Available online at www.sciencedirect.com
ScienceDirect
BioSystems xxx (2006) xxx–xxx
Bio Systems
www.elsevier.com/locate/biosystems

Algorithmic and complexity results for decompositions of biological networks into monotone subsystems

Bhaskar DasGupta[a,1,*], German Andres Enciso[b,2], Eduardo Sontag[c,3], Yi Zhang[a,1]

[a] *Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, United States*
[b] *Mathematical Biosciences Institute, 250 Mathematics Building, 231 W 18th Avenue, Columbus, OH 43210, United States*
[c] *Department of Mathematics, Rutgers University, New Brunswick, NJ 08903, United States*

Received 23 January 2006; received in revised form 3 August 2006; accepted 3 August 2006

Abstract

A useful approach to the mathematical analysis of large-scale biological networks is based upon their decompositions into monotone dynamical systems. This paper deals with two computational problems associated to finding decompositions which are optimal in an appropriate sense. In graph-theoretic language, the problems can be recast in terms of maximal sign-consistent subgraphs. The theoretical results include polynomial-time approximation algorithms as well as constant-ratio inapproximability results. One of the algorithms, which has a worst-case guarantee of 87.9% from optimality, is based on the semidefinite programming relaxation approach of Goemans–Williamson [Goemans, M., Williamson, D., 1995. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J. ACM 42 (6), 1115–1145]. The algorithm was implemented and tested on a Drosophila segmentation network and an Epidermal Growth Factor Receptor pathway model, and it was found to perform close to optimally.

1. Introduction

In living cells, networks of proteins, RNA, DNA, metabolites, and other species process environmental signals, control internal events such as gene expression, and produce appropriate cellular responses. The field of systems (molecular) biology is largely concerned with the study of such networks, viewed as dynamical systems. One approach to their mathematical analysis relies upon viewing them as made up of subsystems whose behavior is simpler and easier to understand. Coupled with appropriate interconnection rules, the hope is that emergent properties of the complete system can be deduced from the understanding of these subsystems. Diagrammatically, we picture this as in Fig. 1, which shows a full system as composed of four subsystems.

A particularly appealing class of candidates for "simpler behaved" subsystems are *monotone systems*, as in Hirsch (1985, 1983) and Smith (1995). Monotone systems are a class of dynamical systems for which pathological behavior ("chaos") is ruled out. Even though they may have arbitrarily large dimensionality, monotone systems behave in many ways like one-dimensional systems. For instance, in monotone systems, bounded trajectories generically converge to steady states, and there are no stable oscillatory behaviors. More precisely, see below, one must extend the notion of monotone system so as to incorporate input and output channels, as

* Corresponding author. Tel.: +1 3123551319; fax: +1 3124130024.
E-mail addresses: dasgupta@cs.uic.edu (B. DasGupta), yzhang3@cs.uic.edu (Y. Zhang), genciso@mbi.osu.edu (G.A. Enciso), sontag@math.rutgers.edu (E. Sontag).
[1] Partly supported by NSF grants CCR-0296041, CCR-0206795, CCR-0208749 and IIS-0346973.
[2] Work done while the author was with the Mathematics Department of Rutgers University and partly supported by NSF grant CCR-0206789.
[3] Partly supported by NSF grants EIA 0205116 and DMS-0504557.

0303-2647/$ – see front matter © 2006 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.biosystems.2006.08.001

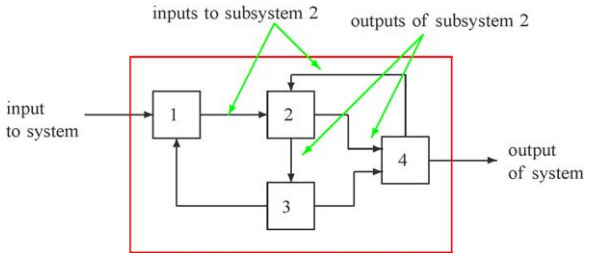

Fig. 1. A system composed of four subsystems.

introduced and initially developed in Angeli and Sontag (2003); inputs and outputs are required so that interconnections like those shown in Fig. 1 can be defined.

Monotonicity is closely related, as explained later, to positive and feedback loops in systems. The topic of analyzing the behaviors of such feedback loops is a long-standing one in biology in the context of regulation, metabolism, and development; a classical reference in that regard is the work (Monod and Jacob, 1961) of Monod and Jacob in 1961. See also, for example, Angeli et al. (2004), Angeli and Sontag (2004), Cinquin and Demongeot (2002), Lewis et al. (1977), Meinhardt (1978), Plathe et al. (1995), Remy et al. (2003), Snoussi (1998) and Thomas (1978).

