearing in the Pareto fronts former to pf_p) dominating $\vec{Z}(0)$. The ranking strategy in (16) ensures that the target vectors in the first (optimal) Pareto front pf_1 are allocated states only based on their respective crowding distance based ranks. Hence the solution with rank $r_i(0)$ is assigned to the state $r_i(0) \in [s_1(t), s_{NP}(t)]$. This is repeated for $i=[1, NP]$.

(c) Adaptive Selection of Scale Factor of DEMO: The reward/penalty based adaptation of the action probabilities helps in the right selection of scale factor *F* for the target vectors of the population. For example, a target vector at state $s_i(t)$ has a high probability of selecting $F=F_j$ if $p_{i,j}(t)$ is the largest among $p_{i,l}(t)$ for $l=[1, n]$. It is apparent that if $p_{i,j}(t) > p_{i,l}(t)$, for all *l*, then selection of $F=F_j$ at state $s_i(t)$ by the population members was rewarded many times before in the evolution process. Naturally, the learning experience will guide the target vector at state $s_i(t)$ to select $F=F_j$ with a high probability. Hence the probability of selecting the scale factor $F=F_j$ by a target vector at state $s_i(t)$ is governed by $p_{i,j}(t)$.

To maintain population diversity (by overcoming the premature convergence), in addition to the action probabilities, Roulette-wheel selection strategy is employed for selection of potentially useful scale factors. The Roulette-wheel selection of scale factor F=F_j for a target vector st state $s_i(t)$ being governed by the action probabilities $p_{i.l}(t)$, for l=[1, n], is realized by the following strategy. First, a random number *rand* is generated between (0, 1). Then we determine F_j, such that the cumulative probability of F=F_1 through F=F_{j-1} is less than *rand*, and the cumulative probability for F=F_1 through F=F_j is greater than *rand*. Symbolically, we need to hold

$$\sum_{l=1}^{j-1} p_{i,l}(t) \le rand \le \sum_{l=1}^{j} p_{i,l}(t) \qquad (17)$$

The adaptive selection strategy of scale factors F=F_j, for $F_j \in [F_1,\ F_n]$, is done for all the target vectors $\vec{Z}_i(t)$ for i=[1, NP].

(d) DEMO Algorithm: Once a scale factor $F_j \in [F_1,\ F_n]$ is selected for individual population members (using the action probability driven Roulette-wheel criterion), each target vector $\vec{Z}_i(t)$ participates in mutation and crossover machanisms of the traditional DEMO (as in sections III.A(b) and (c)) to generate a trial vector $\vec{U}_i(t)$ for i=[1, NP]. The objective function values $J_k(\vec{U}_i(t))$ are evaluated for k=[1, K] and i=[1, NP].

(e) Population Update and Reward/Penalty based Update of State Transition Table: Let the state of the target vector $\vec{Z}_i(t)$ before generation of $\vec{U}_i(t)$ is given by $r_i(t)$. If $\vec{U}_i(t)$ dominates $\vec{Z}_i(t)$ (indicating quality improvement of $\vec{Z}_i(t)$ on selection of $F_j \in [F_1,\ F_n]$), $\vec{U}_i(t)$ replaces $\vec{Z}_i(t)$ in the current population P_t and the corresponding action probabilities will be updated at state $r_i(t)$ following (14). Again, if $\vec{U}_i(t)$ and $\vec{Z}_i(t)$ are non-dominated, $\vec{U}_i(t)$ is incorporated in P_t. Otherwise, $\vec{U}_i(t)$ is abandoned from P_t. However, it is worth mentioning that in both the cases, there is no quality improvement of $\vec{U}_i(t)$ over $\vec{Z}_i(t)$ and hence the action probabilities at state $r_i(0)$ will be evaluated by (15) with non-zero penalty $x_{i,j}(t)$ = 1. Repeating the step for i=[1, NP] ultimately yiels a population P_t of size $|P_t| \in [NP,\ 2NP]$.

(f) Non-dominated Sorting based Ranking of Members: The members of P_t are then grouped under subsequent Pareto fronts using the non-dominating criteria. Then the non-dominated members of individual Pareto fronts are next sorted in the descending order of the crowding distance. Once the Pareto rank and the crowding distance based rank of a member have been identified, its composite rank $r \in [1,\ |P_t|]$ is determined using (16), where the size of the updated population $|P_t| \in [NP,\ 2NP]$. The entire population P_t is then sorted in ascending order of the composite rank of individual members.

