

# IDENTIFYING CRITICAL NODES IN PROTEIN-PROTEIN INTERACTION NETWORKS

VLADIMIR BOGINSKI AND CLAYTON W. COMMANDER

**ABSTRACT.** In recent years, the study of biological networks has increased dramatically. These problems have piqued the interest of researchers in many disciplines from biology to mathematics. In particular, many problems of interest to biological scientists can be modeled as combinatorial optimization problems and studied by operations researchers. In this chapter, we consider the problem of identifying the critical nodes of a network and its potential applications to protein-protein interaction networks. More specifically, we are interested in determining the smallest set of nodes whose removal from the graph maximally disconnects the network. Recent techniques for identifying critical nodes in telecommunication networks are applied to the study of protein-protein interaction graphs and the results are analyzed.

## 1. INTRODUCTION

Optimization problems abound in the study of biological networks. This is a timely research topic and has been the focus of great attention in the recent literature [1, 4, 6, 19, 21, 22, 23, 27]. In this chapter, we investigate the detection of *critical nodes* in *protein-protein interaction networks*. The CRITICAL NODE DETECTION PROBLEM (CNDP) is a combinatorial optimization problem recently introduced by Arulselman et al. [2]. Given a graph  $G = (V, E)$  and an integer  $k \in \mathbb{Z} \setminus |V|$ , the objective is to determine a subset  $A \subseteq V$ , such that  $|A| = k$ , whose deletion from the graph results in a minimum cohesion and ensures a minimum difference in the sizes of the components. A related problem, the CARDINALITY CONSTRAINED CNDP (CC-CNDP) seeks to determine a minimum cardinality subset of nodes whose deletion ensures that the number of nodes reachable from any other node in the network does not exceed some threshold value.

The CNDP has applications in many fields including social network analysis, quality assurance and risk management in telecommunication networks, transportation science, and control of social contagion [2, 3, 7]. Our proposition is that identifying the critical nodes in protein-protein interaction networks can have applications in computational biology, in particular in drug design. The interpretation of the critical nodes in the context of protein-protein interaction networks is that these nodes represent a minimum cardinality set of proteins whose removal would destroy the primary interactions and thus help neutralize potentially harmful organisms (e.g., bacteria or viruses).

The organization of this chapter is as follows. In the next section, protein-protein interaction networks are discussed. In Section 3, we provide mathematical programming formulations for both variants of the CRITICAL NODE DETECTION PROBLEM described above. In Section 4, we discuss the implementation of several heuristics for both problems, and provide some preliminary computational results of critical node detection on

---

To appear in *Clustering Challenges in Biological Networks*, S. Butenko, W.A. Chaovilitwongse, and P.M. Pardalos (editors), Springer, 2008.

real-world protein-protein interaction networks in Section 5. Conclusions and future directions of research are identified in Section 6.

## 2. PROTEIN-PROTEIN INTERACTION NETWORKS

In recent years, the biological research community's interest in studying proteins from different aspects has steadily increased. As a result, the field of *proteomics*, which investigates proteins' structures and functions, has been developed. In particular, *protein-protein interactions* have been extensively studied using various advanced techniques. [5, 14, 15, 26, 28] Many biological functions involve interactions between proteins at different levels, including signal transduction in cells (i.e., conversion of one kind of a signal to another inside a cell, which may play an important role in biological processes, including disease development), formation of protein complexes (i.e., stable over time structures involving multiple proteins), brief interactions between proteins involving the processes of modification of one protein by another, etc.

Protein-protein interactions can be represented in terms of graph theory as a set of vertices (proteins) and edges (certain types of interactions between proteins). These structures are referred to as *protein-protein interaction networks*. These networks play an important role in computational biology. In many cases, they can be easily visualized and are convenient for understanding the complex nature of different types of interactions between proteins. As a result of extensive research efforts in studying protein-protein interactions for different biological organisms (e.g., certain types of bacteria), massive amounts of data have been obtained. In particular, detailed and comprehensive data on protein-protein interactions is available for the yeast *Saccharomyces cerevisiae*, which has been considered in a number of works.[17, 16, 25]

Moreover, protein-protein interactions are studied from the perspective of *drug design* applications. [11, 24] In particular, drugs that target specific types of proteins can be developed. This research direction has significantly grown recently in the context of identifying *target proteins* responsible for certain diseases based on the available protein-protein interaction data. Nowadays, experimental studies in this area are extensively conducted by scientists in the pharmaceutical industry and research communities. [20]

On the other hand, protein-protein interaction networks can be investigated from the network optimization perspective. In this chapter, we make the first attempt to put the aforementioned problem of identifying target proteins in protein-protein interaction networks in the framework of combinatorial optimization. Specifically, we propose to apply the recently introduced CRITICAL NODE DETECTION PROBLEM (CNDP) to analyze protein-protein interactions and detect the nodes (proteins) that are the most important for the connectivity of these networks. We believe that identifying these critical nodes can provide information that can be used in drug design and other applications.