An interconnection of monotone subsystems, that is to say, an entire system made up of monotone components, may or may not be monotone: "positive feedback" (in a sense that can be made precise) preserves monotonicity, while "negative feedback" destroys it. Thus, oscillators such as circadian rhythm generators require negative feedback loops in order for periodic orbits to arise, and hence are not themselves monotone systems, although they can be decomposed into monotone subsystems (cf. Angeli and Sontag, 2004). A rich theory is beginning to arise, characterizing the behavior of non-monotone interconnections. For example, Angeli and Sontag (2003) shows how to preserve convergence to equilibria; see also the follow-up papers (Angeli et al., 2004; Enciso et al., 2005; Enciso and Sontag, 2006; Gedeon and Sontag, 2005; De Leenheer et al., 2005). Even for monotone interconnections, the decomposition approach is very useful, as it permits locating and characterizing the stability of steady states based upon input/output behaviors of components, as described in Angeli and Sontag (2004); see also the follow-up papers (Angeli et al., 2004; Enciso and Sontag, 2005; De Leenheer and Malisoff, 2006).

Moreover, a key point brought up in Sontag (2004, 2005) is that new techniques for monotone systems in many situations allow one to characterize the behavior of an entire system, based upon the "qualitative" knowledge represented by general network topology and the inhibitory or activating character of interconnections, combined with only a relatively *small amount of quantitative* data. The latter data may consist of steady-state responses of components (dose-response curves and so forth), and there is no need to know the precise form of dynamics or parameters such as kinetic constants in order to obtain global stability conclusions.

In Section 2 of this paper, we briefly discuss monotonicity of systems described by ordinary differential equations (the study of monotonicity can be extended to partial differential equations, delay-differential equations, and even more arbitrary dynamical systems, see e.g. Enciso and Sontag, 2006 in the context of monotone systems with inputs and outputs). We explain there how the study of monotone systems, and more generally of decompositions into monotone systems, relates to a *sign-consistency* property for the graph which describes how each state variable influences each other variable in a given system.

Generally, a graph, whose edges are labeled by "+" or "−" signs (sometimes one writes +1, −1 instead of +, −, or uses respectively activating "→" or inhibiting

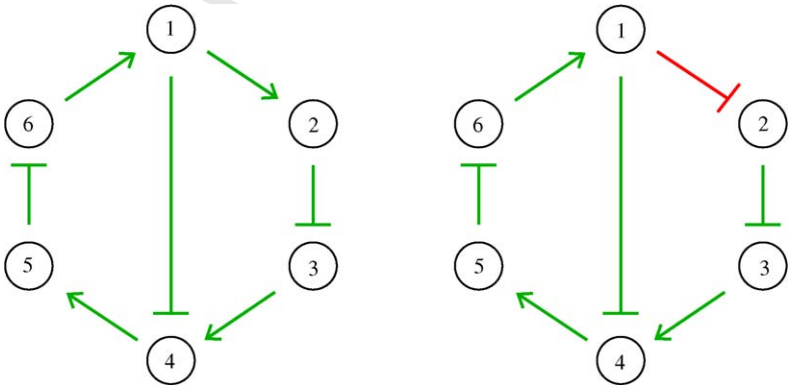

Fig. 2. A consistent and an inconsistent graph.

Please cite this article as: Bhaskar DasGupta et al., Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, BioSystems (2006), doi:10.1016/j.biosystems.2006.08.001

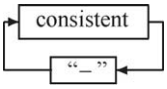

Fig. 3. Pulling-out inconsistent connections.

"⊣" arrows as shown in Fig. 2), is said to be *sign-consistent* if all paths between any two nodes have the same net sign, or equivalently, all closed loops have positive parity, i.e. an even number, possibly 0, of negative edges. (For technical reasons, one ignores the direction of arrows, looking only at undirected graphs; see more details in Section 2.) Thus, the first graph in Fig. 2 is consistent, but the second one, which differs in just one edge from the first one, is not (two paths with different parity are possible from node 1 to node 4, a direct odd one as well as an even one transversing nodes 2 and 3). Self-loops, which in biochemical systems often represent degradation terms, are ignored in this definition. (We discuss this point further below.)

When applying decomposition theorems such as those described in Angeli et al. (2004), Angeli et al. (2004), Angeli and Sontag (2003, 2004), Enciso et al. (2005), Enciso and Sontag (2005), Enciso and Sontag (2006), Gedeon and Sontag (2005), De Leenheer et al. (2005) and De Leenheer and Malisoff (2006), Sontag (2004, 2005), it tends to be the case that *the fewer the number of interconnections among components, the easier it is to obtain useful conclusions.* One may view a decomposition into interconnections of monotone subsystems as the "pulling out" of "inconsistent" connections among monotone components, the original system being a "negative feedback" loop around an otherwise consistent system, as represented in Fig. 3. In this interpretation, the number of interconnections among monotone components corresponds to the number of variables being fed-back. In addition, and independently from the theory developed in the above references, one might speculate that nature tends to favor systems that are decomposable into small monotone interconnections (or equivalently, have a small number of inconsistent paths). There are two reasons for this.