(g) State Assignment of Population Members of Next Generation: The next generation population P_{t+1} is formed by selecting the first NP target vectors of the sorted population P_t. A target vector $\vec{Z}_i(t+1)$ of P_{t+1}, acquiring a rank of $r_i(t+1)$ (using (16)) is allocated to a state $r_i(t+1) \in [s_1(t+1),\ s_{NP}(t+1)]$. This is repeated for i=[1, NP].

(h) Convergence: After each evolution, steps (c) to (g) are repeated until one of the following conditions for convergence is satisfied. The conditions include restricting the number of iterations, maintaining function evaluations, or both, whichever occurs earlier.

```
Procedure DEMO-SLA-Induced-PPI-Prediction
Input: Set of functions and annotating GO terms of $N$ proteins under
       consideration and their individual ASAs.
Output: Predicted interaction between $N$ proteins.
Begin
I. Initialization:
  (i) Set the generation number $t \leftarrow 0$ and randomly initialize
      $D=N\times(N-1)/2+1$-dimensional $NP$ target vectors $\vec{Z}_i(t)$ of
      population $P_t$ for $i$=[1, $NP$] with $z_j(t)$ within [0, 1].
  (ii) Evaluate the objective function values $J_k(\vec{Z}_i(t))$ for $k$=[1, 3]
      and $i$= [1, $NP$] using (5), (8), and (11) respectively.
  (iii)Set $\gamma \leftarrow 1$ and $\lambda=\delta=0.5$. Initialize the action probabilities
      $[p_{i,j}(t)]$= 0.1 of the state transition table where $i$= [1, $NP$]
      represents the rank of a target vector at state $r_i(t)$ and $j$=[1, $n$]
      denotes the index of $n$ uniformly quantized scale factors $F$.
II. Ranking and State Assignment of Population Members:
  (i) Group the target solutions of population $P_0$ into Pareto fronts
      based on non-dominating criterion.
  (ii) Sort the members of individual Pareto fronts in descending
      order of crowding distance.
  (iii) Determine the rank $r_i(0) \in [1, NP]$ using (16) and assign it to a
      state $r_i(0) \in [s_1(0), s_{NP}(0)]$ for $i$=[1, $NP$].
III. While stopping criterion is not reached, do begin
   For $i$=1 to $NP$ do begin
    Let assign $s \leftarrow r_i(t)$.
    (i) Roulette-Wheel Adaptive Selection of Scale Factor:
      (a) Generate a random number $rand$ in [0, 1].
      (b) Select the scale factor $F_j \in [F_1, F_n]$ if
          $\sum_{l=1}^{j-1} p_{s,l}(t) \le rand \le \sum_{l=1}^{j} p_{s,l}(t)$ .
    (ii) Mutation: Generate $\vec{V}_i(t)$ for each $\vec{Z}_i(t)$ using the selected
         scale factor $F_j \in [F_1, F_n]$ following the dynamic of (12).
    (iii) Crossover: Generate $\vec{U}_i(t)$ for each pair of $\vec{Z}_i(t)$ and
         $\vec{V}_i(t)$ using (13). Evaluate $J_k(\vec{U}_i(t))$ for $k$=[1, 3] using (5),
         (8), and (11) respectively.
    (iv) Update Population and Action Probabilities:
         If $\vec{U}_i(t)$ dominates $\vec{Z}_i(t)$ then do
          (a) Assign zero penalty response $x_{i,j}(t) \leftarrow 0$ and update the
              action probabilities at state $s \in [s_1(t), s_{NP}(t)]$ by setting
                $p_{s,j}(t+1) \leftarrow p_{s,j}(t) + \lambda \times (1 - p_{s,j}(t))$
                $p_{s,l}(t+1) \leftarrow p_{s,l}(t) - \lambda \times p_{s,j}(t),\quad$ for $l=[1,n],\ l \ne j$
          (b) Replace $\vec{Z}_i(t)$ by $\vec{U}_i(t)$ .
         Else do
          (a) Assign non-zero penalty response $x_{i,j}(t) \leftarrow 1$ and update
              the action probabilities at state $s \in [s_1(t), s_{NP}(t)]$ by
```

```
            $p_{s,j}(t+1) \leftarrow p_{s,j}(t) - \delta \times (1 - p_{s,j}(t)) - \frac{\delta}{n}(n-1) + (n-1)$
            $p_{s,l}(t+1) \leftarrow p_{s,l}(t) + \delta \times p_{s,l}(t) + \frac{\delta}{n} - 1, \quad \text{for } l = [1,n], \; l \neq j$
         (b) If $\vec{U}_i(t)$ and $\vec{Z}_i(t)$ are non-dominated
              then include $\vec{U}_i(t)$ in $P_t$.
              Else discard $\vec{U}_i(t)$ .
              End If.
       End If.
   End For
 (v) Non-dominated Sorting based Ranking of Population Members:
      (a) Group the members of population $P_t$ of size $|P_t| \in [NP, 2NP]$ into Pareto fronts based on non-dominating criterion.
      (b) Sort the members of individual Pareto fronts in descending order of crowding distance.
      (c) Determine the composite rank of each population member using (16).
      (d) Sort the population $P_t$ in ascending order of the composite ranks.
 (vi) State Assignment of Members of Next Generation Population:
      (a) Pass the first (top) $NP$ members of sorted population $P_t$ to $P_{t+1}$ and discard the rest of the members of $P_t$ (having composite ranks greater than $NP$).
      (b) Assign a target solution $\vec{Z}_i(t+1) \in P_{t+1}$, acquiring a composite rank $r_i(t+1) \in [1, NP]$ to a state $r_i(t+1) \in [s_1(t+1), s_{NP}(t+1)]$ for $i=[1, NP]$.
 (vii) Increase the generation value $t \leftarrow t+1$.
End While.
IV. Decode the best target solution $\vec{Z}^{best}$ using Fig. 2 and 3 residing at state $s_1(t)$ and return the predicted PPI network.
```

