Discovery of pathways in protein-protein interaction networks using a genetic algorithm

Nguyen Hoai Anh, Vu Cong Long, Tu Minh Phuong, Bui Thu Lam

PII: S0169-023X(15)00013-0
DOI: doi: 10.1016/j.datak.2015.04.002
Reference: DATAK 1503

To appear in: *Data & Knowledge Engineering*

DISCOVERY OF PATHWAYS IN PROTEIN-PROTEIN INTERACTION NETWORKS USING A GENETIC ALGORITHM

NGUYEN Hoai Anh[a], VU Cong Long[a], TU Minh Phuong[b], BUI Thu Lam[a]

[a]*Faculty of Information Technology - Le Quy Don Technical University, Ha Noi, Viet Nam*
[b]*Department of Computer Science - Posts and Telecommunications Institute of Technology, Ha Noi, Viet Nam*

Abstract

Biological pathways have played an important role in understanding cell activities and evolution. In order to find these pathways, it is neccessary to orient protein-protein interactions, which are usually given in forms of undirected networks or graphs. Previous findings indicate that orienting protein interactions can improve the process of 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.

This paper proposes a method for orienting protein-protein interaction networks (PPIs) and discovering pathways. For our proposal, the mathematical model of the problem is given and then a genetic algorithm is designed to find the solution for the problem taking into account the problem's characteristics. We conducted multiple runs on the data of yeast PPI networks to test the best option for the problem. The obtained results were compared with a wellknown algorithm (ROLS), which was shown to the best in dealing with this problem, in terms of the run time, fitness function values, and especially the ratio of matching gold standard pathways. The results show the good performance of our approach in addressing this problem.

Email addresses: nguyenhoaianh@yahoo.com (NGUYEN Hoai Anh), long.vucong@gmail.com (VU Cong Long), phuongtm@ptit.edu.vn (TU Minh Phuong), lam.bui07@gmail.com (BUI Thu Lam)

Preprint submitted to DKE journal on KSE 2013 *April 9, 2015*

Keywords: genetic algorithms; protein; interaction; network;

1. Introduction

Recently, there is a great interest in PPI databases, the source of interaction information for case studies in bioinformatics, being aggregated over time from the experimental findings. Given the large amount of PPI data collected, a challenging problem is to get biological insights, in particular to discover biological pathways from the data. Note that edges representing PPIs have been experimentally defined and tested. Certainly the reconstruction of the biological processes of cell (pathway or networks) has attracted a lot of attentions: the reconstruction of regulatory networks [1, 2, 3, 4, 5], the analysis of metabolic networks [6, 7, 8, 9], and the discovery of signaling networks and pathways [10, 11, 12, 13]. However, directionality of interactions in networks has not been thoroughly investigated, while direction is essential in finding how information is moved from one to another. The orientation of the signaling network is more difficult than the regulatory and metabolic networks, due to the lack of orientation information. For example, orientation of gene regulatory network is often determined by transcription factors regulating genes, studies of microRNAs often look for targets and motif studies are implemented upstream of genes [14, 15, 16]. Similarily, metabolic networks are modeled by knowledge about the order of genes and enzymes [17]. Meanwhile, 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 [12, 18, 19] underlining the need for finding an efficient algorithm for edge-orientation in PPI networks, which has been indentified as an NP-hard problem.

In [12], authors 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 was consistent with the experimental or not. The results were also compared with several algorithms proposed in [20, 21, 22]. When 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.

In the framework of this paper, we extended further our preliminary results on PPI edge orienting [23]. 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 [24], especially for dealing with NP-hard problems. Along with other intelligent computing techniques such as fuzzy computing, neural networks and multi-agent systems, GAs develop more and more strongly and are widely applied in different fields [25, 26]. Our GA design takes into account conflicting elements in PPI networks in order to reduce unnecessary edges, thus greatly improves computing speed. We examined different aspects including running time and objective values. Results showed that our algorithm found a good solution for the problem and this finding was supported by comparison to other algorithms' results. Especially, we answered the question of what is the meaning of the obtained pathways by extending biological validation.

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.

2. Background

2.1. Problem of orienting edges in protein interaction networks

Proteins are important components in the cell 's structure. They are involved in most of biological processes. During cell functioning, they interact with each other or with macromolecules such as DNA and RNA. They together form a complex network of interactions to perform biological functions. An example is given in Figure 1 where the graph shows a part of the network of protein interactions in yeast created by the *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. In order to support 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.

With signaling pathways verified by experiment, they are gathered into a database to serve for the interpretation of biological problems. The discovery of the signaling pathway in protein interaction networks is 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 for making predictions about new signaling pathways for purposes of biological studies such as understanding disease signals [27], creating new drugs to treat diseases caused by the deviation from the signal pathway.

