2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009



# Graph-Theoretic Topological Control of Biological Genetic Networks

Anil Aswani, Nicholas Boyd, and Claire Tomlin

*Abstract*— **The control of biological genetic networks is an important problem. If the system is abstracted into a graph, then the affect of drugs, pharmaceuticals, and gene therapy can be abstracted as changing the topology of the graph. We consider the control objective of removing the stable oscillations of the genetic network. This control is done using several theorems relating the topology of the network to the dynamics of the system. These theorems suggest that the controller should remove all the negative feedback in the networks. We prove that the problem of minimizing the edges and vertices to remove, in order to remove negative feedback, is NP-hard. In light of this result, a heuristic algorithm to solve this graph problem is presented. The algorithm is applied to several genetic networks, and it is shown that the heuristic gives reasonable results. Additionally, we consider the p53 network and show that the algorithm gives biologically relevant results.**

## I. Introduction

The problem of control of biological genetic networks is an important one. The chief applications of such control are medical in nature. For instance, the problem of deciding which components (e.g. protein or mRNA) of a network to target pharmaceuticals in order to treat a condition is such an application.

Unfortunately, biological genetic networks contain features that make it difficult to do traditional control. Measuring the states of a system for the purposes of feedback control can be prohibitively difficult or even not feasible with current technologies. Moreover, such genetic networks do not typically have inputs that can be changed to do control. In light of these difficulties, we proposed a new framework in [1] for doing the topological control for such networks. A related line of research that has been developed concurrently is [2].

In our framework, we abstract the affect of drugs, pharmaceuticals, and gene therapy to having a graph theoretic interpretation. It is common for biologists to abstract genetic networks, which are dynamical systems, to a signed, directed graph which qualitatively describes the influence of a state on another state. This graph is often referred to as a promotion-inhibition network. Using a quasi-steady-state approximation, we were able to show that drugs and pharmaceuticals can be interpreted as modifying the signed, directed graph by removing vertices or edges of the graph [1].

This work was supported in part by NSF award #CCR-0225610 (ITR), which supports the CHESS at UC Berkeley.

A.Aswani and C.Tomlin are with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA 94720, USA {aaswani,tomlin}@eecs.berkeley.edu

N. Boyd is with the Department of Computer Science, University of California at Berkeley, Berkeley, CA 94720, USA nickboyd@berkeley.edu

### A. Topological Control

If we interpret pharmaceuticals as modifying a graph, then we can try to do control by intelligently modifying the topology of the genetic networks. The topology of a network will remain modified only in the presence of the pharmaceutical: As soon as the pharmaceutical degrades, the topology of the network will return to an unmodified nature, and the system will go back to being uncontrolled. The control that we choose also depends on having correct knowledge of the genetic network.

We can do the topological control by using theorems that relate the topology of the network to the dynamical behavior of the network. A big class of results concerns monotone systems [3], [4], [5], systems with no undirected, negative cycles within the graph of the network. These systems do not have any stable oscillations, and all trajectories converge to equilibrium points. A related class of results concerns systems with no directed, negative cycles within the graph of the network. If the graph is also strongly connected, then all trajectories of the system converge to equilibrium points and there are no stable oscillations [6]. Similar results are found in [7], [8], [9], [10]. These results can be extended to prove that all systems with no directed, negative cycles have the same behavior. The particular case of piecewise-affine hybrid systems with no self-inhibition was proved in [11], [12], and the more general case of arbitrary smooth vector fields was proved in [4].

Directed, negative cycles correspond to our intuitive notion of negative feedback in a system. Undirected, negative cycles, which are a superset of directed, negative cycles, do not always match the intuitive notion of negative feedback. We do the control by removing the negative feedback of the system, which removes the oscillations of the system. This is the goal of the current paper: How can we remove the negative feedback of the system, so that the system trajectories do not have stable oscillations and converge to equilibrium points? The goal is: What should we design pharmaceuticals to target, so that the concentrations in the genetic network converge to equilibria?

This is a crude level of control, but we will provide a biological example of the p53 pathway – which is implicated in cancer – that shows that it can generate useful controllers. This type of control is related to the work in [13], [14]; however, the difference is that removing undirected, negative cycles is a more restrictive condition, because negative feedback is a subset of undirected, negative cycles.