IV. Experiments and Results

A. Performance Metrics

Four different classes of interaction can be observed when the PPI network obtained by proposed method is compared with the standard PPI network. The classes are namely *true positive* (*TP*), *true negative* (*TN*), *false positive* (*FP*) and *false negative* (*FN*).The relative performance of our proposed PPI prediction algorithm is compared with the competitors based on the well-known metrics enlisted in Table-I.

B. Comparative Framework and Database Used

The proposed method is compared with two groups of algorithms. The first group comprises evolutionary/swarm multi-objective meta-heuristic search algorithms including differential evolution for multi-objective optimization (DEMO) [10], firefly algorithm with non-dominated sorting (FANS) [12] and non-dominated sorting genetic algorithm-II (NSGA-II) [17]. All of them attempt to maximize the objective functions as proposed here to solve PPI prediction problem. The second category of competitor algorithms encompasses the existing non meta-heuristic computational models for PPI prediction including fuzzy support vector machine (FuzzSVM) classifier (using GO terms, domains and amino-acid sequence information) [18], relative specific similarity (RSS) method (using GO terms)[19], random decision forest (RDF) [20]. The best parameter setting is used for all the algorithms as mentioned in the respective source papers.

TABLE I: PERFORMANCE METRICS FOR PPI PREDICTION

Performance	Expressions
sensitivity (*recall*)	$\frac{TP}{TP + FN}$
specificity	$\frac{TN}{FP + TN}$
positive likelihood ratio	$Sensitivity/(1 - Specificity)$
negative likelihood ratio	$(1 - Sensitivity)/Specificity$
precision/positive predicted value	$\frac{TP}{TP + FP}$
negative predicted value	$\frac{TN}{TN + FN}$
accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
*F*1_*score*	$\frac{2 \times TP}{2 \times TP + FP + FN}$
Mathews correlation	$\frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$
receiver operating curve	Plot of *sensitivity* against (1−*specificity*)
area under curve (*AUC*)	Area under *ROC* Curve

To analyze the efficiency of our proposed algorithm to predict PPI and to validate the predictions, the predicted protein interactions are compared with the protein interaction data of BioGrid (July 2013) [21]. The BioGrid database consists of 6391 proteins and 326967 interactions. The standard non-interacting (or negative) datasets play an important role to validate our proposed method. In this paper we have generated the non-interacting protein pair by randomly pairing the proteins and removing those pairs which are already identified as positive pair. The Cartesian coordinates of the proteins in *Saccharomyces cerevisiae* (*SC*) are acquired from Protein Data Bank [22]. The GO terms of each protein for evaluating functional similarity is obtained from Saccharomyces Genome Database [23]. ASA is calculated using GETAREA [24].

