A GENETIC-BASED APPROACH FOR DISCOVERING PATHWAYS IN PROTEIN-PROTEIN INTERACTION NETWORKS

NGUYEN Hoai Anh*, VU Cong Long*, TU Minh Phuong† and BUI Thu Lam*
*Faculty of Information Technology
Le Quy Don Technical University, Ha Noi, Viet Nam
Emails: nguyenhoaianh@yahoo.com, long.vucong@gmail.com, lam.bui07@gmail.com
† Department of Computer Science
Posts and Communications Institute of Technology, Ha Noi, Viet Nam
Email: phuong.tu@gmail.com

***Abstract*—This paper introduces an approach of using the genetic algorithm for orienting protein-protein interaction networks (PPIs) and discovering pathways. Biological pathways such as metabolic or signaling ones play an important role in understanding cell activities and evolution. A cost-effective method to discover such pathways is analyzing accumulated information about protein-protein interactions, which are usually given in forms of undirected networks or graphs. Previous findings show that orienting protein interactions can improve pathway discovery. However, assigning orientation for protein interactions is a combinatorial optimization problem which has been proved to be NP-hard, making it critical to develop efficient algorithms.**

For our proposal, we first study the mathematical model of the problem. Then, based on this model, a genetic algorithm is designed to find the solution for the problem. We conducted multiple runs on the data of yeast PPI networks to test the best option for the problem. The preliminary results were compared with the results of the random search algorithm, which was shown to the best in dealing with this problem, in terms of the run time, fitness function values, especially the ratio of gold standard pathways. The findings show that our genetic-based approach addressed this problem better than the random search algorithm did.

***Keywords*-genetic algorithms; protein; interaction; network;**

I. INTRODUCTION

PPI databases have been the source of interaction information in biological cells. This kind of databases is usually large since data is aggregated over time from the experimental findings. So the discovery of new knowledge from the database has become a challenge for computational biology. Note that edges representing PPIs have been experimentally defined and tested. Certainly reconstructing signaling pathways (or networks) has attracted a lot of attentions: the reconstruction of regulatory networks [1, 2], the analysis of metabolic networks [3, 4], and the discovery of signaling networks and pathways [5, 6]. However, directionality of interactions in networks has not been investigated thoroughly, while direction is necessary to know how information is moved from one to another. The orientation of the signaling network is more difficult than the regulatory and metabolic networks, because lack of orientation information. For example, ChIP-chip and ChIP-Seq studies [7] identify which transcription factors regulate genes, studies of microRNAs often look for targets [8] and motif studies are performed upstream of genes [9]. Similarly, metabolic networks are often modeled using knowledge regarding the order of genes and enzymes [10]. In contrast, it is a fact that PPI data is almost always undirected; therefore the problem of orienting interaction edges for signal transmission in signaling networks is costly. Typical works in this area can be found in [11, 12, 13]. This demonstrates the attraction of finding an efficient algorithm for edge-orientation in PPI networks.

For an overview, authors in [11] stated the problem of orienting edges in the protein interaction network as an optimization problem and proved that this problem is NP hard. Then they presented a random orientation (plus local search) algorithm (ROLS) to perform edge orientation and evaluated calculated results with the data from biological experiments in order to determine if the path found is consistent with the experimental or not. The results were also compared with several algorithms proposed in [14, 15]. In evaluating the algorithm results, the authors found out 37 standard pathways that had been tested through biological experiments. But there were still paths that did not appear in the standard set and such interactions could not occur in biological experiments, even though the objective function values of these pathways were high. Formerly, relatively few methods have been developed to clearly solve the edge orientation problem. In [16], the authors defined the maximum tree orientation (MTO) problem, which focused on reachability.