978-1-4244-4524-0/09/$25.00 ©2009 AACC

### B. Abstract Problem Statement

An influence graph (promotion-inhibition network) is a signed, directed graph $G = (V, E, S)$. $V = \{v_1, \ldots, v_n\}$ is the set of vertices, $E \subseteq \{(u, v) : u, v \in V\}$ is the set of directed edges, and $S : E \to \{-1, +1\}$ is a function that gives the sign of an edge. For an edge $e = (u, v)$: $u$ is the direct predecessor of $v$, and $v$ is the direct successor of $u$. Edges labeled $-1$ are called *inhibition* edges, while edges labeled maps to 1 are called *promotion* edges. A simple directed cycle $l = (e_1, e_2, \cdots e_n)$ with all $e_i \in E$ is called a *negative feedback loop* if and only if it contains an odd number of inhibitory edges; in other words:

$$\prod_{e \in l} \gamma(e) = -1.$$

We consider the problem of modifying an influence graph to eliminate such cycles: Given an influence graph $G$ and a weighting function $\omega : E \cup V \mapsto \mathbb{R}$, find the minimum weight subsets (possibly empty) $E' \subset E$ and $V' \subset V$ such that removal of both of these subsets from the influence graph causes the graph to have no negative feedback. As mentioned before, this problem, while similar to the problem of balancing signed directed graphs discussed in [13], [14], is different in that we consider directed, as opposed to undirected, feedback cycles. This problem is also similar to that in [15], but there are differences in: the weights used in the problem, the application of the problem, and the algorithm used to solve the problem.

The weights $\omega$ can be interpreted as the cost of removing an edge or vertex, and there are many biological interpretations of the weight. For instance, if an existing drug can remove an edge, then the weight of that edge can be set low, because there is a lower cost to removing that edge.

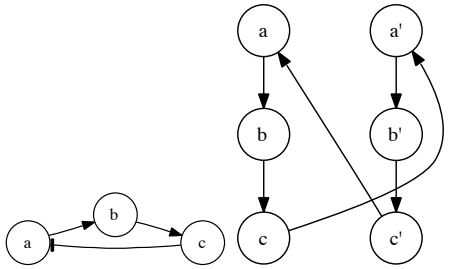
### C. Overview

We first prove that the decision version of the negative feedback edge-vertex deletion problem is in NP by demonstrating a polynomial time algorithm to check whether a given influence graph has negative feedback or not. Then, using results from [16], we prove that the node deletion problem – and thus the node/vertex deletion problem – for negative feedback is NP-hard. We then propose an integer linear program (ILP) for eliminating negative feedback in an arbitrary graph. One can use trivial modifications of efficient approximation algorithms for the directed multicut problem [17], [18] to solve this ILP. Lastly, we use our approach on a few biological examples and focus on the p53 pathway, a pathway involved in human cancers.

## II. Negative Feedback Edge/Vertex Removal is NP-Hard

### A. Decision Version of Negative Feedback Edge/Vertex Removal is in NP

To show that the decision version of the negative feedback problem lies in NP we demonstrate a polynomial time algorithm to determine if a given influence graph has negative feedback cycles. Though this fact was stated without proof in [15], it is useful to formally prove this, because the proof will provide the intuition behind our algorithm. To do this, we first define the operation $\delta : G \to G'$ that maps an influence graph $G = (V, E, S)$ to a directed graph $G' = (V', E')$. We will define the operation $\delta(G) = (\eta(V), \tau(E, S)) = (V', E')$ by defining the two functions $\eta(\cdot)$ and $\tau(\cdot)$.



(a) $G$ (b) $\gamma(G)$

Fig. 1: Influence graph $G$ and $\gamma(G)$

Specifically, we define a bijection $\eta : V \to P$ that makes a clone of every vertex. Here, $P = \{\eta(v) : v \in V\}$ is a clone of every vertex $V$. We define the vertices of the graph $G$ as $V' = V \cup P$.

Next, we define a function that doubles every edge. If the edge is an promotion edge, we make two edges: one of which stays within $V$ and one of which stays within $P$. If, the edge is an inhibition edge, we make an edge that crosses from $V$ to $P$ and an edge that crosses from $P$ to $V$ (see Fig. 1). We do this using a one-to-two correspondence $\tau$ which maps each edge in $E$ to two edges in $\delta(G)$. In particular,

$$\tau((u, v)) = \begin{cases} \{(u, v), (\eta(u), \eta(v))\} & \text{if } \gamma((u, v)) = 1 \\ \{(u, \eta(v)), (\eta(u), v)\} & \text{if } \gamma((u, v)) = -1 \end{cases}$$

$$E' = \bigcup_{e \in E} \tau(e).$$

Note that $\tau^{-1}$ is a function. Note that our construction, illustrated in Fig. 1, superficially resembles the embedding construction in [4]. A closer examination reveals that these two constructions are quite different in terms of operations, and they serve different purposes.