C. Results and Performance Analysis

The experiment undertaken involves comparison of our proposed PPI prediction algorithms with the existing state-of-art techniques. In this paper PPI is represented as MOO problem. Since all the vectors in the approximate Pareto front *A* (*Front_Set*(1) of DEMO-SLA), found by an evolutionary multi-objective optimization algorithm, will be equally good, so to select the best one among many possible candidates, the following composite measure is considered for each vector $\vec{Z}_i \in A$.

$$J(\vec{Z}_i) = J_1^*(\vec{Z}_i) \times J_2^*(\vec{Z}_i) \times J_3^*(\vec{Z}_i) \quad \text{for } i = [1, |A|] \qquad (18)$$

where $|A|$ is the number of non-dominated solutions in *A* and the normalized estimate of $J_k(\vec{Z}_i) \in (0,1)$ for k=[1, 3] is given by

$$J_k^*(\vec{Z}_i) = J_k(\vec{Z}_i) \Big/ \sum_{l=1}^{|A|} J_l(\vec{Z}_k) \qquad (19)$$

The effective non-dominated solution vector $\vec{Z} \in A$ having the smallest $J\left(\vec{Z}_i\right)$ for i= [1, |A|] is considered for decoding the optimal PPI network.

The *receiver operating curves* (*ROC*s) for different PPI prediction algorithms for interactions obtained from BioGrid database is plotted in Fig. 5(a). *ROC*s are a useful technique for examining the efficacy of a prediction algorithm in inferring true 'interacting' and 'non-interacting' pairs of proteins. The curve plots *sensitivity* against (1−*specificity*). The relative positions of the ROC curves indicate the relative efficiency of the algorithms to predict truly interacting protein-pairs. A plot, corresponding to algorithm *X*, lying above and to the left to the plot for another algorithm *Y*, indicates greater efficiency of *X* over *Y* for PPI prediction. Based on the above-mentioned fact, it is evident from Fig. 5(a) that FANS exhibits highest efficiency. A quantitative measure of the *ROC* induced efficiency of a PPI prediction algorithm can be captured by its respective *area under curve* (*AUC*), as reported in Table-II. It is apparent from Fig. 5(a) and Table-II that *AUC*s for DEMO-SLA, DEMO, FANS and NSGA-II employing evolutionary/swarm optimization methods have attained higher values than other competitor classification method.

TABLE II: AREA UNDER CURVE OBTAINED FROM FIG. 5(A)

DEMO-SLA	DEMO	FANS	NSGA-II	FUZZ SVM	RSS	RDF
0.948 **(0.09)**	0.917 (0.17)	0.903 (0.19)	0.889 (0.19)	0.879 (0.29)	0.796 (0.38)	0.604 (0.96)

A plot of *precision* versus *recall (PROC curve)* is given in Fig. 5(b).we aim to obtain a reasonable value of *sensitivity* (*recall*) while putting more emphasis on *precision* since low reliability is one of the main weaknesses of the experimental methods. To assess the relative merits of the algorithms, a straight line is drawn making an angle of 45^0 with the *recall* axis such that it passes through all the curves corresponding to all contender algorithms. In our analysis, we have taken the distance from the origin to the intersecting point between the straight line and each *PROC* curve as a measure of the performance of respective algorithm. The higher the measure, the better is the performance. Symbol "≥" is used to represent the relative performance of any two algorithms. Using this convention, the ranking of the algorithms can be depicted as: DEMO-SLA ≥ DEMO ≥ FANS ≥ NSGA-II ≥ FuzzSVM ≥ RSS ≥ RDF. The plot of Fig. 5(b) indicates that the proposed PPI prediction algorithm in general offers good level of *precision* and *recall.*

Table-III is used to report the mean and standard deviation of best-of-run values of the performance metrics, for 25 independent runs of each PPI prediction algorithm. The standard deviation is given in parenthesis below its respective mean value. The statistical significance level of the difference of the 25 sample values of each metric of DEMO-SLA and any one of the remaining six competitive algorithms (DEMO, FANS, NSGA-II, FuzzSVM, RSS and RDF) is judged by Wilcoxon rank sum test [25] with a significance level α=0.05. The *p*-values obtained through the rank sum test between the best algorithm (DEMO-SLA as evident from Table-II) and each of the six remaining algorithms over the nine performance metrics are reported in third brackets in Table-III. Here NA signifies *not applicable* cases of comparing the best algorithm (DEMO-SLA) with itself. The null hypothesis concerned with the statistically equivalent performance of the *i*-th and the proposed DEMO-SLA based PPI prediction algorithms is rejected, if the *p*-value associated with their comparative performance analysis is less than α.