In the framework of this paper, a different approach is used to solve the problem of edge orienting outlined in [11]; in particular we designed a genetic algorithm (GA) for it. GA is one of popular and successful computational models in the field of intelligent computing [17], especially for dealing with NP-hard problems. Along with other intelli-

978-1-4799-3400-3/13/$31.00 ©2013 IEEE

gent computing techniques such as fuzzy computing, neural networks, multi-agent systems, genetic algorithms develop more and more strongly and are widely applied in different fields of life. Our GA design takes into account conflicting elements in PPI networks in order to reduce unnecessary edges, thus greatly improve computing speed. Results show that our algorithm found a good solution for this problem.

The structure of our paper consists of 5 sections: Section 1 introduces the problem, Section 2 gives general knowledge of the problem and the genetic algorithm, Section 3 describes in detail the GA algorithm designed to solve the problem posed, Section 4 presents actual experimental data on PPIs of yeast and make an assessment of the results achieved by the algorithm. The final part is the paper conclusion.

II. Background

A. Problem of orienting edges in protein interaction networks PPIs

Proteins are involved in most of biological processes in cells; however, instead of operating independently, they interact with other proteins or macromolecules such as DNA and RNA. They together form a complex network of interactions to perform biological functions. Along with experimental studies, the database of information about protein interactions (PPI) is also formed and developed over time. This database is constantly updated and added with new elements of protein interactions announced by researchers around the world. An example is given in Figure 1 where the graph shows a part of the network of protein interactions in yeast created by *Cytospase* software. From the graph, we can see that the protein interaction network of an organism can be represented by an undirected graph in which each vertex denoted is a protein and each edge represents an interaction of PPIs network. This interactive network contains signaling pathways that comes from a protein source through transformation to transmit biological information to a specific target protein. With signaling pathways verified by experiment, it is gathered into a database to serve for the interpretation of biological problems. The discovery of the signaling pathway in protein interaction networks are still performed by scientists. The problem here is the need to have a certain method to reconstruct the known signal pathways from the undirected protein interaction networks and analysis to make predictions about new signaling pathways for purposes of biological studies such as understanding disease signals, creating new drugs to treat diseases caused by the deviation from the signal pathway.

This is a difficult problem because there are many paths that can link two proteins in the interaction network. However, we can establish assumptions to simplify the problem. It is likely that biological responses are controlled by reasonably short signaling cascades, so we should only search for length-bounded paths. So far, pathways in signaling

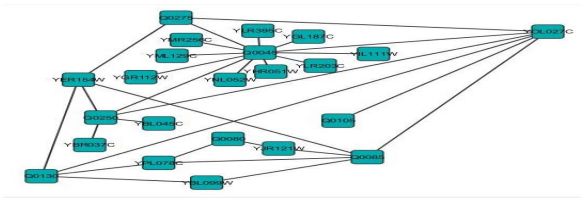

Figure 1. A part of the protein interaction network of yeast includes 23 proteins and 30 interactions. According to database BIOGRID network 2.0.51 of yeast, its PPIs have 5570 proteins and 140849 interactions [18].

databases such as KEGG and the Science Signaling Database of Cell Signaling on average contain only five edges between a target and its closest source [11]. The goal of the problem is to extract the signal pathways of length k from source to target that are highy reliable.

We can formally express a PPI network by a weighted undirected graph $G = (V, E)$, where V is the set of vertices of the graph labeled by names of proteins, E is the set of edges of the graph describing interactions between proteins. We have $u, v \in V$, edge $e = (u, v) \in E$ if and only if u, v can interact with each other. We define $S \subseteq V$ as the set of source vertices of paths and $T \subseteq V$ as the set of target vertices of paths. A path has a maximum length of at most k between pairs of sources - target in form of $\langle s_i, t_i \rangle$, where $s_i \in S \subseteq V$ and $t_i \in T \subseteq V$. The goal is to orient the edge $e = (u, v) \in E$ from u to v or from v to u so that the weight of the path from source to target with the maximum length k is the largest. Each path has the form $p = (v_1, v_2), (v_2, v_3), , (v_l, v_{l+1})$ and $l \leq k$ for some pairs $\langle s_i, t_i \rangle$. A path is said to be satisfied in the orientation graph if and only if every edge (v_j, v_{j+1}) has its orientation path from v_j to v_{j+1} in the network.