When we discuss a path in $\delta(G)$, we may talk about its preimage in $G$. We obtain the preimage of a path by mapping $\tau^{-1}$ onto each edge in the path. Additionally, for every path $p$ in $G$, we can construct a corresponding path in $\delta(G)$ by choosing an appropriate edge from $\tau(e)$ – for each edge $e$ in $p$ – for the new path.

We contend that the existence of a path in $G'$ from any vertex $x$ to its duplicate $\eta(x)$ implies the existence of a negative feedback loop in the original influence graph $G$, and that any negative feedback loop in $G$ that contains $x$ implies the existence of a path from $x$ to $\eta(x)$ in $G'$.

The first lemma shows that paths that cross an odd number of inhibitory edges are equivalent to paths in $\delta(G)$ that cross from $V$ to $P$.

*Lemma 1:* A path in $\delta(G)$ that begins in $V$ and ends in $P$

has a preimage in $G$ that crosses an odd number of negative edges, and the converse is also true.

*Proof:*
1) By definition of $\tau$, any edge that crosses between $V$ and $P$ must be the image of an inhibitory edge in $G$ - and any inhibitory edge in $G$ must map to two edges that cross between $V$ and $P$.
2) As $V$ and $P$ form a set cover for $\delta(G)$ and are disjoint sets, any path that crosses between $V$ and $P$ must cross between $V$ and $P$ an odd number of times.
3) Following from 1 and 2, a path in $\delta(G)$ crosses between $V$ and $P$ if and only if the preimage of the path in the $G$ cross an odd number of inhibitory edges. ■

Our second Lemma considers the parity of paths with specific start and end vertices.

*Lemma 2:* The existence of a path in $\delta(G)$ that begins at $v \in V$ and ends at $p \in P$ implies the existence of a path in $G$ from $v$ to $\eta^{-1}(p)$ that crosses an odd number of negative edges, and the converse is also true.

*Proof:*
1) By Lemma 1, if we have a path in $\delta(G)$ from $v \in V$ to $p \in P$, the preimage of that path – which goes from $v$ to $\eta^{-1}(p)$ – must cross an odd number of inhibitory edges.
2) Furthermore, if we have a path in $G$ from $v$ to $\eta^{-1}(p)$ that crosses an odd number of inhibitory edges, we can construct a path starting at $v$ in $\delta(G)$ – for each edge $e$ in the path, we choose the edge in $\tau(e)$ such that the path is consistent. Because the original path crossed an odd number of inhibitory edges, by Lemma 1 our new path must end at $p \in P$. ■

Using the prior two Lemmas, we consider the question of odd cycles in $G$ and paths in $\delta(G)$.

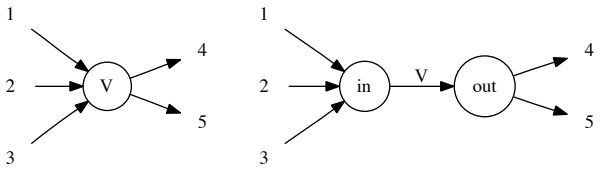
*Theorem 1:* The existence of a negative feedback cycle in $G$ at vertex $v$ implies the existence of a path in $\delta(G)$ from $v$ to $\eta(v)$, and the existence of a path in $\delta(G)$ from $v$ to $\eta(v)$ implies the existence of a negative feedback cycle in $G$.

*Proof:* By definition, a negative feedback cycle in $G$ is a path that crosses an odd number of inhibitory edges. Thus the proof follows directly from Lemma 2. ■

Using these ideas, we can rephrase the question of whether or not $G$ has negative feedback cycles as a question about connectivity in $\delta(G)$. This allows us to check for the existence of negative feedback cycles by checking if there exists a path between any $v \in V$ and $\eta(v) \in P$ in $\delta(G)$. One easy way to do this is by performing $n = |V|$ depth first searches in $\delta(G)$ - a polynomial time operation.