Next, we discuss mathematical programming formulations of the considered problems and present computational results obtained for some available protein-protein interaction datasets.

## 3. OPTIMIZATION APPROACHES FOR CRITICAL NODE DETECTION

Denote a graph  $G = (V, E)$  as a pair consisting of a set of vertices  $V$ , and a set of edges  $E$ . All graphs in this chapter are assumed to be undirected and unweighted. For a subset  $W \subseteq V$ , let  $G(W)$  denote the subgraph induced by  $W$  on  $G$ . A set of vertices  $I \subseteq V$  is called an *independent* or *stable set* if for every  $i, j \in I$ ,  $(i, j) \notin E$ . That is, the graph  $G(I)$  induced by  $I$  is edgeless. An independent set is *maximal* if it is not a subset of any

larger independent set (i.e., it is maximal by inclusion), and *maximum* if there are no larger independent sets in the graph.

**3.1. The Critical Node Detection Problem.** Given a graph  $G = (V, E)$ , let  $u : V \times V \mapsto \{0, 1\}$ , where  $u_{ij} = 1$  if nodes  $i$  and  $j$  are in the same component of  $V$ . Then the objective of the CNDP is to find a subset  $A \subseteq V$  of nodes such that  $|A| \leq k$ , whose deletion results in the minimum value of  $\sum u_{ij}$  in the edge induced subgraph  $G(V \setminus A)$ . This objective function results in a minimum cohesion in the network, while also ensuring a minimum difference in the sizes of the components. An integer programming formulation of the CNDP has been formulated by Arulselman et al. [2]

Let  $u$  be defined as above and define  $v : V \mapsto \{0, 1\}$  as

$$v_i := \begin{cases} 1, & \text{if node } i \text{ is deleted in the optimal solution,} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Then the CRITICAL NODE DETECTION PROBLEM is given as

$$\text{(CNDP) Minimize} \quad \sum_{i,j \in V} u_{ij} \quad (2)$$

s.t.

$$u_{ij} + v_i + v_j \geq 1, \quad \forall (i, j) \in E, \quad (3)$$

$$u_{ij} + u_{jk} - u_{ki} \leq 1, \quad \forall (i, j, k) \in V, \quad (4)$$

$$u_{ij} - u_{jk} + u_{ki} \leq 1, \quad \forall (i, j, k) \in V, \quad (5)$$

$$-u_{ij} + u_{jk} + u_{ki} \leq 1, \quad \forall (i, j, k) \in V, \quad (6)$$

$$\sum_{i \in V} v_i \leq k, \quad (7)$$

$$u_{ij} \in \{0, 1\}, \quad \forall i, j \in V, \quad (8)$$

$$v_i \in \{0, 1\}, \quad \forall i \in V. \quad (9)$$

Constraints (3) ensure that if  $(i, j) \in E$  and nodes  $i$  and  $j$  are in separate components, then one or both of them is deleted. The set of constraints (4-6) ensure that if nodes  $i$  and  $j$  are in the same component and nodes  $j$  and  $k$  are in the same component, then necessarily  $i$  and  $k$  belong to the same component. Finally, (7) constrains the maximum number of nodes to be deleted. The CNDP was shown to be  $\mathcal{NP}$ -hard[12] by a reduction of MAXIMUM INDEPENDENT SET to an instance of the CNDP[3].

**3.2. Cardinality Constrained Problem.** Given a graph  $G = (V, E)$ , the *connectivity index* of a node is defined as the number of nodes reachable from that vertex (see Figure 1 for examples). To constrain the network connectivity in optimization models, we can impose constraints on the connectivity indices of the nodes [8].