All vertices and edges in the graph have weights which are denoted $w(v)$ and $w(e)$ respectively. Because all vertices $v \in V$ are involved in the signaling pathway (i.e., the graph does not have any isolation peak) so here we assign 1 to every weight of the vertices. Edge weight is a value in the range $[0, 1]$, which is based on the probability of each protein interaction. The value of the weight is typical for reliability in the presence of an edge or the involvement of a protein in the path, and the weight of the path is the probability of a protein interaction in path calculated by the formula

$$w(p) = \prod_{v \in p} w(v) * \prod_{e \in p} w(e) \tag{1}$$

Thus, the goal of the problem is to maximize the total weight of the satisfied paths; or in other words, to optimize the objective function

$$\sum_{p \in P} Is(p) * w(p) \tag{2}$$

Where P is the set of paths between sources and targets with lengths of at most k. $w(p)$ is the path weight. $Is(p)$ is

a function of only two values 0 or 1. $Is(p) = 0$ if path p is not satisfied, $Is(p) = 1$ if path p is satisfied.

1) Overview on Genetic Algorithms: GA is one of development tendencies in evolutionary computation. It was researched, developed, and applied since the last century in search, optimization and machine learning. The exploitation of the evolution principle as a heuristics has made the genetic algorithm an effective approach for the optimization problem (with acceptable solutions) without using traditional conditions (continuous or differentiable) as prerequisites.

One of the important characteristics of GA is the usage of a set (or *population*) of solutions. The search is done parallel on multiple points that can interact with each other according to natural evolution principles. In the context of using genetic algorithms, we can use the concept of "*individual*" in equivalence with the notion of "*solution*". The basic steps of a genetic algorithm are described as follows:

- ***Step 1***: $t = 0$; Initialize $pop(t) = \{x_1, x_2, \ldots, x_N\}$, N is the population size.
- ***Step 2***: Evaluate $pop(t)$.
- ***Step 3***: Create the mating pool $MP = se\{pop(t)\}$ with se is the selection operator.
- ***Step 4***: Define $pop'(t)$=$cr\{MP\}$, with cr is the crossover operator.
- ***Step 5***: Define $pop''(t)$=$mu\{pop'(t)\}$, with mu is mutation operator.
- ***Step 6***: Evaluate $pop''(t)$
- ***Step 7***: Define $pop(t+1)$=$pop''(t)$ and set $t = t + 1$
- ***Step 8***: Go back Step 3, if the stopping criterion is not satisfied.

Individual representation: This is one of the important tasks in designing genetic algorithms, deciding the application of evolutionary operators. One of the traditional representations of genetic algorithms is binary representation. With this, each individual in the population is represented as a sequence of bits 0 and 1, also known as chromosomes. Each chromosome represents a parameter of individual components.

Selection operator: The selection of individuals can be done when we need a number of individuals to produce the next generation. Each individual has an adaptive value (fitness). This value is used to determine which individual to choose. The selection method used in this paper is tournament selection. This method bases on the fitness function value to choose individuals.

Crossover operator: crossover operator is applied to generate new children individuals from parent individuals with the best traits inherited from their parents. In the search context, the crossover operator performs a search around the area of the solution represented by individual parents.

Mutation operator: Similar to crossover operator, mutation operator is used to simulate biological mutations. The result of mutations often generates new individuals which are different from their parents. The purpose of mutation operator is to expand searching areas out of local ones.

III. METHODOLOGY

The idea of designing genetic algorithms to solve the edge orientation problem starts with a randomly initialized population (population P) of individuals in which the number of individuals of the population is a constant natural number n, each individual is represented by the sequence of the chromosomes. Population will be evolved over many generations. The best individual of each generation is kept for the next population and we apply the local search as well. After the evolution process completed, the best individual in the population will eventually be the selected path of the problem.

In the following, we will discuss the design of representation as well as operators.