An analysis of Table-III indicates that the proposed DEMO-SLA has performed better than other algorithms. It is interesting to see that DEMO-SLA outperforms all its contenders (both meta-heuristic and non meta-heuristic PPI predictors) in a statistically significant fashion with respect to all the performance metrics except for DEMO. Here, DEMO based prediction method remains the second best algorithm, being surpassed by DEMO-SLA, however, insignificantly. One of the reasons of the superiority of our proposed meta-heuristic search algorithm over the traditional classification techniques (e.g., FuzzSVM, RSS, and RDF) used to predict PPI network, is its ability to handle the unbalanced dataset.

In order to justify the philosophy of maximizing individual objectives for predicting PPI, we have used a sub-network of the PPI dataset of *SC*, as given in Fig. 6, comprising 40 proteins (including CTR9, LEO1, RTF1, SPT16, PAF1, POB3, CKA2, CKA1, RPO21, CKB1, CKB2, HTZ1, SET2, CTK1, GAL11, SSF1, ELF1, DOA1, SRB5, CCR4, SUA7, SPT5, TFG2, SPT2, VPS71, SPT15, SWR1, HPR1, UBP6, SSN3, MFT1, SWI4, THP2, LGE1, DST1, RTR1, VPS72, SAP30, PHO23 and SIN3). The sub-network involves 283 positive and 497 negative interactions. The predicted PPIs for the same set of proteins, obtained using seven competitor algorithms, are pictorially represented in Fig. 7. Comparing Fig. 6 with Fig. 7, it is apparent that DEMO-SLA based method outperforms other competitors in predicting correct PPIs. The sub-network predicted by various competitor algorithms is quantitatively analyzed with respect to nine performance metrics as provided in Table- IV. It is apparent from Table-IV that DEMO-SLA based PPI predictor has consistently performed better with near optimal values of nine performance metrics. The values in the table in turn attest the closeness between the original and DEMO-SLA based PPI sub-networks.

V. CONCLUSION

In this paper, a novel method is proposed using SLA induced DEMO (called DEMO-SLA), to predict PPI network. The solution to the possible PPI network here is represented by a solution vector comprising the weights (in [0, 1]) of protein interactions and the threshold based on which the connection is established between the proteins. Here the effect of three essential characteristic features to predict PPI is analyzed. The features include i) commonality in neighboring proteins of interacting protein pairs, ii) their functional similarity and iii) their accessible solvent area upon binding. Experiments undertaken reveal the superiority of the proposed method and the results clearly show that the proposed method is effective in predicting PPIs.