The CARDINALITY CONSTRAINED CNDP can be formulated in a similar manner to the the CNDP above. Recall that in this problem, we are given an integer  $L \in \mathbb{Z}$ , and we are interested in determining a minimum cardinality subset  $A \subseteq V$  such that the *connectivity index* of the remaining nodes in the vertex deleted subgraph  $G(V \setminus A)$  does not exceed  $L$ . Using the same definition of the variables as in the previous subsection, we can formulate

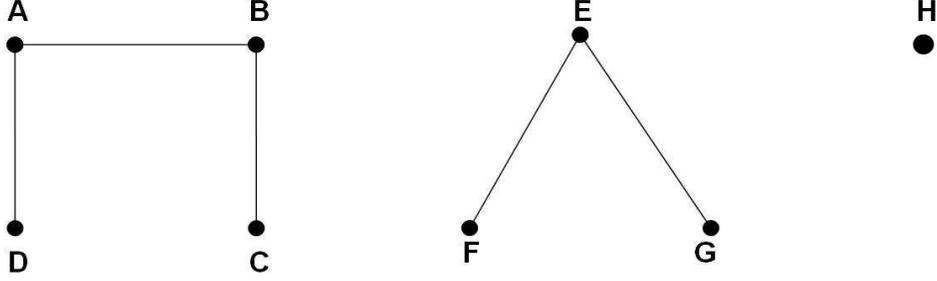


FIGURE 1. Connectivity Index of nodes A,B,C,D is 3. Connectivity Index of E,F,G is 2. Connectivity Index of H is 0.

the CC-CNDP as the following integer linear programming problem.

$$\text{(CC-CNDP) Minimize } \sum_{i \in V} v_i \quad (10)$$

s.t.

$$u_{ij} + v_i + v_j \geq 1, \forall (i, j) \in E, \quad (11)$$

$$u_{ij} + u_{jk} - u_{ki} \leq 1, \forall (i, j, k) \in V, \quad (12)$$

$$u_{ij} - u_{jk} + u_{ki} \leq 1, \forall (i, j, k) \in V, \quad (13)$$

$$-u_{ij} + u_{jk} + u_{ki} \leq 1, \forall (i, j, k) \in V, \quad (14)$$

$$\sum_{i, j \in V} u_{ij} \leq L, \quad (15)$$

$$u_{ij} \in \{0, 1\}, \forall i, j \in V, \quad (16)$$

$$v_i \in \{0, 1\}, \forall i \in V, \quad (17)$$

where  $L$  is the maximum allowable connectivity index for any node in the vertex deleted subgraph  $G(V \setminus A)$ . Notice that the objective is to minimize the number of nodes deleted. Constraints (11) follow exactly as in the CNDP model. Also, Constraints (12)–(14) are equivalent to the constraint  $u_{ij} + u_{jk} + u_{ki} \neq 2, \forall (i, j, k) \in V$ , which ensures that if nodes  $i$  and  $j$  are in the same component and nodes  $j$  and  $k$  are in the same component, then necessarily  $i$  and  $k$  belong to the same component (as in the CNDP model). Constraints (15) ensure that the connectivity indices of all nodes does not exceed  $L$ . The CC-CNDP is also  $\mathcal{NP}$ -hard as proved by Arulselman et al[3]. For the application of interest in this chapter, the CC-CNDP is the most applicable critical node optimization model. Therefore as we consider solutions approaches and computational experiments in the following sections, this is the problem on which we will focus.

#### 4. HEURISTIC APPROACHES FOR CRITICAL NODE DETECTION

Due to the computational complexity of the mathematical programming formulations presented above, the prior work in this area has concentrated on the development of efficient heuristics for critical node detection problems[2, 3, 7].

**4.1. Multi-start Combinatorial Heuristic.** Arulselman et al.[3] have proposed an efficient combinatorial heuristic for the CNDP. Pseudo-code for the proposed algorithm is provided in Figure 2. The heuristic starts off by identifying a maximal independent set

```

procedure CriticalNode( $G, L$ )
1  MIS  $\leftarrow$  MaximalIndepSet( $G$ )
2  OPT  $\leftarrow$  FALSE
3  NoAdd  $\leftarrow$  0
4  while (OPT .NOT.TRUE) do
5    for ( $i = 1$  to  $|V|$ ) do
6      if ( $\frac{|s|(|s|-1)}{2} \leq L \forall s \in S \subseteq G(\text{MIS} \cup \{i\}) : i \in V \setminus \text{MIS}$ ) then
7        MIS  $\leftarrow$  MIS  $\cup$   $\{i\}$ 
8      else
9        NoAdd  $\leftarrow$  NoAdd + 1
10     end if
11     if (NoAdd =  $|V| - |\text{MIS}|$ ) then
12       OPT  $\leftarrow$  TRUE
13       BREAK
14     end if
15   end for
16 end while
17 return  $V \setminus \text{MIS}$  /* set of nodes to delete */
end procedure CriticalNode

```

FIGURE 2. Combinatorial heuristic for the CARDINALITY CONSTRAINED CRITICAL NODE PROBLEM.