### *B. Negative Feedback Edge/Vertex Removal is NP-Hard*

Using the results of [16], we can show that the node-deletion problem for negative feedback is NP-hard. [16] proves that for any graph property $\Pi$, which is "nontrivial" and "hereditary," the node-deletion problem is NP-hard. *Nontrivial* properties are true for infinitely many graphs and false for infinitely many graphs, while a *hereditary* property is true on all vertex-induced subgraphs of a satisfying graph. For our purposes, we take our $\Pi(G)$ to mean that $G$ has no negative feedback cycles.




(a) Before (b) After

Fig. 2: Vertex before and after splitting

*Theorem 2:* $\Pi$ = "no negative feedback", is a nontrivial property.

*Proof:* It is easy to see that $\Pi$ is nontrivial. Consider the directed cycle graphs $C_n$ ($C_n$ is an $n$-vertex graph that consists of a single, directed cycle) with each edge inhibitory; when $n$ is odd, $C_n$ is clearly a negative feedback loop, and when $n$ is even, $C_n$ must have no negative feedback. Both of these sets are infinite, so $\Pi$ must be nontrivial. ■

*Theorem 3:* $\Pi$ = "no negative feedback", is a hereditary property.

*Proof:* Proof by contradiction. Assume $\Pi(G)$. If $\Pi$ is not hereditary, then for some $G$, there is some vertex-induced subgraph $S$ of $G$ that contains a negative feedback loop. If $S$ is a subgraph of $G$, any feedback loop in $S$ is also in $G$, thus $G$ has a negative feedback loop. Contradiction. ■

Thus we can appeal to Theorem 7 of [16] and conclude that the node deletion problem for negative feedback is NP-hard. By appropriate choice of the weighting function $\omega$, the edge-node deletion problem for negative feedback can also be shown to be NP-hard.

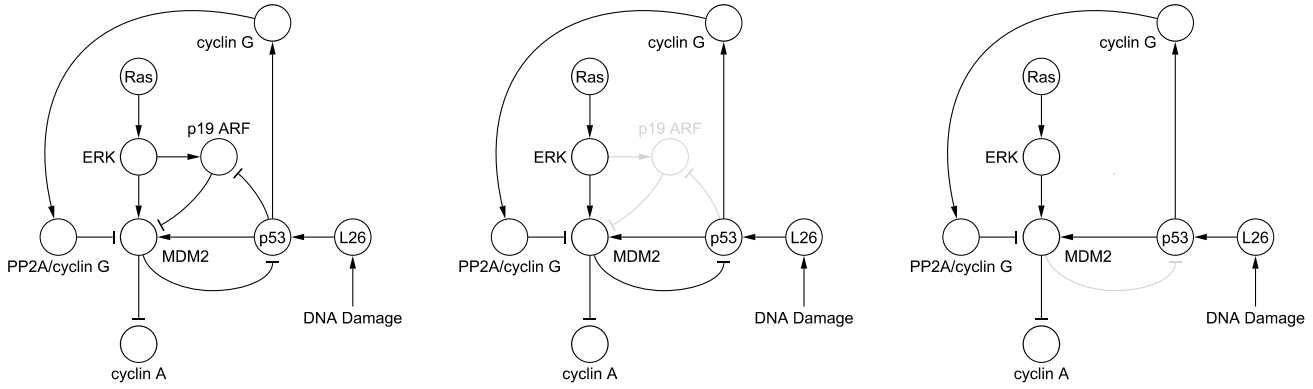
## III. Heuristic Algorithm

The operation $\delta(G)$ provides an interesting way to pose the problem of negative feedback removal: By Theorem 1, we can consider the equivalent objective of disconnecting each pair $v$, $\eta(v)$ in $\delta(G)$. This would ensure no negative feedback in the original influence graph. This rephrased problem is very nearly an instance of the *directed multicut* problem discussed in [23].

The directed multicut problem takes a directed graph and a list of source/sink pairs ($[(s_1, t_1)...(s_k, t_k)]$) to be separated. An optimal solution to multicut finds the minimum weight subset of edges that must be cut in order to separate each source from its corresponding sink. The directed multicut problem is NP-hard. We provide a short description of the multicut integer linear program (ILP).