A. Representation

Individual representation is a very important task in the design of genetic algorithms because it will affect all operations of the algorithm and calculation of fitness values. With the edge orientation problem for weighted undirected graphs, we assume the followings:

- Only two directions can be assigned for an edge of the graph, so binary representation is suitable to represent an individual's chromosome (*if we consider a valid direction is* 1*, then the opposite direction will be* 0).
- For the graph G, we will find out P' which is the set of all possible paths for pairs of source $s \in S$ and target $t \in T$, here we only consider edges in P' that have the conflicting (i.e. both directions are existed in paths). After having set P', we will find E' which is the set of all edges in the path $p \in P'$ that have conflict. Assume that E' have n conflicted edges, the question is orienting them.

Thus the initial population P is the set of orientation possibilities for conflicted edges in E', each individual of this population represents an orientation possibility and each individual's gene will correspond to a conflicted edge, so each individual will have n genes and each gene receives one of two values: 0 or 1. So there are 2^n individuals forming a huge search space given a large n.

Let suppose that PPIs have been performed by an undirected graph G whose weight is shown in Figure 2. The input consists of a set of source proteins $S = ProA, ProF, ProG$; a set of target proteins $T = ProB, ProF, ProG$ the largest path length is $k = 5$.

Apply to the graph G in Figure 2 we see set P' includes paths:

$p_1 = \{(proA, proC), (proC, proB)\}$

$p_2 = \{(proA, proC), (proC, proD), (proD, proE), (proE, proF)\}$

$p_3 = \{(proA, proC), (proC, proD), (proD, proE),$

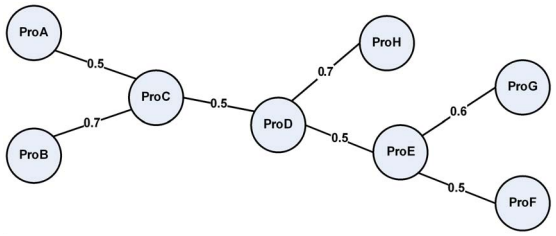

Figure 2. A weighted undirected graph representing a network of protein interactions.

$(proE, proG)\}$
$p_4 = \{(proF, proE), (proE, proD), (proD, proC),$
$(proC, proB)\}$
$p_5 = \{(proF, proE), (proE, proG)\}$
$p_6 = \{(proG, proE), (proE, proD), (proD, proC),$
$(proC, proB)\}$
$p_7 = \{(proG, proE), (proE, proF)\}$

There are the following conflicted edges in set P'
$E' = \{(proC, proD), (proD, proE), (proE, proF),$
$(proE, proG)\}$

So the 2^4 individuals and each individual have 4 genes.

B. Evaluation of individuals

The individual evaluation involves calculating the fitness value (2), An individual with greater fitness function value is assessed to be better. For example, in the initial population P of the graph G in Figure 2.

Case 1. Choose individual $C1$, then the graph G is oriented as shown in Figure 3a. The paths which are satisfied with this orientation is p_1, p_2, p_3 , fitness function value in this case is
$f(C1) = w(p_1) + w(p_2) + w(p_3) = 0.5 * 0.7 + 0.5 * 0.5 * 0.5 * 0.5 + 0.5 * 0.5 * 0.6 = 0.485$

Case 2. Choose individual C5, then the graph G is oriented as shown in Figure 3b. Only one path is satisfied with this orientation, which is p_1, fitness function value in this case is
$f(C5) = w(p_1) = 0.5 * 0.7 = 0.35$

We say case 1 has better objective function value than case 2.

C. The operators

Selection operator: For GA, we need to create a mating pool by the mean of selection. In order to get an individual for the pool, we randomly choose two individuals in the current population, compare their fitness values, and then pick the one with better fitness. For example in population P there are 4 individuals shown in Figure 4a. In the first random selection, we get two individuals $C1$ and $C5$, compare their fitness, we have $f(C1) > f(C5)$, should we choose $C1$ (Figure 4b). Similarly, with the second random selection we choose $C10$. So after two times, we get two good individuals for the pool (in other word becoming parents for the next life).