(MIS). This array holds the set of non-critical nodes. In the main loop from lines 4-16, the procedure iterates through the vertices and determines which of them can be added back to the graph while still maintaining feasibility. If vertex  $i$  can be replaced, MIS is augmented to include  $i$  in step 7, otherwise NoAdd is incremented. When NoAdd is equal to  $|V| - |\text{MIS}|$ , then no nodes can be returned to the graph and OPT is set to TRUE. The loop is then exited and the algorithm returns the set of critical nodes, i.e.,  $V \setminus \text{MIS}$ .

The solution found by this procedure is then further improved through a local search step and incorporated within a multi-start mechanism (see Figure 3). Computational testing was carried out on a benchmark network of the social interactions of the terrorists involved in the 9 \ 11 hijacking[18] and on some networks generated by the authors. Their tests indicated that the heuristic is very effective in producing optimal solutions in modest computing time as compared to solving the mixed integer formulation of the problem using the branch-and-bound based solver CPLEX[9].

**4.2. Genetic Algorithms.** Genetic algorithms (GAs) represent a broad class of heuristics for global optimization problems. Intuitively, they are designed to mimic the biological process of evolution, and follow Darwin's Theory of Natural Selection [10]. GAs store a set of solutions, or a *population*, and the population *evolves* by replacing these solutions with better ones based on certain fitness criteria represented by the objective function value. In successive iterations, or *generations*, the population evolves by *reproduction*, *crossover*, and *mutation*.

Reproduction is the probabilistic selection of the next generations elements determined by their fitness level (i.e., objective function value). Crossover is the combination of two current solutions, called *parents*, which produces one or more other solutions, referred to as their *offspring*. Finally, mutation is the random perturbation of the offspring and

```

procedure CriticalNodeLS( $G, k$ )
1   $X^* \leftarrow \emptyset$ 
2   $f(X^*) \leftarrow \infty$ 
3  for  $j = 1$  to MaxIter do
4     $X \leftarrow \text{CriticalNode}(G, k)$ 
5     $X \leftarrow \text{LocalSearch}(X)$ 
6    if  $f(X) < f(X^*)$  then
7       $X^* \leftarrow X$ 
8    end if
9  end
10 return  $(V \setminus X^*)$  /* set of  $k$  nodes to delete */
end procedure CriticalNodeLS

```

FIGURE 3. The multi-start framework for the critical node heuristic.

```

procedure GeneticAlgorithm
1  Generate population  $P_k$ 
2  Evaluate population  $P_k$ 
3  while terminating condition not met do
4    Select individuals from  $P_k$  and copy to  $P_{k+1}$ 
5    Crossover individuals from  $P_k$  and put in  $P_{k+1}$ 
6    Mutate individuals from  $P_k$  and put in  $P_{k+1}$ 
7    Evaluate population  $P_{k+1}$ 
8     $P_k \leftarrow P_{k+1}$ 
9     $P_{k+1} \leftarrow \emptyset$ 
10 end while
11 return best individual in  $P_k$ 
end procedure GeneticAlgorithm

```

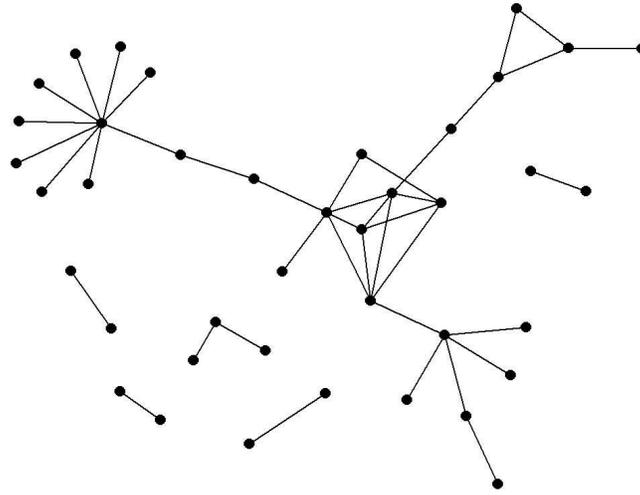
FIGURE 4. Pseudo-code for a generic genetic algorithm.