This is a difficult problem because there exist many linking paths between two proteins in the interaction network. However, we can establish assumptions to simplify the problem. Firstly, we can assume that biological responses are controlled by reasonably short signaling cascades, so we can have a limit length of the path, which is called length-bounded paths. So far, pathways in signaling 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 [12]. The goal of the problem is to extract the signal pathways of length k from a source to a target that are highy reliable. Second, we can calculate the reliability of each interaction database PPIs, then only use the interactions that have high reliability to have better pathways [12]. Finally, in case of existing many pathways linking sources and targets, we will choose the the direction that creates a better overall network [12].

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. With a pair of $u, v \in V$, we have 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. All vertices and edges in the graph have weights which are denoted $w(v)$ and $w(e)$ respectively. While all vertices (proteins) have the same weight in our current implementation, allowing for varying protein weights is a useful feature in cases where some proteins are known to be involved in the signal transmission. Edge weight is a value in the range $[0, 1]$, which is based on the probability of each protein interaction. A path has a maximum length of at most k between pairs of sources - targets in form of $\langle s_i, t_i \rangle$, where $s_i \in S \subseteq V$

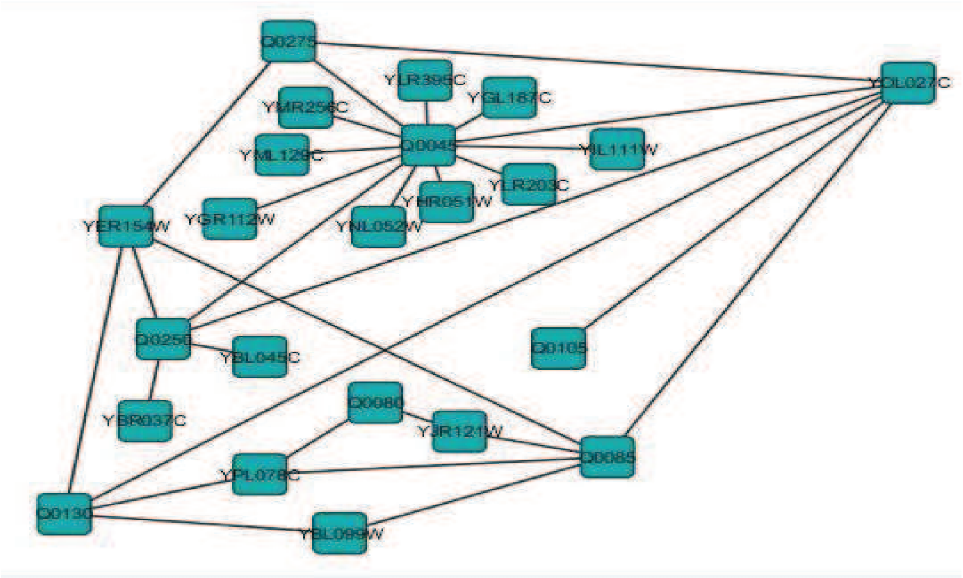

Figure 1: A part of the protein interaction network of yeast includes 23 proteins and 30 interactions. According to database BIOGRID network 2.0.51 of yeast, its PPIs have 5570 proteins and 140849 interactions [18].

and $t_i \in T \subseteq V$. 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 $\langle s_i, t_i \rangle$. 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

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

The goal is to orient the edge $e = (u, v) \in E$ from u to v or from v to u. 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 from v_j to v_{j+1} in the network. 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.

2.2. An Overview of Genetic Algorithms

Genetic algorithm (GA) is one of main streams 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 an heuristic has made the genetic algorithm as an effective approach for the optimization problems (with acceptable solutions) without the need of using conventional conditions (i.e continuity or differentiability) as prerequisites [28].

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 $\mathit{pop}(t) = \{x_1, x_2, \ldots, x_N\}$, N is the population size.
- ***Step 2***: Evaluate $\mathit{pop}(t)$.
- ***Step 3***: Create the mating pool $MP = se\{\mathit{pop}(t)\}$ with se is the selection operator (2.2.2).
- ***Step 4***: Define $\mathit{pop}'(t)=cr\{MP\}$, with cr is the crossover operator (2.2.3).
- ***Step 5***: Define $\mathit{pop}''(t)=mu\{\mathit{pop}'(t)\}$, with mu is mutation operator (2.2.4).
- ***Step 6***: Evaluate $\mathit{pop}''(t)$
- ***Step 7***: Define $\mathit{pop}(t + 1)=\mathit{pop}''(t)$ and set $t = t + 1$
- ***Step 8***: Go back Step 3, if the stopping criterion is not satisfied.

Where $\mathit{pop}(t)$ is the original population at time t, $\mathit{pop}(t)$ is the population after using crossover operator, $\mathit{pop}(t)$ is the population after using mutation operator.