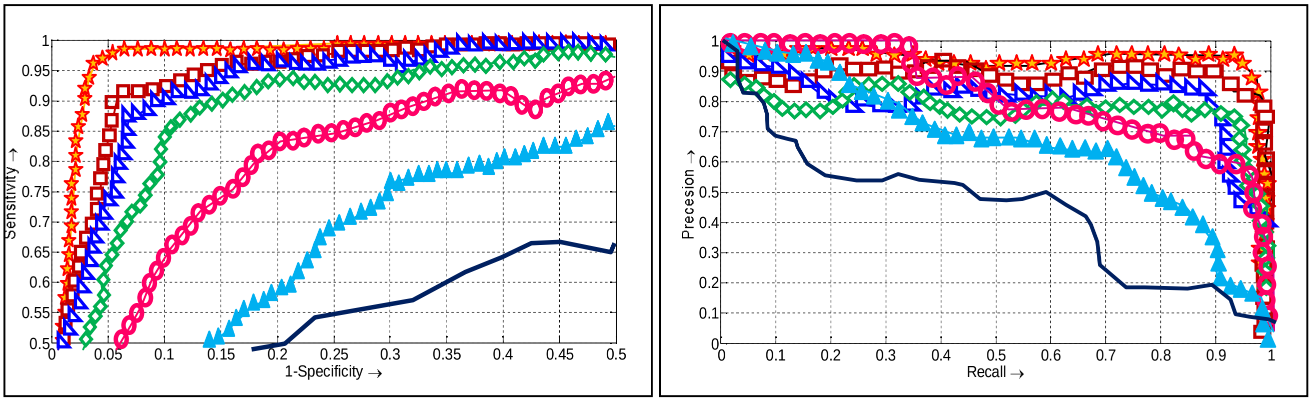


Fig. 5. (a) ROC and (b) PROC plots for different PPI prediction algorithms for BioGrid with symbols: ☆for DEMO-SLA, □for DEMO, ◇for FANS, ◺for NSGA-II, ○for FuzzSVM, ▲for RSS and −− for RDF

TABLE III: COMPARISON OF DIFFERENT PPI PREDICTION ALGORITHMS FOR 25 RUNS (BEST METRIC VALUES MARKED IN BOLD)

Algorithms	Sensitivity	Specificity	PLR	NLR	Precision	NPV	Accuracy	F1_score	MCC
DEMO- SLA	**0.9399** **(0.167)** **[NA]**	**0.9488** **(0.233)** **[NA]**	**18.4552** **(0.139)** **[NA]**	**0.0869** **(0.142)** **[NA]**	**0.9687** **(0.016)** **[NA]**	**0.8901** **(0.117)** **[NA]**	**0.9123** **(0.159)** **[NA]**	**0.9456** **(0.068)** **[NA]**	**0.8892** **(0.018)** **[NA]**
DEMO	0.9023 (0.178) [0.0742]	0.9301 (0.251) [.0566]	16.7843 (0.152) [0.0798]	0.1221 (0.162) [0.1895]	0.9132 (0.014) [0.0643]	0.8765 (0.212) [0.0652]	0.8912 (0.163) [0.1486]	0.9032 (0.070) [0.1939]	0.8411 (0.033) [0.0789]
FANS	0.8843 (0.183) [0.0409]	0.9145 (0.318) [0.0408]	8.4563 (0.173) [0.0572]	0.1670 (0.231) [0.0812]	0.8934 (0.017) [0.0357]	0.8025 (0.227) [0.0451]	0.8511 (0.175) [0.0645]	0.8726 (0.074) [0.0724]	0.6985 (0.047) [0.0429]
NSGA-II	0.8612 (0.200) [0.0402]	0.8937 (0.320) [0.0326]	7.5819 (0.180) [0.0361]	0.2045 (0.270) [0.0250]	0.8517 (0.020) [0.0340]	0.7388 (0.260) [0.0259]	0.8226 (0.187) [0.0339]	0.8466 (0.080) [0.0308]	0.6450 (0.050) [0.0400]
FuzzSVM	0.8437 (0.420) [0.0356]	0.7816 (0.560) [0.0288]	3.6563 (0.530) [0.0247]	0.3129 (0.510) [0.0119]	0.7822 (0.390) [0.0339]	0.6619 (0.650) [0.0197]	0.7154 (0.560) [0.0304]	0.7496 (0.440) [0.0260]	0.5925 (0.670) [0.0360]
RSS	0.7582 (0.600) [0.0260]	0.6534 (0.650) [0.0250]	2.0147 (0.560) [0.0244]	0.4664 (0.670) [0.0074]	0.7123 (0.500) [0.0329]	0.6693 (0.680) [0.0167]	0.7110 (0.610) [0.0244]	0.7172 (0.520) [0.0167]	0.5456 (0.790) [0.0349]
RDF	0.6718 (0.820) [0.0053]	0.5845 (0.790) [0.0015]	1.6732 (0.630) [0.0091]	0.6616 (0.750) [0.0029]	0.5997 (0.610) [0.0130]	0.5876 (0.830) [0.0014]	0.6321 (0.710) [0.0035]	0.6121 (0.890) [0.0068]	0.5123 (0.830) [0.0018]

TABLE IV: VALUES OF PERFORMANCE METRICS OBTAINED FROM THE PPI SUB-NETWORK (FIG. 6AND 7) (BEST METRIC VALUES MARKED IN BOLD)