is implemented as an escape mechanism to avoid getting trapped at local optima.[13]. In successive generations, only those solutions having the best *fitness* are carried to the next generation in a process which mimics the fundamental principle of natural selection, *survival of the fittest* [10]. Figure 4 provides pseudo-code for a standard genetic algorithm. Genetic algorithms were introduced in 1977 by Holland, and were greatly invigorated by the work of Goldberg. [13]

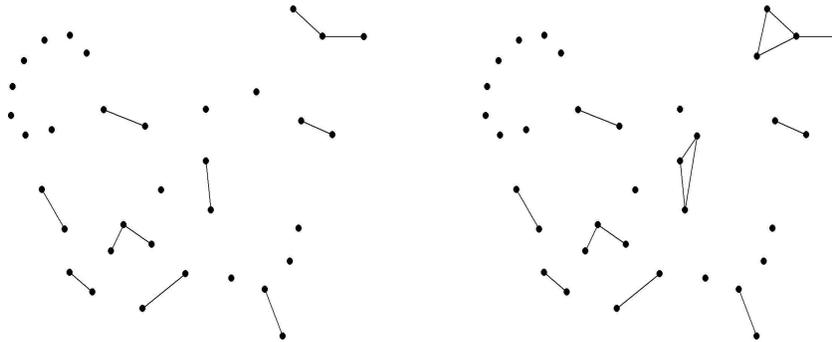
## 5. COMPUTATIONAL EXPERIMENTS

In this section, we present some preliminary computational results on real protein-protein interaction networks obtained from the literature. In particular, three graphs are tested with various values of the connectivity index bound. Both of the aforementioned heuristics were implemented in the C++ programming language and compiled using GNU g++ version 4.1.2, using optimization flag -O4. They were compiled on a Linux workstation equipped with a 3.0 GHz Intel<sup>®</sup> Xeon<sup>®</sup> processor and 1.0 gigabytes of RAM. For more information on specific tuning parameters of the heuristics, the reader is referred to

the papers by Arulsevan et al.[2, 3]. The major focus of this chapter is to introduce the concept and related techniques of finding critical nodes in a graph to the computational biology research community.



(a) The original network.



(b) The optimal subgraph for  $L = 3$ .

(c) The optimal subgraph for  $L = 4$ .

FIGURE 5. The CARDINALITY CONSTRAINED CRITICAL NODE DETECTION PROBLEM is solved for the 46 primary interactions of the *S. cerevisiae* cell cycle.

We begin by examining the 46 primary interactions of the yeast *S. cerevisiae* cell cycle[29]. The graph of the original network can be seen in Figure 5(a). The optimal solutions for the case where  $L = 3$  and  $L = 4$  are provided in Figures 5(b) and 5(c) respectively. The

corresponding numerical results are provided in Table 1. The table contains the objective function values and the corresponding computation times required by the algorithms. We provide the optimal solutions as computed by CPLEX and the solutions provided by the heuristics mentioned above. For this small, and relatively unconnected example, we see that all the methods were able to obtain optimal solutions in a negligible amount of time.

TABLE 1. Results of the IP model and the genetic algorithm for the 46 interactions of *S. cerevisiae*.

Instance Max Conn. Index ( $L$ )	IP Model		Genetic Alg		Comb. Alg	
	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)
2	8	0.18	8	0.05	8	0.04
3	6	1.19	6	0.01	6	0.00
4	5	2.57	5	0.05	5	0.00
5	4	1.05	4	0.01	4	0.00
6	4	2.63	4	0.04	4	0.00