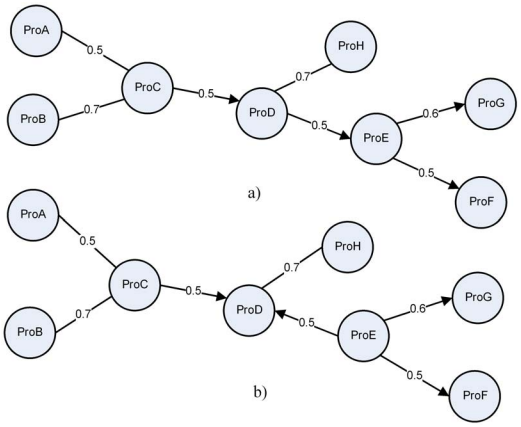

Figure 3. Orientation of conflicted edges in the graph G.

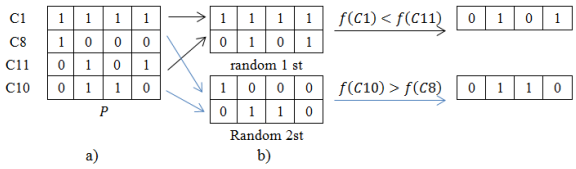

Figure 4. Operator selection in the populations P.

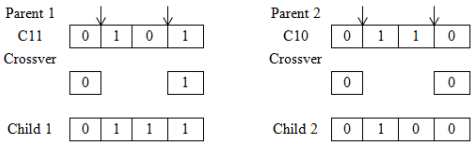

Figure 5. Simulation crossover operator between C11, C10 individuals.

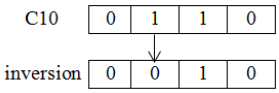

Figure 6. Simulation of mutation operator with individual C10.

Crossover operator: In our algorithm, we use a two-point crossover operator. The crossover operation calls for two index points to be selected on the parent bit-strings. Everything between the two points is swapped between the parent organisms, rendering two child organisms. For example in population P in Figure 4a, we have selected two individuals for life after parenthood which are $C11, C10$. Crossover operator is modeled as shown in Figure 5.

Mutation operator: This is a simulation of mutations in biology. In our search problem, the mutation operator is seen as a way to bypass local extreme points of the fitness function. Our method uses bit inversion: randomly select a bit and change its state to the opposite state. The mutation operator is modeled as shown in Figure 6.

D. Conservation of elite individuals

After each generation, we can always find out the best individual from that generation. We try to look around such individual to find better ones to preserve. Doing this enables the algorithm to quickly converge to the global extreme point, thus improve the running time of the algorithm. In our algorithm, we use local search methods in an individual *localSearch(individual)*. We reverse each edge in the individual, then calculate the disparity of the fitness function . The reversed edge with maximum positive will give better individual.

IV. CASE STUDY

A. Prepare data

Yeast PPIs interaction network

In this paper we use the database of yeast PPIs taken from database BioGRID (`http://thebiogrid.org/download.php`), this is an on-line database of genetic interactions of organisms on a large scale. As mentioned above, this database is updated over time basing on new researches and findings by experiments from biologists. Therefore, for ease of comparison between our results and those of existing algorithms, we use the same database version 2.0.51 BioGrid with the authors [11]. This database is a two-dimensional data table that has 140849 lines; each line contains interactive information of a pair of source and target proteins. We are interested in information about experiment types which are used to detect interactions, because we will combine this information with the table of confidence scores for each type of experiment to determine weights for each interaction edge [18].

The edge weight depends on two factors: First, the reliability of the experiment type; Second, the number of separate experiments that have such interactions. In essence, the edge weight of an edge $(Pro1, Pro2)$ is the probability of interacting protein pairs $Pro1$ and $Pro2$ which is calculated using the formula

$$P(interact(Pro1, Pro2)) = 1 - \prod_{i \in I_{Pro1,Pro2}} (1 - c(i)) \quad (3)$$

where i is a member of the set $I_{Pro1,Pro2}$, which contains all separate interactive experiments from the database of PPIs, and $c(i)$ the reliability of experiment type i.