Algorithms	Sensitivity	Specificity	PLR	NLR	Precision	NPV	Accuracy	F1_score	MCC
DEMO- SLA	**0.9175**	**0.9015**	**12.463**	**0.1256**	**0.8856**	**0.9278**	**0.9065**	**0.8712**	**0.8198**
DEMO	0.8654	0.8389	7.8335	0.1273	0.7478	0.9121	0.8322	0.8231	0.6729
FANS	0.7812	0.8128	5.1136	0.2497	0.6345	0.8641	0.7767	0.7564	0.5744
NSGA-II	0.6389	0.7305	3.4728	0.5351	0.5545	0.7547	0.6658	0.5634	0.3457
FuzzSVM	0.5595	0.5670	2.1755	0.8163	0.4129	0.6532	0.5343	0.4469	0.2533
RSS	0.5106	0.4828	1.0686	1.1349	0.3546	0.6376	0.4255	0.4186	0.1558
RDF	0.4940	0.5090	0.9860	1.0134	0.3595	0.6321	0.5110	0.4112	0.0565

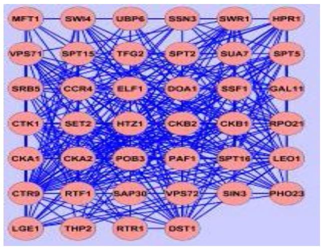
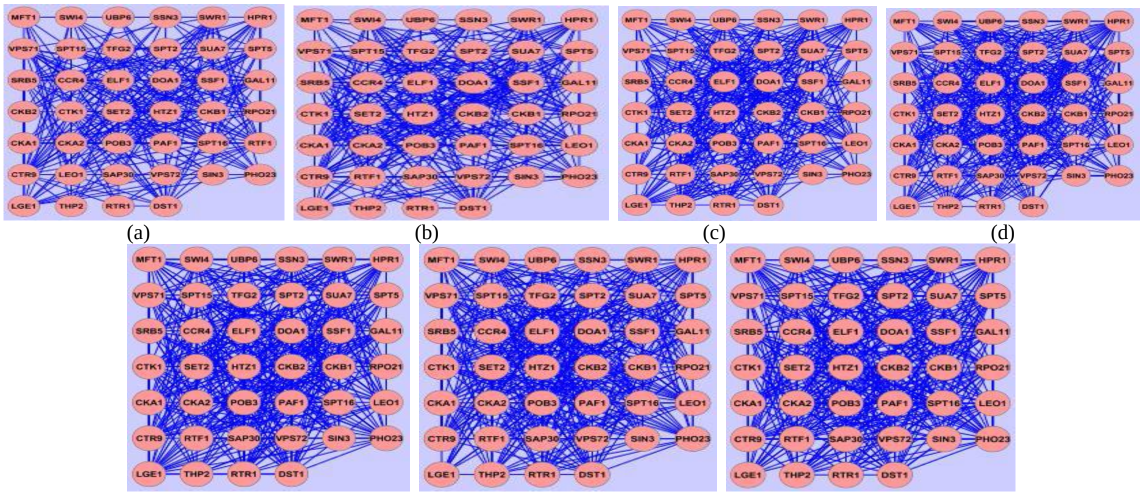


Fig. 6. Original sub-network of PPI network in yeast



(a) (b) (c) (d)

(e) (f) (g)

Fig. 7. Sub-network obtained by PPI prediction algorithms: (a) DEMO-SLA (b) DEMO (c) FANS (d) NSGA-II (e) Fuzz-SVM (f) RSS, and (g) RDF

REFERENCES

[1] Alberts, B., Bray, D., Johnson, A., Lewis, J., Raff, M., Roberts, K. ... & Latchman, D. S. (1998). Essential Cell Biology: An Introduction to the Molecular Biology of the Cell. Trends in Biochemical Sciences, 23(7), 268-268.

[2] Schwikowski, B., Uetz, P., & Fields, S. (2000). A network of protein–protein interactions in yeast. *Nature biotechnology*, *18*(12), 1257-1261.

[3] Chowdhury, A., Rakshit, P., & Konar, A. (2015). Prediction Of Protein-Protein Interaction Network Using A Multi-Objective Optimization Approach. *Journal of Bioinformatics and Computational Biology*.

[4] Theofilatos, K.; Dimitrakopoulos, C.; Tsakalidis, A.. Likothanassis, S.; T. Papadimitriou, S.; Mavroudi, S. Computational Approaches for the Prediction of Protein-Protein Interactions: A Survey. *Curr.Bioinform.*, **2011,** *6*(4), 398-414.