2.2.1. 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's element represents a parameter of individual components.

2.2.2. 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.

2.2.3. 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.

2.2.4. 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.

3. Methodology

The main idea is to design a genetic algorithm to tailor the orientation problem characteristics making the search process effective. It 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 represents the orientation.

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

3.1. 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:

- Since 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 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. Among found pathways, there are edges with a single orientation and edges with different orientations. The edges with different orientations are called conflicting edges. 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 conflicting edges, the question is how to orient them?

Thus the initial population P is the set of orientation possibilities for conflicting 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:
$p_1 = \{(proA, proC), (proC, proB)\}$
$p_2 = \{(proA, proC), (proC, proD), (proD, proE), (proE, proF)\}$
$p_3 = \{(proA, proC), (proC, proD), (proD, proE), (proE, proG)\}$
$p_4 = \{(proF, proE), (proE, proD), (proD, proC), (proC, proB)\}$

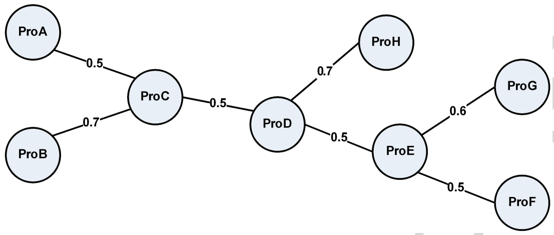

Figure 2: A weighted undirected graph representing a network of protein interactions.

Table 1: 16 individuals could be selected for the initial population P

C1	1	1	1	1
C2	1	1	1	0
C3	1	1	0	1
C4	1	1	0	0
C5	1	0	1	1
C6	1	0	1	0
C7	1	0	0	1
C8	1	0	0	0
C9	0	1	1	1
C10	0	1	1	0
C11	0	1	0	1
C12	0	1	0	0
C13	0	0	1	1
C14	0	0	1	0
C15	0	0	0	1
C16	0	0	0	0

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

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

So the 2^4 individuals are described in Table and each individual have 4 bits.

3.2. 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
$f(C1) = w(p_1) + w(p_2) + w(p_3) = 0.5*0.7 + 0.5*0.5*0.5*0.5 + 0.5*0.5*0.6 = 0.485$

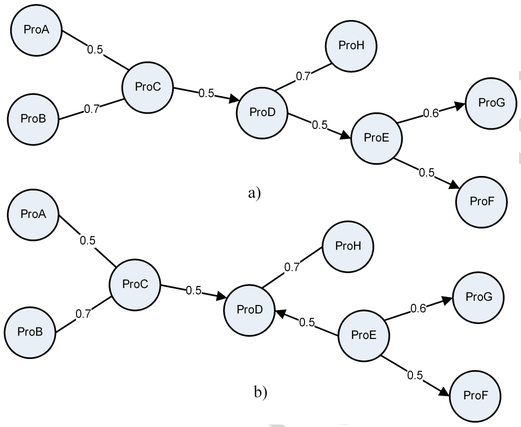

Figure 3: Orientation of conflicted edges in the graph G.

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 a better objective value than that of Case 2.

3.3. 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 use the binary tournament selection: 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 (as being 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 words, becoming parents for the next life).

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.

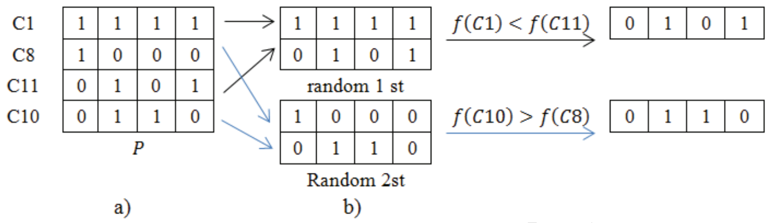

Figure 4: Operator selection in the populations P.

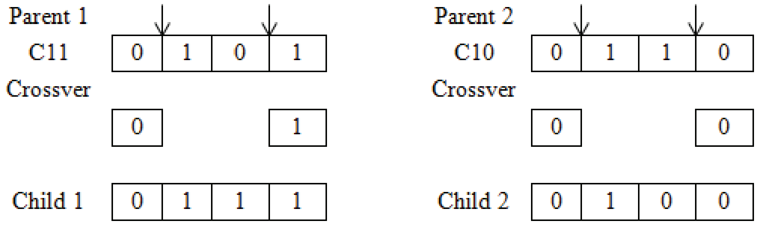

Figure 5: Simulation crossover operator between C11, C10 individuals.

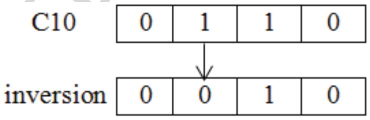

Figure 6: Simulation of mutation operator with individual C10.

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.

3.4. 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 for preservation. 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.