After determining weights for protein interactions in (3) we get a data sheet of the protein interaction pairs and weights of the interactions. This is the input data of the algorithm.

The source target protein pairs

They are used to make assessments in biology perspective on the paths found by our algorithm in comparison with those proven experimentally (called the *gold standard pathway*). The algorithm inputs will use a set $S \subseteq V$ that includes experimentally proven source proteins in a path and a set $T \subseteq V$ that contains proteins where signaling pathway ends. List of source - target pairs of is determined basing on the standard pathway taken from [18].

B. Testing scenario

First, we use the Depth First Search algorithm for a set of paths from source to target, then generating a set of conflicted edges. The yeast PPIs database gives us 993 conflicted edges. After that, GA is used to find the best orientation setting for conflicted edges. We conducted the test run many times to compare results obtained by Genetic Algorithms (GA) designed by us with the results of the random orientation algorithm plus local search, called ROLS, (Note that in [11], ROLSs performance was shown better than that of the algorithms MIN-SAT, MAX-CSP and MTO). With each loop, we find an outstanding individual and keep it for the next generation.

The test run is planned as follows: Set the initial population of 100 individuals, each individual has n chromosomes (which equals to the total number of conflicted edges in the set of conflicted edges). Input parameters for genetic algorithms include: total generation number of 50, crossover probability of 0.9 and mutation probability of 0.001. To ensure the same experimental conditions, we also run tests 20 times and take the highest value of the objective values (like in [11]), for each run populations and individuals are initialized randomly.

C. Results and analysis

1) Performance assessment using the objective function: In terms of the objective value, GA designed by us has given out greater performance than that of ROLS (see Table I). The average objective values obtained by GA are much better than ROLS's (7943.65 ± 53.11 comparing to 7831.75 ± 56.17). Also GA found the best value of 8027, which was not found by any run of ROLS. This shows that our GA can solve this problem more effectively than ROLS. The use of the simulated evolution makes GA much better than its random counterpart ROLS.

2) Evaluation of the algorithm using gold standard pathways: Regarding biology perspective, we use number of standard pathways as a criterion to assess the ability of the algorithm to find experimentally proven pathways. To assess this criterion, we ranked all the paths found by the genetic algorithm and ROLS according to different metrics and calculate how many of the 100 paths have exactly 5 edges (or 6 proteins) that are at least partially located in the standard pathway. According to the criteria given in [11], partially means the path has at least four of the six proteins consecutively found in both standard path and the satisfactory path returned by the algorithm. The results were listed in Tables II and III for both GA and ROLS and with

Table I
THE RESULTS OF THE BEST OBJECTIVE VALUES OBTAINED BY GA AND ROLS AMONG 20 RUNS.

Run	**GA**	**ROLS**
1	7825	7737
2	7880	7767
3	7884	7769
4	7896	7778
5	7913	7778
6	7914	7781
7	7916	7787
8	7916	7798
9	7917	7812
10	7933	7827
11	7956	7834
12	7958	7843
13	7968	7852
14	7982	7877
15	7982	7881
16	7983	7884
17	7985	7898
18	8016	7902
19	8022	7914
20	8027	7916
MEAN	**7943.65**	**7831.75**
STD	**53.11**	**56.17**