[5] Friedel, C. C., & Zimmer, R. (2006). Inferring topology from clustering coefficients in protein-protein interaction networks. *BMC bioinformatics*, *7*(1), 519.

[6] Paradesi, M.S.R., Caragea, D., & Hsu, W.H. (2007). Structural Prediction of Protein-Protein Interactions in Saccharomyces cerevisiae, Proc. of IEEE 7th International Symposium on BioInformatics and BioEngineering, vol. 2, pp. 1270-1274.

[7] Han, D., Kim, H. S., Seo, J., & Jang, W. (2003). A domain combination based probabilistic framework for protein-protein interaction prediction. *Genome Informatics*, *14*, 250-259.

[8] Eisenberg, D., & McLachlan, A. D. (1986). Solvation energy in protein folding and binding.

[9] Bender, E. A. (1996). *Mathematical methods in artificial intelligence*. IEEE Computer Society Press.

[10] Rakshit, P., Konar, A., Das, S., Jain, L. C., & Nagar, A. K. (2014). Uncertainty management in differential evolution induced multiobjective optimization in presence of measurement noise. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, *44*(7), 922-937.

[11] Lakshmivarahan, S., & Thathachar, M. A. L. (1973). Absolutely expedient learning algorithms for stochastic automata. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, (3), 281-286.

[12] Chowdhury, A., Rakshit, P., Konar, A., & Nagar, A. K. (2015, May). A multi-objective evolutionary approach to predict Protein-Protein Interaction network. In *Evolutionary Computation (CEC), 2015 IEEE Congress on* (pp. 1628-1635). IEEE.

[13] Licamele, L., & Getoor, L. (2007). Predicting protein-protein interactions using relational features.

[14] Schlicker, A., Domingues, F. S., Rahnenführer, J., & Lengauer, T. (2006). A new measure for functional similarity of gene products based on Gene Ontology. *BMC bioinformatics*, *7*(1), 302.

[15] Lee, B., & Richards, F. M. (1971). The interpretation of protein structures: estimation of static accessibility. *Journal of molecular biology*, *55*(3), 379-IN4.

[16] Basak, A., Das, S., & Tan, K. C. (2013). Multimodal optimization using a biobjective differential evolution algorithm enhanced with mean distance-based selection. *Evolutionary Computation, IEEE Transactions on*, *17*(5), 666-685.

[17] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, *6*(2), 182-197.

[18] Chiang, J. H., & Lee, T. L. M. (2008). In Silico Prediction of Human Protein Interactions Using Fuzzy–SVM Mixture Models and Its Application to Cancer Research. *Fuzzy Systems, IEEE Transactions on*, *16*(4), 1087-1095.

[19] Wu, X., Zhu, L., Guo, J., Zhang, D. Y., & Lin, K. (2006). Prediction of yeast protein–protein interaction network: insights from the Gene Ontology and annotations. *Nucleic acids research*, *34*(7), 2137-2150.

[20] Chen, X. W., & Liu, M. (2005). Prediction of protein–protein interactions using random decision forest framework. *Bioinformatics*, *21*(24), 4394-4400.

[21] Stark, C., Breitkreutz, B. J., Reguly, T., Boucher, L., Breitkreutz, A., & Tyers, M. (2006). BioGRID: a general repository for interaction datasets. *Nucleic acids research*, *34*(suppl 1), D535-D539.

[22] Bernstein, F. C., Koetzle, T. F., Williams, G. J., Meyer, E. F., Brice, M. D., Rodgers, J. R., ... & Tasumi, M. (1978). The Protein Data Bank: a computer-based archival file for macromolecular structures. *Archives of biochemistry and biophysics*, *185*(2), 584-591.

[23] Cherry, J. M., Hong, E. L., Amundsen, C., Balakrishnan, R., Binkley, G., Chan, E. T., ... & Fisk, D. G. (2011). Saccharomyces Genome Database: the genomics resource of budding yeast. *Nucleic acids research*, gkr1029.

[24] Fraczkiewicz, R., & Braun, W. (1998). Exact and efficient analytical calculation of the accessible surface areas and their gradients for macromolecules. *Journal of Computational Chemistry*, *19*(3), 319-333.

[25] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research, 7,1-30.

[26] Seco N, Veale T, Hayes J (2004) An intrinsic information content metric for semantic similarity in wordnet. In: ECAI. pp 1089–1090.