From a dynamical systems perspective, negative feedback loops, although required for homeostasis and for periodic behavior, have potentially destabilizing effects, especially if there are signal propagation delays; thus, minimizing their number is desirable.

Another advantage of consistency is as follows (Sontag, in preparation). Suppose that the nodes in the graphs shown in Fig. 2 represent concentrations of a chemical species in a cell, such as receptors in a certain activated state or transcription factors. Assume now that a perturbation instantaneously increases the value of the concentration of node 1. For the graph on the left, the instantaneous effect on the other nodes is predictable: nodes 2 and 6 will increase, while nodes 3, 4, and 5 will decrease. This unambiguous global effect holds true regardless of the actual algebraic forms of reactions, values of parameters such and kinetic constants, etc. In contrast, consider the graph shown on the right. Now the net effect of an increase in node 1 is ambiguous. It is impossible to know if node 4 will be repressed (because of the direct edge from 1 to 4) or activated (because of the indirect path). There is no way to resolve this ambiguity unless equations and precise parameter values are assigned to the arrows. Since cells of the same type differ in precise parameter values, due to varying concentrations of ATP, enzymes, and other chemicals, two cells of the same type may react in different ways to the same "stimulus" (increase in concentration of chemical 1). While such epigenetic diversity is sometimes desirable, it makes behavior less predictable. From an evolutionary viewpoint, a "change in wiring" due to a mutation will have an ambiguous effect, in this inconsistent network.

Of course, one should not expect large networks to be globally consistent. However, if the number of inconsistencies in a biological interaction graph is small, it may well be the case that the network is in fact consistent in a practical sense. For example, a gene regulatory network represents all *potential* effects among genes. These effects are mediated by proteins which themselves may need to be "activated" in order to perform their function, and this activation may, in turn, depend on certain extracellular ligands being present. Thus, depending on the particular combination of external signals present, different subgraphs of the original graph describe the system under those conditions, and these graphs may be individually consistent. For example, for the system in Fig. 2, the edge from 1 to 2 may not be present under environmental conditions A, while the edge from 2 to 3 may not be present under conditions B. Thus, under either conditions, A or B, the graph would be consistent, even though the entire network is not. See Sontag (in preparation) for more discussion of these issues. In summary, consistency in biological networks may be desirable, and therefore one might conjecture that true biological networks tend to maximize it. Evidence that this is indeed the case is provided by Ma'ayan et al. (in preparation), where the authors compare certain biological networks and appropriately randomized versions of them and show that the original networks are closer to being consistent, when consistency is measured using a simple heuristic. In the last section of this paper, we apply our algorithms to perform a similar analysis, and once again derive the conclusion that nature seems to favor consistency.

Please cite this article as: Bhaskar DasGupta et al., Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, BioSystems (2006), doi:10.1016/j.biosystems.2006.08.001

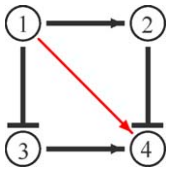

Fig. 4. Dropping the diagonal edge gives consistency.

Thus, we are led to the subject of this paper, namely computing the smallest number of edges that have to be removed so that there remains a consistent graph. For example, for the particular graph shown in Fig. 4 the answer is that one edge (the diagonal positive one) suffices (in this case, the solution is unique: no single other edge would suffice; in other problems, there may be more than one optimizing solutions).

There has been other work dealing with efficient knock-out strategies in biochemical reaction networks, also formulated, as in this paper, as edge deletion problems. As an example, we mention the recent paper (Klamt, 2006), which dealt with the question of identifying a minimal set of reactions whose removal would block the operation of a prespecified reaction. The problem that we consider is completely different, however.

In this paper, we will study the computational complexity of the question of how many edges must be removed in order to obtain consistency, and we provide a relaxation-based polynomial-time approximation algorithm guaranteed to solve the problem to about 87.9% of the optimum solution, which is based on the semidefinite programming relaxation approach of Goemans–Williamson Goemans and Williamson (1995) (A variant of the problem is discussed as well.) We also observe that it is not possible to have a polynomial-time algorithm with performance too close to the optimal. While our emphasis is on theory, one of the algorithms was implemented, and we show results of its application to a Drosophila segmentation network and to an Epidermal Growth Factor Receptor pathway model. It turns out that, when applying the algorithm, often the solution is much closer to optimal than the worst-case guarantee of 87.9%, and indeed often gives an optimal solution.