3.5. The complexity of the algorithm

Note that the computational cost of individual's evaluation is the same as what it is carried with Random Orientation Algorithm plus Local Search (ROLS). So, here we focus only on the computational time (or complexity of the algorithm) for GA framework. We call n is chromosome's length, m is the number of generations and s is the population size. Initializing the initial population requires the time of $O(n * s)$. The creation of the population for the next generation includes procedures to implement the following operators: time required for the selection operator $O(s/2)$, time required for the crossover operator $O(s*n)$, time required for the mutation operator $O(s*n)$, using localsearch to conserve time required elite individuals $O(s * n)$. Since these procedures are implemented one by one, the time required to create one next generation is equal to $max(O(s/2), O(s * n), O(n))$ and hence $O(s * n)$. Therefore the necessary time for creating next m generations is $O(m * s * n)$. According to our design n it is equal to the total number of conflicted edges in the set of conflicted edges and clearly it depends on the size of the actual PPI database. The general complexity of GA is $O(m * s * n)$.

4. Case studies

4.1. Prepare data

Yeast's interaction network

For experiments, we used the database of yeast PPIs taken from database BioGRID (`http://thebiogrid.org`). This is an online database of genetic interactions of organisms on a large scale. As mentioned above, this database is extensively 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 [29]. This database is a two-dimensional data table having 140849 lines; each line contains interactive information of a pair of 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 [29].

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

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

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. To ensure the reliability of the interaction, our GA only takes into consideration interactions with weights of 0.6 or larger, or of high reliability.

Gold standard pathways

To confirm that the orientations produced by our algorithm not only achieve good objective function values but also produce biologically meaningful results, we compared the PPI network of yeast that it oriented by our algorithm with all yeast signaling pathways from the Science Signaling Database of Cell Signaling. The database focuses all the signal path has been verified by experiment called the gold standard pathway. Collection of all the gold standard pathway is called the gold standard network. To evaluate our algorithm, we compared the overlap of the individual pathways in the set of pathways found by the algorithm with the gold standard network. We see that, the gold standard network is much smaller compared with the complete interaction network, containing 76 proteins and 122 interactions.

The source - target protein pairs

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 is determined basing on the standard pathway taken from [29].

4.2. Testing scenario

First, we use the Depth First Search algorithm for finding a set of paths from source to target, then generating a set of conflicted edges. The yeast PPIs database gives us 993 conflicting 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 (GA) designed by us with the results of the Random Orientation Algorithm plus Local Search, called ROLS, (Note that in [12], ROLS's performance was shown better than that of the algorithms MIN-SAT, MAX-CSP and MTO).

We also compared our results to a multistart random search (MRS) method making sure the effectiveness of GA. For MRS, a population of solutions are allowed to do random search without interaction (using genetic operators).

For both algorithms GA and MRS, we 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: 993 in this paper). Input parameters for GA include: total generation number of 50, crossover probability of 0.9 and mutation probability of 0.001, and each run populations and individuals are initialized randomly. There was no particular reason for choosing these values, we just followed the common settings in the field of evolutionry computation. To ensure the same experimental conditions for comparison, the algorithm ROLS's performance is based upon 20 random restarts and take the average value of the objective values.

4.3. Results and analysis

We analyze performance of GA from different aspects including runtime, objective values, and biological validation.

4.3.1. Algorithm's runtimes

For analytical methods used high-throughput data sets, the scalability is extremely important because current database is incomplete and interactive network of other organisms may be larger than the yeast. So we used interactive network of yeast to examine the runtimes of our orientation algorithm and ROLS for various combinations of maximum path length and source-target pairs. Table 2 presents the runtimes of the algorithms for various combinations of sources, targets and maximum path length (k) using a core i5, 2.4GHz machine with 4 GB of RAM. Times for ROLS and GA are averaged over these 20 runs.

For smaller instances, all algorithms were very fast, terminating in less than a second. In case k = 4, for all cases pairs of sources - targets, ROLS was slower than our designed GA. However, from the case of k = 5 and above, when the number of pairs of sources - targets increases, our GA was much faster runtime than ROLS. Typically, in the case 16 pairs of sources -

Table 2: Algorithm runtimes in seconds.

Sources	**Targets**	**k**	**ROLS**	**GA**
4	4	4	0.052	0.776
8	8	4	0.515	2.738
16	16	4	3.488	6.284
4	4	5	4.850	6.930
8	8	5	29.430	22.506
16	16	5	340.899	88.933
4	4	6	618.809	189.182
8	8	6	2773.108	554.495
16	16	6	23663.883	3137.720

targets, our GA ran 7 times faster than ROLS did (Table 2). This indicates that our algorithm worked more efficiently when the problem size increases.