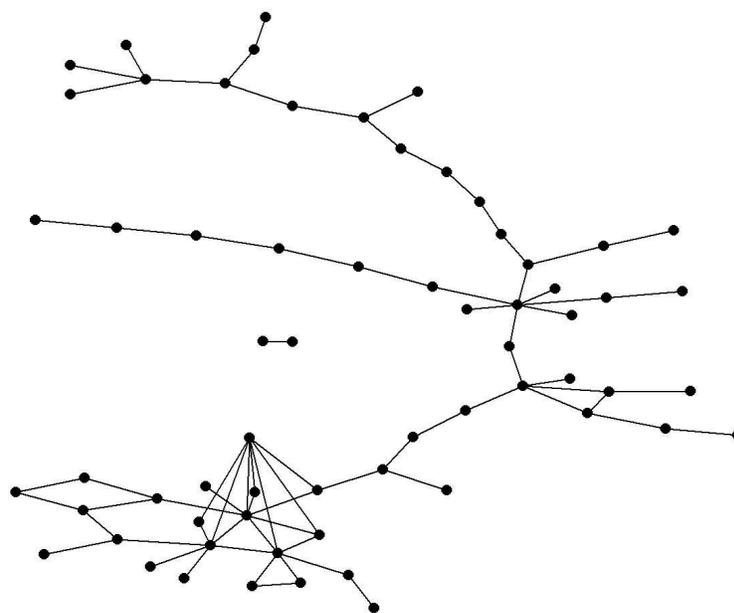
The next network considered is from the 78 protein-protein interactions from the development of *D. melanogaster*[29]. The graph of the network can be seen in Figure 6(a). Similar to the previous example, we provide the graphs corresponding to the optimal solution for the cases of  $L = 5$  and  $L = 4$ . Table 2 reflects the computational results for this instance. As above, we see that the heuristics were able to provide the optimal solutions for each value of  $L$  tested. However, we see that even for this relatively small instance, the required computation time to compute the optimal solution has increased by two orders of magnitude from the previous example.

TABLE 2. Results of the IP model and the genetic algorithm for the 77 primary interactions of *D. melanogaster* development.

Instance Max Conn. Index ( $L$ )	IP Model		Genetic Alg		Comb. Alg	
	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)
2	17	0.87	17	0.17	17	0.04
3	14	36.5	14	0.27	17	0.03
4	12	276.29	12	0.19	17	0.01
5	10	382.88	10	0.28	17	0.02

As a final test case we examine the network comprising 186 yeast two-hybrid system interactions of *S. cerevisiae* proteins[17]. The original network is shown in Figure 7. For this case, CPLEX was unable to compute optimal solutions for any values of  $L$ . Therefore, we only provide solutions for the two heuristics in Table 3. Notice that both heuristics computed the same objective function value in each case. However, the combinatorial algorithm required over 30 seconds for the case where  $L = 2$ .

Though promising, these preliminary results indicate the need for advanced heuristics and exact solution methods for computing critical nodes in protein-protein interaction networks. The primary challenge to computing optimal solutions in real-world networks is that the sizes of the networks prohibit optimal solutions from being calculated using standard branch-and-bound techniques. The test cases presented represent relatively small



(a) The original network.

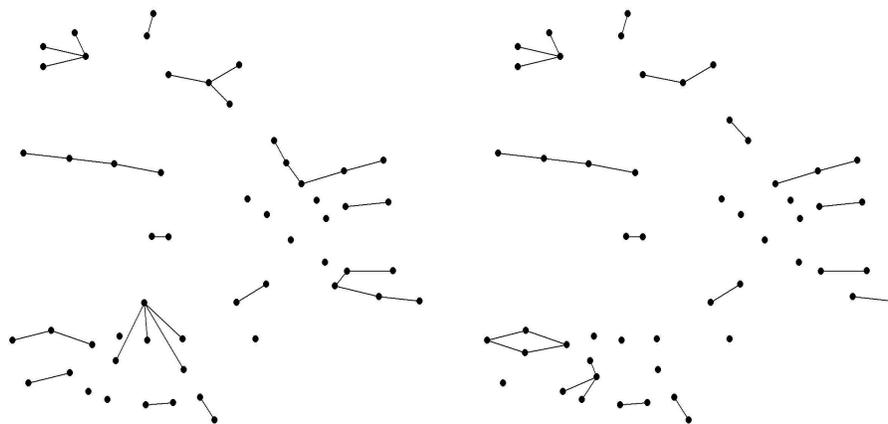
(b) The optimal subgraph for  $L = 5$ .(c) The optimal subgraph for  $L = 4$ .

FIGURE 6. The CARDINALITY CONSTRAINED CRITICAL NODE DETECTION PROBLEM is solved for the 77 primary interactions during the development of *D. melanogaster*.

instances of protein-protein interaction networks found in the literature. It is not uncommon for these graphs to contain tens of thousands of nodes and arcs[16]. Clearly, more sophisticated algorithms are required for graphs of this size.