Each edge is assigned a variable $x_e$ that is either zero or one. If $x_e$ is high in the solution, we cut $e$. In formulating the constraints, we interpret these variables as lengths on each edge. Intuitively, the constraints specify that the minimum distance between a source and its corresponding sink be at least one. This ensures that they cannot be connected once we make the cut.

To achieve this, each vertex is assigned $k$ variables $d_{v,i}$, with $i \in [1, k]$. The optimization variable $d_{v,i}$ is the distance




(a) Normal p53 Pathway (b) Abnormal p53 Pathway (c) Abnormal p53 Pathway with Controller

Fig. 3: When a subsegment of the normal p53 pathway [19], [20], [21], [22] becomes abnormal, such as loss of p19 ARF function [20], the system behaves unfavorably by underexpressing p53. Using a controller, the grayed edge can be removed to make the system behavior more favorably.

between the $i$-th source and the vertex $v$. We then require that each distance is consistent with the edge lengths. Specifically, if vertex $u$ is connected by an edge with zero length to vertex $v$, each $d_{v,i}$ is at most $d_{u,i}$. If they are connected by an edge with length one, each $d_{v,i}$ is at most $d_{u,i} + 1$. Formally, we have:

**Directed Multicut ILP**

$$\begin{aligned}
&\min \sum_{e \in E} \omega(e) \cdot x_e \\
&\text{s.t. for all } i \in [1, k] \\
&\quad d_{t_i,i} - d_{s_i,i} \geq 1 \\
&\quad d_{v,i} \leq d_{u,i} + x_e \qquad \text{for all } e = (u, v) \in E \\
&\quad x_e \in \{0, 1\} \qquad \text{for all } e = (u, v) \in E
\end{aligned}$$

Unfortunately, there are two problems with simply running multicut on $\delta(G)$ with each $v$, $\eta(v)$ pair as a source/sink pair. The first problem is that the multicut problem deals exclusively with cutting edges. Therefore, we must reduce the edge/node deletion problem into the edge deletion problem in order to take advantage of known approximations of multi-terminal cuts [17], [18]. The reduction is fairly intuitive and can be applied to the unmodified influence graph $G$ (with some modification to the sign function) or to the altered digraph $\delta(G)$. To do the reduction, we split each vertex in two to create an in-terminal and an out-terminal for the node. All incoming edges to the vertex are connected to the in-terminal, and all outgoing edges are connected to the out-terminal. We then connect the two terminals with an edge that is equivalent to the original vertex (see Fig. 2). More formally, we transform $\delta(G) = (V, E)$ into $G' = (V', E')$ where

$$V' = \{v_{in}, v_{out} : v \in V\}$$
$$E' = \{(u_{out}, v_{in}) : (u, v) \in E\} \cup \{(v_{in}, v_{out}) : v \in V\}.$$

We then reassign the weights for the vertices to the new edges that connect each in/out vertex pair. The new edges can be considered equivalent to the original vertices. Coincidentally, this manipulation can be used to show that the edge-deletion problem is also NP-hard. From here on, we will consider only the edge-deletion variant of the problem - as we have shown the node/edge deletion problem is equivalent.

Our second problem with directed multicut stems from the fact that $\tau$ relates each edge in $G$ to two edges in $\delta(G)$. So, we separately cut the two edges. To get around this problem, we modify the directed multicut ILP slightly, by having each edge variable correspond to two edges instead of one, to obtain our final formulation of the edge deletion negative feedback problem:

**Negative Feedback ILP**

$$\begin{aligned}
&\min \sum_{e \in E} \omega(e) \cdot x_e \\
&\text{s.t. for all } i \in [1, |V|] \\
&\quad d_{v_i,i} - d_{\eta(v_i),i} \geq 1 \\
&\quad d_{v_i,i} \leq d_{u_i,i} + x_e \qquad \text{for all excitatory edges,} \\
&\quad d_{\eta(v_i),i} \leq d_{\eta(u_i),i} \qquad e = (u, v) \in E \\
&\quad d_{v_i,i} \leq d_{\eta(u_i),i} + x_e \qquad \text{for all inhibitory edges,} \\
&\quad d_{\eta(v_i),i} \leq d_{u_i,i} + x_e \qquad e = (u, v) \in E \\
&\quad x_e \in \{0, 1\} \qquad \text{for all } e \in E
\end{aligned}$$

One can use the approximation algorithms for directed multicut given in [17], [18] to solve the negative feedback ILP. As a substep, these algorithms require the solution of the linear program (LP) formed by relaxing the integer constraints in the directed multicut ILP. We can trivially modify these approximation algorithms to solve our negative feedback ILP by relaxing its integer constraints to form a LP.