4.3.2. 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 3). The average objective values obtained by GA are much better than ROLS's (7961.3 ± 52.09 comparing to 7836.6 ± 52.37). In most of runs, our GA's objective function value was higher than that of ROLS. Also GA found the best value of 8062, 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.

We also validated GA performance against MRS. Similar finding is also obtained in this case. GA still showed better performance than that of MRS. Genetic operators helped GA wokring effectively in this case studies.

4.3.3. Evaluation of the algorithm using gold standard pathways

Regarding the biological validation, we employed the number of standard pathways as a criterion to assess the ability of the algorithm in finding pathways. To forllow this criterion, we ranked all paths found by GA and ROLS according to different metrics and calculate how many of top 100 paths having exactly 5 edges (or 6 proteins) that are at least partially matched in the standard pathways. According to the criteria given in [12], 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

Table 3: The results of the best objective values obtained by MRS, GA and ROLS among 20 runs.

Seed	**MRS**	**GA**	**ROLS**
1	7715	8015	7739
2	7824	7951	7781
3	7855	7929	7818
4	7832	7946	7820
5	7821	7977	7886
6	7887	7979	7876
7	7873	7976	**7942**
8	7799	7916	7813
9	7889	7976	7817
10	7795	7877	7788
11	7859	7883	7781
12	7761	7894	7900
13	7869	8002	7806
14	7833	8005	7823
15	**7895**	7898	7935
16	7827	**8062**	7805
17	7801	8020	7896
18	7806	7951	7855
19	7827	7926	7835
20	7822	8043	7816
MEAN	**7829.5**	**7961.3**	**7836.6**
STD	**43.35**	**52.09**	**52.37**

results were listed in Tables 4, 5 and 6 for MRS, GA and ROLS and with 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 and more stable than MRS and ROLS did with the mean of 36.3 in comparison to that of 28.1 and 27.2 respectively. This finding is backup by other metrics such as the edge weight, edge use or vertex degree, which show better results

Table 4: Results for MRS: Number of the top 100 ranked paths that partially matched gold standard pathways

Seed	Path weight	Max. edge weight	Avg. edge weight	Min. edge weight	Max. edge use	Avg. edge use	Min. edge use	Max. degree
1	29	7	29	25	4	20	**41**	7
2	28	11	28	19	8	40	33	14
3	49	12	**51**	42	**12**	**45**	39	20
4	28	11	28	19	8	37	33	14
5	5	0	6	3	2	16	28	4
6	23	3	23	17	8	19	33	13
7	18	8	18	23	3	14	24	5
8	37	11	36	32	9	40	30	14
9	29	11	29	19	8	40	33	14
10	36	10	36	33	7	40	30	10
11	38	11	37	34	8	40	30	14
12	11	5	12	8	4	18	39	7
13	9	9	9	6	5	20	16	8
14	**51**	12	**51**	39	**12**	40	39	**20**
15	39	**14**	39	30	11	44	16	17
16	17	11	19	9	8	35	36	15
17	33	9	33	25	6	37	30	10
18	17	3	17	14	3	16	33	5
19	29	11	29	19	8	40	33	14
20	35	11	34	28	8	37	30	14
MEAN	**28.1**	**9**	**28.2**	**22.2**	**7.1**	**31.9**	**31.3**	**14**

in GA than in MRS and 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 this does not directly incorporate the edge or path weights, the top-ranked paths are still influenced when sorted by edge use because edge use is dependent on the network orientation, which is dependent on the path weights.

4.4. Discussion

We have observed the above sections that GA found many good pathways. However, we wonder if the new pathways, which are ranked high according

Table 5: Results for GA: Number of the top 100 ranked paths that partially matched gold standard pathways

Seed	Path weight	Max. edge weight	Avg. edge weight	Min. edge weight	Max. edge use	Avg. edge use	Min. edge use	Max. degree
1	38	11	37	34	8	**40**	30	14
2	29	11	29	19	8	37	33	14
3	38	11	38	32	8	**40**	39	14
4	35	11	35	27	8	37	39	14
5	34	12	34	27	8	37	30	14
6	38	**13**	37	31	8	**40**	30	14
7	38	10	37	34	8	**40**	30	14
8	35	11	34	27	8	37	30	14
9	35	11	35	28	8	37	30	14
10	39	11	38	34	8	**40**	39	14
11	35	11	34	27	8	37	30	14
12	28	7	27	18	5	28	16	8
13	35	11	34	27	8	37	30	14
14	**50**	12	**49**	**36**	**12**	**40**	**42**	**20**
15	33	5	33	32	8	20	33	13
16	35	11	35	27	8	37	30	14
17	39	**13**	38	33	8	**40**	30	14
18	38	11	37	33	8	**40**	30	14
19	35	12	36	33	8	**40**	30	14
20	38	11	38	34	8	**40**	30	14
MEAN	**36.3**	**10.8**	**35.8**	**29.7**	**8.1**	**37.2**	**31.6**	**14**

to the criteria, may biologically be correct and represent information that is missing from current PPI databases? Using the same analysis with [12], we divided the pathways predicted by our GA into three groups and analyzed the top 20 pathways in each group using the metric of path-weight for ranking. The first group (Figure 7A) contains pathways having five or six proteins whose its edges match with the gold standard pathways. The second group (Figure 7B) contains pathways having six proteins that have some edges coinciding with the gold standard pathways. With these pathways, we need to find out whether the new orientation interaction given by GA might mean-