## 6. CONCLUSIONS

In this chapter, we have identified an important practical application of the recently introduced problem of detecting critical nodes to protein-protein interaction networks. As

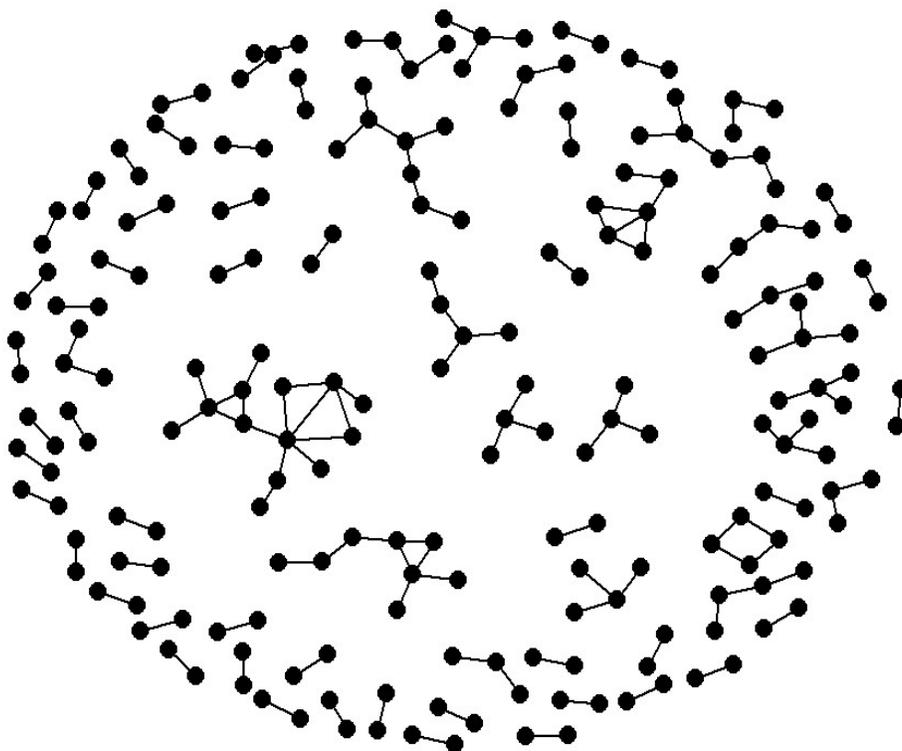


FIGURE 7. The graph shows 186 yeast two-hybrid system interactions of *S. cerevisiae* proteins.

TABLE 3. Results of the genetic algorithm and the combinatorial heuristic for the 186 yeast two-hybrid system interactions of *S. cerevisiae*.

Instance Max Conn. Index ( $L$ )	Genetic Alg		Comb. Alg	
	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)
2	31	7.64	31	23.63
3	20	2.88	20	0.08
4	7	1.92	7	0.03
5	6	3.35	6	0.03
6	3	3.49	3	0.01

indicated above, in many cases, potentially harmful biological organisms, such as bacteria and viruses that cause diseases, can be studied in terms of their protein-protein interaction patterns. Therefore, finding the critical nodes corresponding to the proteins that are the most important for the integrity of the network would be very helpful in terms of identifying the proteins that need to be targeted in the efficient process of destroying this network and neutralizing the corresponding organisms. This approach can potentially be used in drug design applications, e.g., in developing drugs that target specific proteins that are the

most “important” in the considered networks. It would be interesting to consider protein-protein interaction networks corresponding to different dangerous viruses, such as HIV (although obtaining detailed information on these interactions certainly represents another challenge) and identify key proteins that need to be targeted to ensure that these networks are sufficiently disconnected.

In addition to their potential important practical applications, the considered problems need to be studied from the computational perspective as well. As indicated above, these problems are  $\mathcal{NP}$ -hard, and the available exact and heuristic methods do not always perform sufficiently well, especially on large problem instances. Clearly, large-scale protein-protein interaction networks can provide valuable information about the structure of complex molecules and organisms; therefore, efficient techniques for solving the considered problems on massive networks need to be developed.

Overall, we believe that the area of research proposed in this chapter is promising and challenging due to multiple reasons; therefore, this research clearly needs to be conducted further, including both biological and mathematical aspects.