TABLE I: Number of edge deletions for monotonicity and number of edge deletions required to eliminate negative feedback

| Network | Vertices | Edges | Monotone | No Negative Feedback | Time (min.) |
|---|---|---|---|---|---|
| EGFR | 330 | 885 | 210 | 45 | 6.5 |
| Yeast | 690 | 1082 | 41 | 1 | 6 |
| Mac rophage | 678 | 1582 | 374 | 74 | 12.5 |

If one makes this change to the approximation algorithms in [17], [18], then one has a heuristic algorithm to solve the negative feedback ILP. We conjecture that is an approximation algorithm for the negative feedback ILP, but we have not made the necessary calculations.

## IV. Results

We implemented our algorithm in Python, using the PuLP library to interface with the COIN LP solver [24]. We ran all tests on a 2.4 GHz Intel Core 2 Duo MacBook Pro with 2 GB of RAM. We evaluated the edge-deletion variant of our algorithm on the three regulatory networks used in [13]: Macrophage, EGFR, and Yeast. We include the size of each network, the number of edges that need to be deleted in order to make the system monotone, and the results of the LP relaxation of the negative feedback ILP. The results are seen in Table I. For the EFGR and Yeast networks, the optimal LP solutions were integer, but for the Macrophage network the solution was fractional. Rounding up increased the value of the objective function from 66 to 74.

It is interesting if we compare our results of removing negative feedback versus the results of removing monotone feedback. Since negative feedback is a subset of monotone feedback, we would expect that we have to remove less edges than the approach of [13]. This is what we see in Table I. The number of edges required to remove negative feedback is significantly less than the number of edges required to remove monotone feedback. This is also interesting, because our approach is a heuristic approach that is not guaranteed to give the true minimum number of edges to remove; [13] uses an algorithm that computes the optimal solution.

### A. p53 Pathway

The p53 protein is an important tumor suppressor, which reacts to stress signals and induces an appropriate cellular response [19], [25], [26], [27]. These stress signals include DNA damage, heat shock, cold shock, and spindle damage. These stress signals lead to a post-translational modification of p53, causing the p53 to trigger downstream pathways involved with cell cycle arrest, cell senescence, or apoptosis [19]. The inactivation of p53 can lead to tumor development [26].

A promotion-inhibition network for a subsegment of the p53 pathway is shown in Fig. 3a. In roughly 10% of human tumors, p53 is inactivated through overexpression of MDM2 [25]. MDM2 can be overexpressed through an inactivation of p19 [20], and this is shown in Fig. 3b. MDM2 works to reduce expression of p53 [19], [25], [26], [27] by increasing the degradation rate of p53 and facilitating the nuclear export of p53 [26], [27]. Thus, inhibition of MDM2 has been considered as a possible strategy for cancer treatment [27], [25].

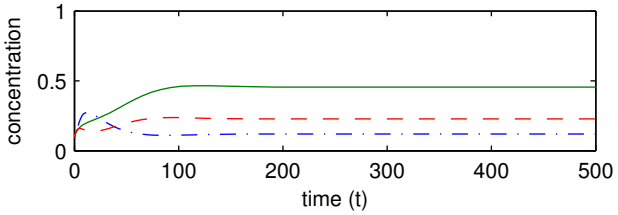
After using our heuristic algorithm on the mutated p53 pathway 3b, our algorithm tells us to cut the grayed edge shown in Fig. 3c. The results are simple, but they are interesting. This is because the edge that we are told to cut is an edge that biologists have studied in detail and devised chemicals to cut [27], [25]. It is also interesting to compare the results of this algorithm to the control of the same system in [1], because the two controllers superficially look the same but have different behaviors and modalities. In the present paper, our control removes an edge; whereas in [1], the control removes a vertex. The behaviors of the controlled system are also different, and this can been seen by comparing Fig. 4 to Fig. 3 in [1].