Table 6: 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	39	13	39	34	8	**40**	**39**	14
2	17	3	17	14	3	16	33	5
3	34	13	35	33	8	**40**	30	14
4	19	**15**	20	11	8	35	36	15
5	34	11	34	24	8	37	30	14
6	38	10	37	33	8	**40**	30	14
7	**51**	11	**51**	**37**	**12**	**40**	30	**20**
8	8	5	8	14	1	4	16	2
9	38	11	38	31	8	39	30	14
10	17	3	17	14	3	16	33	5
11	9	8	9	7	4	20	16	6
12	29	11	29	19	8	**40**	33	14
13	17	3	17	14	3	16	33	5
14	38	11	37	33	8	**40**	30	14
15	39	10	38	33	9	**40**	30	14
16	17	11	18	9	8	35	36	15
17	37	11	37	32	8	**40**	30	14
18	12	4	12	20	1	4	16	2
19	29	11	29	18	8	37	33	14
20	21	11	21	25	5	20	16	8
MEAN	**27.2**	**9.3**	**27.2**	**22.8**	**6.5**	**30**	**29**	**11.2**

ingfully show extensions to the pathways that were not previously known or were not recorded in the databases. The third group (Figure 7C) are pathways discovered by our GA that do not consist with any known pathways in the gold standard network. With these pathways, we wondered whether they represent significant pathways in biology which have not been detected experimentally. We can also merge paths of three groups discovered by our GA into a larger network signals because each edge is oriented uniquely in all paths. This is necessary to form a signaling network in the cell.

In Figure 7A (representing pathways matched the gold standard ones),

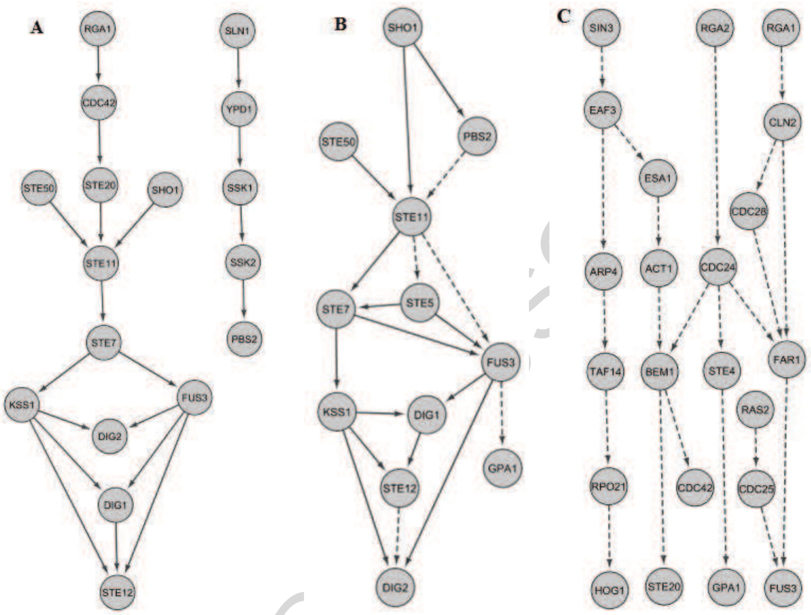

Figure 7: The top-ranked pathways discovered by the GA algorithm. Solid edges were present in the gold standard network and dashed edges were not exist or oriented reversing direction. (A) Pathways that its edges entirely coincide with the gold standard network. (B) Pathways that some of its edges coincide with the gold standard network and contain new edges as well. (C) Pathways that do not consist with any known pathways in the gold standard network.

the path Sln1 → Ypd1 → Ssk1 → Ssk22 → Pbs2 is inside of the high-osmolarity glycerol (HOG) pathway; it was also found in [12]. We also found that the pheromone pathway is filled with the cascade of Rga1 → Cdc42 → Ste20 → Ste11 → Ste7 → Fus3. The other paths that begin at Ste50 or Sho1 and extend to Dig1, Dig2, Ste7 and Ste12 are members of the filamentous growth pathway.