#### REFERENCES

- [1] C. Alves, P.M. Pardalos, and L.N. Vicente, editors. *Optimization in Medicine*. Springer, 2008.
- [2] A. Arulseivan, C.W. Commander, L. Elefteriadou, and P.M. Pardalos. Detecting critical nodes in sparse graphs. *Computers and Operations Research*, under revision, 2008.
- [3] A. Arulseivan, C.W. Commander, P.M. Pardalos, and O. Shylo. Managing network risk via critical node identification. In N. Gulpinar and B. Rustem, editors, *Risk Management in Telecommunication Networks*. Springer, 2008.
- [4] B. Balasundaram, S. Butenko, and S. Trukhanov. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization*, 10:23–29, 2005.
- [5] A.M. Bonvin. Flexible protein-protein docking. *Current Opinion in Structural Biology*, 16(2):194–200, April 2006.
- [6] S. Butenko and W. Wilhelm. Clique-detection models in computational biochemistry and genomics. *European Journal of Operational Research*, to appear, 2008.
- [7] C.W. Commander. *Optimization Problems in Telecommunications with Military Applications*. PhD thesis, University of Florida, 2007.
- [8] C.W. Commander, P.M. Pardalos, V. Ryabchenko, S. Uryasev, and G. Zrazhevsky. The wireless network jamming problem. *Journal of Combinatorial Optimization*, 14(4):481–498, 2007.
- [9] ILOG CPLEX. <http://www.ilog.com/products/cplex>. Accessed October 2006.
- [10] C. Darwin. *The Origin of Species*. Murray, sixth edition, 1872.
- [11] J. Drews. Drug discovery: A historical perspective. *Science*, 287:1960–1964, March 2000.
- [12] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.
- [13] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [14] J.J. Gray. High-resolution protein-protein docking. *Current Opinion in Structural Biology*, 16(2):183–93, April 2006.
- [15] C. Herzberg, L.A. Weidinger, B. Dorrbecker, S. Hubner, J. Stulke, and F.M. Commichau. Spine: A method for the rapid detection and analysis of protein-protein interactions *in vivo*. *Proteomics*, 7(22):4032–5, November 2007.
- [16] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences*, 98(8):4569–74, 2001.
- [17] T. Ito, K. Tashiro, S. Muta, R. Ozawa, T. Chiba, M. Nishizawa, K. Yamamoto, S. Kuhara, and Y. Sakaki. Toward a protein-protein interaction map of the budding yeast: A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proceedings of the National Academy of Sciences*, 97(3):1143–1147, 2000.
- [18] V. Krebs. Uncloaking terrorist networks. *First Monday*, 7, 2002.
- [19] R.P. Mondaini and P.M. Pardalos, editors. *Mathematical Modeling of Biosystems*. Springer, 2008.
- [20] Conference on Drug Discovery Chemistry. La Jolla, CA, April 28-30 2008.

- [21] P.M. Pardalos, V. Boginski, and A. Vazacopoulos, editors. *Data Mining in Biomedicine*. Springer, 2007.
- [22] P.M. Pardalos and J. Principe, editors. *Biocomputing*. Kluwer Academic Publishers, 2002.
- [23] P.M. Pardalos, J.C. Sackellares, P. Carney, and L. Iasemidis, editors. *Quantitative Neuroscience*. Kluwer Academic Publishers, 2004.
- [24] S.L. Schreiber. Target-oriented and diversity-oriented organic synthesis in drug discovery. *Science*, 275(5460):1964–1969, 2000.
- [25] B. Schwikowski, P. Uetz, and S. Fields. A network of protein-protein interactions in yeast. *Nature Biotechnology*, 18(12):1257–61, December 2000.
- [26] M. Selbach and M. Mann. Protein interaction screening by quantitative immunoprecipitation combined with knockdown (quick). *Nature Methods*, 3(12):981–3, December 2006.
- [27] O. Seref, E. Kundakcioglu, and P.M. Pardalos, editors. *Data Mining, Systems Analysis and Optimization in Biomedicine*. Springer, 2008.
- [28] M. Suchanek, A. Radzikowska, and C. Thiele. Photo-leucine and photo-methionine allow identification of protein-protein interactions in living cells. *Nature Methods*, 2(4):261–7, April 2005.
- [29] <http://www.genome.jp>, last accessed, March 25, 2008.

(V. BOGINSKI) DEPT. OF INDUSTRIAL AND SYSTEMS ENGINEERING, UNIVERSITY OF FLORIDA REEF, SHALIMAR, FL, USA.

*E-mail address:* boginski@reef.ufl.edu

(C.W. COMMANDER) AIR FORCE RESEARCH LABORATORY, MUNITIONS DIRECTORATE, EGLIN AFB, FL USA.

*E-mail address:* clayton.commander@eglin.af.mil