Table II
RESULTS FOR GA: NUMBER OF THE TOP 100 RANKED PATHS THAT PARTIALLY MATCHED GOLD STANDARD PATHWAYS

Run	Path weight	Max. edge weight	Avg. edge weight	Min. edge weight	Max. edge use	Avg. edge use	Min. edge use	Max. de-gree
1	37	11	37	34	8	40	30	14
2	50	11	50	40	10	40	39	16
3	7	9	7	4	3	8	24	5
4	37	11	37	34	8	40	30	14
5	35	11	35	24	8	37	30	14
6	34	11	34	28	8	36	16	12
7	35	12	34	25	8	37	30	14
8	38	11	37	33	8	40	30	14
9	29	11	29	19	8	40	33	14
10	40	11	40	30	8	40	39	14
11	35	11	35	27	8	37	30	14
12	3	4	3	2	1	2	16	2
13	35	11	34	25	8	37	30	14
14	39	13	39	32	8	40	30	14
15	29	11	29	19	8	37	33	14
16	38	13	38	33	8	40	30	14
17	13	8	13	15	3	14	24	5
18	34	12	34	25	8	37	30	14
19	35	11	35	28	8	37	30	14
20	17	3	17	14	3	16	33	5
MEAN	**31**	**10.3**	**30.9**	**24.4**	**7**	**32.8**	**29.4**	**12.1**

different metrics: the path weight, the edge weight (max, min and average), edge use (max, min and average), the sum of the in and out degrees or the vertex degree (the maximum degree value only since the min and average values were zero in all cases).

In terms of the path weight, this is the most natural method for accessing the paths found by the algorithms. It is clear that GA found better paths than ROLS did with the mean

Table III
RESULTS FOR ROLS: NUMBER OF THE TOP 100 RANKED PATHS THAT PARTIALLY MATCHED GOLD STANDARD PATHWAYS

Run	Path weight	Max. edge weight	Avg. edge weight	Min. edge weight	Max. edge use	Avg. edge use	Min. edge use	Max. de-gree
1	7	6	7	5	3	14	24	5
2	27	8	27	16	5	34	33	9
3	29	11	29	19	8	37	33	14
4	17	10	16	17	5	20	16	8
5	28	11	28	19	8	40	62	14
6	35	8	36	32	5	28	16	9
7	38	13	38	34	8	40	30	14
8	17	3	17	14	3	16	33	5
9	39	10	39	31	8	40	30	14
10	22	10	22	14	9	35	37	15
11	34	11	34	26	8	37	30	14
12	34	5	34	33	9	20	33	13
13	29	11	29	19	9	40	33	14
14	33	5	33	33	7	20	33	11
15	28	11	28	16	9	37	33	14
16	38	13	38	33	9	40	30	14
17	36	12	36	33	8	40	30	14
18	16	3	16	14	2	16	33	3
19	39	10	38	33	8	40	30	14
20	29	11	29	15	8	40	33	14
MEAN	**28.8**	**9.1**	**28.7**	**22.8**	**7**	**31.7**	**31.6**	**11.6**

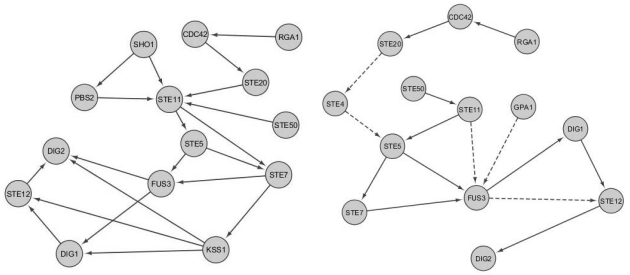

Figure 7. Pathways that are completely (left) or partially (right) contained within a known gold standard pathway.

of 31 in comparison to that of 28.8. Further, GA was able to find up to 50 paths that match the criterion (The best of ROLS is 39). This finding is backup by other metrics, such as the edge weight, edge use or vertex degree, which showed the better results of GA than that of ROLS. Note that for the edge use, reflecting the number of uses for a single edge is the number of times that edge is a member of satisfied paths, although it does not directly incorporate the edge or path weights, it still influences the top-ranked paths when sorting by edge use because edge use is dependent on the network orientation, which is dependent on the path weights. In Figure 7, we demonstrated several paths found by GA.