Figure 7B illustrates the partially-matched pathways found by our GA; they also contain the new orientation edges found in [12]. Some of these paths in the pheromone signaling pathway comprise the edge Ste11 → Ste5. This edge was considered a error because in the gold standard it was oriented reversely. Protein Ste5 performs function of scaffolding proteins, it interacts with Fus3, Ste7, Ste11 to form the active complex [30, 31, 12]. However, recently it has been discovered that protein Ste11 consists of a C-terminal kinase domain and three N-terminal regulatory domains, one of which interacts with Ste5 [32]. Thus, our predicted Ste11 → Ste5 edge is also valid.

Another predicted interaction that disagrees with the direction in the gold standard database is Fus3 → GPA1. The resulting Ste4-Ste18 dimer mediates signal transduction through binding to both the scaffolding protein Ste5 and the PAK kinase Ste20, causing activation of a MAP kinase cascade (Ste11, Ste7, and Fus3) [33]. However, according to recent research, it has been found that there is a feedback loop from Fus3 to GPA1 to Ste4, which is phosphorylated by Fus3 and negatively regulates the pathway [34]. Thus, our predicted Fus3 → GPA1 edge is also valid.

For pathways in Figure 7C, which do not overlap with any of the pathways in the databases of the gold standard network that we used, we found many edges that may be biological hypotheses. For example, edge orientation Cln2 → Cdc28. CLN2 encodes a G1 cyclin involved in regulation of the cell cycle [35]. This process required a periodic activation of cyclin-dependent kinases (CDKs), protein Cdc28 is one of them [36]. Thus, our predicted Cln2 → Cdc28 edge is also valid. Another example, edge orientation Bem1 → Ste20. Protein Bem1 binds protein Ste5 and Ste20, which are central components of the mating pathway, Bem1's role may be to connect the mating signal to the proteins that induce the appropriate changes to the actin cytoskeleton [37].

5. Conclusion

In this paper, we proposed the genetic algorithm design for problem of orienting protein interaction network. This is a challenging problem for computational biology. We presented 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.

[1] E. Segal, M. Shapira, A. Regev, D. Peer, D. Botstein, D. Koller, N. Friedman, Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data, Nat. Genet. 34 (2003) 166–176.

[2] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. D. Favera, A. Califano, Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context, BMC Bioinformatics 7 (2006) S7.

[3] M. Grzegorczyk, D. Husmeier, Improvements in the reconstruction of time-varying gene regulatory networks: dynamic programming and regularization by information sharing among genes, Bioinformatics 27 (2011) 693–699.

[4] D. A. Ravcheev, A. A. Best, N. V. Sernova, M. D. Kazanov, P. S. Novichkov, D. A. Rodionov, Genomic reconstruction of transcriptional regulatory networks in lactic acid bacteria, BMC Genomics 14 (2013) 14–94.

[5] G. xia Liu, W. Feng, H. Wang, L. Liu, C. guang Zhou, Reconstruction of gene regulatory networks based on two-stage bayesian network structure learning algorithm., Journal of Bionic Engineering 6 (2009) 86–92.

[6] J. Kitagawa, H. Iba, Identifying metabolic pathways and gene regulation networks with evolutionary algorithms, Evolution Computation in Bioinformatic (2003) 255–275.

[7] E. Fischer, U.Sauer, Large-scale in vivo flux analysis shows rigidity and suboptimal performance of bacillus subtilis metabolism, Nat. Genet. 37 (2005) 636–640.

[8] E. Ruppin, J. A. Papin, L. F. de Figueiredo, S. Schuster, Metabolic reconstruction, constraint-based analysis and game theory to probe genome-scale metabolic networks, Current Opinion in Biotechnology 21 (2010) 502–510.

[9] D. McCloskey, B. . Palsson, A. M. Feist, Basic and applied uses of genome-scale metabolic network reconstructions of escherichia coli, Molecular Systems Biology 9.

[10] J. Scott, T. Ideker, R. Karp, R. Sharan, Efficient algorithms for detecting signaling pathways in protein interaction networks, J. Comput. Biol. 13 (2006) 133–144.

[11] G. Bebek, J. Yang, Pathfinder: mining signal transduction pathway segments from protein-protein interaction networks, BMC Bioinformatics 8 (2007) 335.

[12] A. Gitter, J. Klein-Seetharaman, A. Gupta, Z. Bar-Joseph, Discovering pathways by orienting edges in protein interaction networks, Nucleic Acids Research 39 (2011) e22.

[13] T. Umezawa, N. Sugiyama, F. Takahashi, J. C. Anderson, Y. Ishihama, S. C. Peck, K. Shinozaki, Genetics and phosphoproteomics reveal a protein phosphorylation network in the abscisic acid signaling pathway in arabidopsis thaliana, Sci. Signal. 6.

[14] T. Mikkelsen, M. Ku, D. Jaffe, B. Issac, E. Lieberman, G. Giannoukos, P. Alvarez, W. Brockman, T. Kim, R. Koche, Genome-wide maps of chromatin state in pluripotent and lineage-committed cells, Nature. 448 (2007) 553–560.