The effect of the control of cutting the grayed edge shown in Fig. 3c can be seen in Fig. 4. Time course concentrations of p53, cyclin A, and MDM2 are shown in Fig. 4a for the normal p53 pathway, Fig. 4b for the abnormal p53 pathway, and Fig. 4c for the abnormal p53 pathway with controller. These simulations come from an ODE model of the network, and in the simulations we remove the edge between MDM2 and p53, but we do not remove the edge between MDM2 and cyclin A. In the normal p53 pathway, concentrations of p53 and cyclin A are high, and concentrations of MDM2 are low. In the abnormal p53 pathway, p53 and cyclin concentrations are low, whereas MDM2 is in high concentration. In the abnormal p53 pathway with controller, the controller is used at times $t = 200$, $t = 250$, and $t = 300$. The controller causes p53 concentrations to increase to higher levels, and reduces MDM2 concentrations. The cyclin A concentration stays at a reduced level. The controller must be used at multiple times, because the controlling drug is modeled to decay. So, the effect of the controller wains as time goes on. If the controller is not applied again, the system returns to an abnormal state.

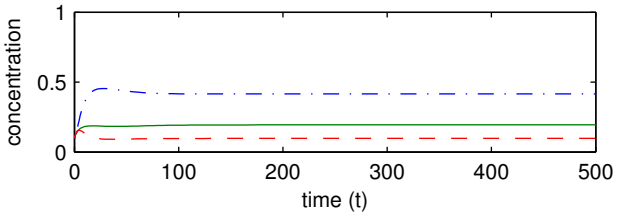
## V. Conclusion

We have devised a heuristic algorithm to remove the negative feedback from a promotion-inhibition network. This is an important problem, because we can abstract the question of which drugs to design to remove stable oscillations from a biological genetic network to a graph-theoretic problem of edge and vertex deletions. Such a method of control is crude, that is it cannot perform specific control actions. However, as seen in the p53 example, the results of the algorithm can be interesting and biologically relevant.

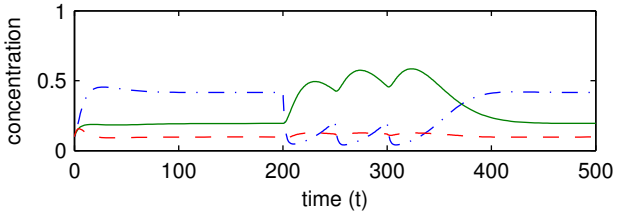
We first showed that the original problem of edge and vertex removal to remove negative feedback from a promotion-




(a) Normal p53 Pathway




(b) Abnormal p53 Pathway




(c) Abnormal p53 Pathway with Controller

Fig. 4: The time course plots for the different pathways displays the effect of the abnormality and the controller. Note that p53 is solid, cyclin A is dashed, and MDM2 is dash-dotted.

inhibition network is NP-hard. Consequently, we focused on developing a heuristic to solve the problem in a reasonable amount of time. By recognizing that our problem was similar to the directed multicut problem (which is also NP-hard), we were able to modify existing approximation algorithms [17], [18] – for solving the directed multicut problem – to solve our problem. We conjecture that our heuristic is an approximation algorithm, because it is based on approximation algorithms for a similar problem. We then applied our new heuristic algorithm to several biological networks, including the p53 pathway.

Future works include two aspects. First of all, we would like to formally prove whether this algorithm is an approximation algorithm. It will likely be a straight-forward extension of the results of [17], [18]. Second of all, we would like to use this algorithm to study other interesting biological networks.

## REFERENCES


[1] A. Aswani and C. Tomlin, "Topology based control of biological genetic networks," in *Proceedings of 47th IEEE CDC*, Cancun, Dec. 2008, to appear.

[2] A. Motter, N. Gulbahce, E. Almaas, and A.-L. Barabási, "Predicting synthetic rescues in metabolic networks," *Molecular Systems Biology*, vol. 4, 2008.

[3] H. L. Smith, *Monotone dynamical systems: an introduction to the theory of competitive and cooperative systems*. American Mathematical Society, 1995.

[4] E. D. Sontag, "Monotone and near-monotone biochemical networks," *Systems and Synthetic Biology*, vol. 1, no. 2, pp. 59–87, April 2007.

[5] H. Kunze and D. Siegel, "A graph theoretical approach to monotonicity with respect to initial conditions," in *Comparison Methods and Stability Theory*, X. Liu and D. Siegal, Eds. CRC, 1994.

[6] T. Kobayashi, L. Chen, and K. Aihara, "Modeling genetic switches with positive feedback loops," *Journal of Theoretical Biology*, vol. 221, pp. 379–399, 2003.