V. CONCLUSION

In this paper, we propose the genetic algorithm design for problem of orienting protein interaction network. This is a challenging problem for computational biology. We present a method to perform populations individuals that fit the

problem, especially our designs take into account conflicting elements for solution representation, thus greatly improve computing speed. Results show that our algorithm properly settles this problem. As evidence of the correctness of our algorithm, we find that our algorithm has reconstructed many known signaling pathways, which is significant in biological research. In the future, we will consider introducing more biological characteristics of the problems in the design process.

REFERENCES

[1] E. Segal, M. Shapira, A. Regev, D. Peer, D. Botstein, D. Koller, and N. Friedman, *Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data*. Nat. Genet. 2003.

[2] A.A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Dalla Favera, and A. Califano, *ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context*. BMC Bioinformatics. 2006.

[3] Junji Kitagawa, and Hitoshi Iba, *Identifying Metabolic Pathways and Gene Regulation Networks with Evolutionary Algorithms*. Chapter 12. Evolution Computation in Bioinformatic. 2003.

[4] E. Fischer, and U.Sauer, *Large-scale in vivo flux analysis shows rigidity and suboptimal performance of Bacillus subtilis metabolism*. Nat. Genet. 2005.

[5] J. Scott, T. Ideker, R.M. Karp, and R. Sharan, *Efficient algorithms for detecting signaling pathways in protein interaction networks*. J. Comput. Biol. 2006.

[6] G. Bebek, and J. Yang, *PathFinder: mining signal transduction pathway segments from protein-protein interaction networks*. BMC Bioinformatics. 2007.

[7] T.S. Mikkelsen, M. Ku, D.B. Jaffe, B. Issac, E. Lieberman, G. Giannoukos, P. Alvarez, W. Brockman, T. Kim, and R.P. Koche, *Genome-wide maps of chromatin state in pluripotent and lineage-committed cells*. Nature. 2007.

[8] B.P. Lewis, C.B. Burge, , and D.P. Bartel, *Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are MicroRNA targets*. Cell. 2005.

[9] X. Xie, J. Lu, E.J. Kulbokas, T.R. Golub, V. Mootha, K. Lindblad-Toh, E.S. Lander, , and M. Kellis, *Systematic discovery of regulatory motifs in human promoters and 3 UTRs by comparison of several mammals*. Nature. 2005.

[10] S.J. Cox, S.S. Levanon, G.N. Bennett, and K. San, *Genetically constrained metabolic flux analysis*. Metab. Eng. 2005.

[11] Anthony Gitter, Judith Klein-Seetharaman, Anupam Gupta, and Ziv Bar-Joseph, *Discovering Pathways by Orienting Edges in Protein Interaction Networks*. Nucleic Acids Research, Vol. 39, No. 4. 2011.

[12] Jinghua Gu, Bradley, Jianhua Xuan, Chen Wang, and Li Chen, *Detecting aberrant signal transduction pathways from high-throughput data using GIST algorithm*. Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2012.

[13] Dima Blokh, Danny Segev, and Roded Sharan, *Approximation Algorithms and Hardness Results for Shortest Path Based Graph Orientations*. Springer Berlin Heidelberg, 2012.

[14] R. Kohli, R. Krishnamurti, and P. Mirchandani, *The minimum satisfiability problem*. SIAM J. Discret. Math, 1994.

[15] M. Charikar, K. Makarychev, and Y. Makarychev, *Near-optimal algorithms for maximum constraint satisfaction problems*. ACM Trans. Alg, 2009.

[16] A. Medvedovsky, V. Bafna, U. Zwick, and R. Sharan, *An algorithm for orienting graphs based on cause-effect pairs and its applications to orienting protein networks*. InProceedings of the 8th international workshop on Algorithms in Bioinformatics, Karlsruhe, Germany, 2008.

[17] T. Back, *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.

[18] Anthony Gitter, Judith Klein-Seetharaman, Anupam Gupta, and Ziv Bar-Joseph, *Supporting Information, Discovering Pathways by Orienting Edges in Protein Interaction Networks,* . download from http://sb.cs.cmu.edu/OrientEdges/. [Accessed 10/8/2013]