[15] B. Lewis, C. Burge, D. Bartel, Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microrna targets, Cell. 120 (2005) 15–20.

[16] X. Xie, J. Lu, E. Kulbokas, T. Golub, V. Mootha, K. Lindblad-Toh, E. Lander, M. Kellis, Systematic discovery of regulatory motifs in human promoters and 3 utrs by comparison of several mammals, Nature. 434 (2005) 338–345.

[17] S. Cox, S. Levanon, G.N.Bennett, K. Y. San, Genetically constrained metabolic flux analysis, Metab. Eng. 7 (2005) 445–456.

[18] J. Gu, J. Xuan, C. Wang, L. Chen, T. L. Wang, L. M. Shih, Detecting aberrant signal transduction pathways from high-throughput data using gist algorithm, Computational Intelligence in Bioinformatics and Computational Biology (CIBCB) (2012) 267–274.

[19] D. Blokh, D. Segev, R. Sharan, Approximation algorithms and hardness results for shortest path based graph orientations, Lecture Notes in Computer Science 7354 (2012) 70–82.

[20] R. Kohli, R. Krishnamurti, P. Mirchandani, The minimum satisfiability problem, SIAM J. Discret. Math 7 (1994) 275–283.

[21] M. Charikar, K. Makarychev, Y. Makarychev, Near-optimal algorithms for maximum constraint satisfaction problems, ACM Trans. Alg 5 (2009) 1–14.

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

[23] N. H. Anh, V. C. Long, T. M. Phuong, B. T. Lam, A genetic-based approach for discovering pathways in protein-protein interaction networks, in: InProceedings of SoCPaR2013, [Accepted], 2013.

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

[25] L. Araujo, H. Zaragoza, J. R. Prez-Agera, J. Prez-Iglesias, Structure of morphologically expanded queries: A genetic algorithm approach., Data & Knowledge Engineering 69 (2010) 279–289.

[26] H. Liu, C. Blouin, V. Keelj, Sentence identification of biological interactions using patricia tree generated patterns and genetic algorithm optimized parameters., Data & Knowledge Engineering 69 (2010) 137–152.

[27] W. Fu, B. Sanders-Beer, K. Katz, D. Maglott, K. Pruitt, R. Ptak, Human immunodeficiency virus type 1, human protein interaction database at ncbi, Nucleic Acids Res 37 (2009) 417–422.

[28] R. Bueno, A. J. Traina, C. T. Jr, Genetic algorithms for approximate similarity queries., Data & Knowledge Engineering 62 (2007) 459–482.

[29] A. Gitter, J. Klein-Seetharaman, A. Gupta, Z. Bar-Joseph, Supporting information, discovering pathways by orienting edges in protein interaction networks, http://sb.cs.cmu.edu/OrientEdges/ (2012).

[30] E. A. Elion, The ste5p scaffold, J Cell Sci 114 (2001) 3967–3978.

[31] C. Inouye, N. Dhillon, T. Durfee, P. Zambryski, J. Thorner, Mutational analysis of ste5 in the yeast saccharomyces cerevisiae: application of a differential interaction trap assay for examining protein-protein interactions., Genetics 147 (1997) 479–492.

[32] L. Bardwell, A walk-through of the yeast mating pheromone response pathway, Peptides 25 (2004) 1465–1476.

[33] S. J. Dowell, A. L. Bishop, S. L. Dyos, A. J. Brown, M. S. Whiteway, Mapping of a yeast g protein betagamma signaling interaction, Genetics 150 (1998) 1407–1417.

[34] M. Metodiev, D. Matheos, M. Rose, D. Stone, Regulation of mapk function by direct interaction with the mating-specific galpha in yeast, Science 296 (2002) 1483–1486.

[35] D. Huang, S. Kaluarachchi, D. van Dyk, H. Friesen, R. Sopko, W. Ye, N. Bastajian, J. Moffat, H. Sassi, M. Costanzo, B. Andrews, Dual regulation by pairs of cyclin-dependent protein kinases and histone deacetylases controls g1 transcription in budding yeast., PLoS Biol 7 (2009) e1000188.

[36] M. Miller, F. Cross, A. Groeger, K. Jameson, Identification of novel and conserved functional and structural elements of the g1 cyclin cln3 important for interactions with the cdk cdc28 in saccharomyces cerevisiae., Yeast 22 (2005) 1021–1036.

[37] T. Leeuw, A. Fourest-Lieuvin, C. Wu, J. Chenevert, K. Clark, M. Whiteway, D. Thomas, E. Leberer, Pheromone response in yeast: association of bem1p with proteins of the map kinase cascade and actin, Science 270 (1995) 1210–1213.