[7] E. Plahte, T. Mestl, and S. Omholt, "Feedback loops, stability, and multistationarity in dynamical systems," *Journal of Biological Systems*, vol. 3, pp. 409–413, 1995.

[8] J.-L. Gouzé, "Positive and negative circuits in dynamical systems," *Journal of Biological Systems*, vol. 6, pp. 11–15, 1998.

[9] O. Cinquin and J. Demongeot, "Positive and negative feedback: striking a balance between necessary antagonists," *Journal of Theoretical Biology*, vol. 216, pp. 229–241, 2002.

[10] C. Soulé, "Graphic requirements for multistationarity," *Complexus*, vol. 1, pp. 123–133, 2003.

[11] A. Aswani and C. Tomlin, "Reachability algorithm for biological piecewise-affine hybrid systems," in *HSCC*, 2007, pp. 633–636.

[12] A. Aswani, "Reachability algorithm for a class of biologically inspired piecewise-affine hybrid systems," Master's thesis, University of California at Berkeley, 2007.

[13] F. Hüffner, N. Betzler, and R. Niedermeier, "Optimal edge deletions for signed graph balancing," in *Experimental Algorithms*, 2007, pp. 297–310.

[14] B. DasGupta, G. Enciso, E. Sontag, and Y. Zhang, "Algorithmic and complexity results for decompositions of biological networks into monotone subsystems," *Biosystems*, vol. 90, no. 1, pp. 161–178, 2007.

[15] E. Sontag, A. Veliz-Cuba, R. Laubenbacher, and A. S. Jarrah, "The effect of negative feedback loops on the dynamics of boolean networks," *Biophysical Journal*, vol. 95, pp. 518–526, 2008.

[16] J. Lewis and M. Yannakakis, "The node-deletion problem for hereditary properties is NP-complete," *Journal Of Computer and System Sciences*, vol. 20, no. 2, pp. 219–230, 1980.

[17] J. Cheriyan, H. Karloff, and Y. Rabani, "Approximating directed multicuts," in *42nd IEEE Symposium on Foundatiosn of Computer Science*, 2001.

[18] A. Gupta, "Improved results for directed multicut," in *14th Annual ACM-SIAM symposium on Discrete algorithms*, 2003.

[19] S. L. Harris and A. J. Levine, "The p53 pathway: positive and negative feedback loops," *Oncogene*, vol. 24, pp. 2899–2908, 2005.

[20] S. Ries, C. Biederer, D. Woods, O. Shifman, S. Shirasawa, T. Sasazuki, M. McMahon, M. Oren, and F. McCormick, "Opposing effects of Ras on p53: Transcriptional activation of MDM2 and induction of p19 ARF," *Cell*, vol. 103, pp. 321–330, 2000.

[21] M. Takagi, M. J. Absalon, K. G. McLure, and M. B. Kastan, "Regulation of p53 translation and induction after DNA damage by ribosomal protein L26 and Nucleolin," *Cell*, vol. 123, pp. 49–63, 2005.

[22] T. Léveillard and B. Wasylyk, "The MDM2 C-terminal region binds to $TAF_{II}250$ and is required for MDM2 regulation of the Cyclin A promoter," *The Journal of Biological Chemistry*, vol. 272, no. 49, pp. 30 651–30 661, 1997.

[23] J. Chuzhoy and S. Khanna, "Hardness of cut problems in directed graphs," in *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 2006, pp. 527–536.

[24] R. Lougee-Heimer, "The Common Optimization INterface for operations research," *IBM Journal of Research and Development*, vol. 47, no. 1, pp. 57–66, 2003.

[25] G. Ganguli and B. Wasylyk, "p53-independent functions of MDM2," *Molecular Cancer Research*, vol. 1, pp. 1027–1035, 2003.

[26] M. Ashcroft and K. H. Vousden, "Regulation of p53 stability," *Oncogene*, vol. 18, pp. 7637–7643, 1999.

[27] L. T. Vassilev, B. T. Vu, B. Graves, D. Carvajal, F. Podlaski, Z. Filipovic, N. Kong, U. Kammlott, C. Lukacs, C. Klein, N. Fotouhi, and E. A. Liu, "In vivo activation of the p53 pathway by small-molecule antagonists of MDM2," *Science*, vol. 303, no. 5659, pp. 844–848, 2004.