The remainder of this paper is organized as follows. Section 2 briefly discusses monotonicity. The discussion is self-contained for the purposes of this paper, and references are given to the dynamical systems results that motivate the problem studied here. The connection to consistency is also explained there. Section 3 discusses the associated graph-theoretic problems and notions of approximability used in the paper, leading to the statement of our main theoretical results in Section 4, which are proved in Section 5. Section 6 contains the mentioned examples of application of the algorithm. Finally, in Section 6.3 we consider a yeast gene regulatory network and various randomized versions of it, concluding that the original network is far closer to consistent than may be expected from chance alone. Several technical proofs are separately provided in Appendix A.

2. Monotone systems and consistency

We will illustrate the motivation for the problem studied here using systems of ordinary differential equations

$$\dot{x} = F(x) \tag{1}$$

(the dot indicates time derivative, and $x = x(t)$ is a vector), although the discussion applies as well to more general types of dynamical systems such as delay-differential systems or certain systems of reaction-diffusion partial differential equations. In applications to biological networks, the component $x_i(t)$ of the vector $x = x(t)$ indicates the concentration of the ith species in the model at time t.

We will restrict attention to models in which the direct effect that one given variable in the model has over another is unambiguous, in the sense that it is always inhibitory or always promoting. Thus, if protein A binds to the promoter region of gene B, we assume that it does so either to prevent the transcription of the gene or to facilitate it, no matter what are the respective concentrations. Mathematically, what we are saying is that we require that for every i, $j = 1, \ldots, n$, $i \neq j$, the partial derivative $\partial F_i/\partial x_j$ be either ≥ 0 at all states or ≤ 0 at all states.

Let us briefly discuss this non-ambiguity assumption. First of all, we remark that this assumption does not prevent protein A from having an indirect influence, through other molecules, perhaps dimmers of A itself, that can ultimately lead to the opposite effect on gene B from that of a direct connection. Indeed, this is the whole point of studying graph consistency. Second, in biomolecular networks, ambiguous signs in Jacobians often represent heterogeneous mechanisms. For example, take the case where protein A enhances the transcription rate of gene B only if it is present at low concentrations, but represses B if its concentration is larger than some threshold. A careful study of the chemical mechanism often reveals the existence of an intermediate form (perhaps a homodimer) that is responsible for this ambiguous effect. (Mathematically, an example is a rate of transcription $k_1 a - k_2 a^2$, where a denotes the concentration of A.) Introducing a new species into the model (mathematically, an additional state variable

Please cite this article as: Bhaskar DasGupta et al., Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, BioSystems (2006), doi:10.1016/j.biosystems.2006.08.001

representing this intermediate form) reduces one to the problem in which Jacobian entries are unambiguous. (In our example, we would write the rate as $k_1 a - k_2 c$, where c is the concentration of the dimer. In addition, there would be a new equation such as $dc/dt = k_3 a^2 - k_4 c$ representing formation of the dimer and its degradation.) Finally, we note that small-scale negative loops are abundant in nature. Self-loops or "auto repression" are an extreme example of these, and appear as a consequence of degradation and other effects. Regarding such self-loops, observe that the requirement of a fixed sign for Jacobian entries is not imposed on diagonal elements. In fact, these elements play no role in the graph to be introduced next, nor on monotonicity—the properties of monotone systems are not affected by them. More generally, it is often the case that small loops represent fast dynamics which may be collapsed into a self-loops via time-scale decomposition (singular perturbations or, specifically for enzymes, "quasi-steady state approximations") and hence may be viewed and diagonal terms which may be safely ignored. This is a modeling question, to be settled before the algorithms studied here are to be applied.

Given any partial order $\leq$ defined on $\mathbb{R}^n$, a system (1) is said to be *monotone with respect to* $\leq$ if $x_0 \leq y_0$ implies $x(t) \leq y(t)$ for every $t \geq 0$. Here $x(t)$, $y(t)$ are the solutions of (1) with initial conditions x_0, y_0, respectively. Of course, whether a system is monotone or not depends on the partial order being considered, but we one says simply that a system is *monotone* if the order is clear from the context. Monotonicity with respect to nontrivial orders rules out chaotic attractors and even stable periodic orbits; see Hirsch (1985, 1983), Smith (1995), and is, as discussed in the introduction, a useful property for components when analyzing larger systems in terms of subsystems.

A useful way to define partial orders in $\mathbb{R}^n$, and the only one to be further considered in this paper, is as follows. Given a tuple $s = (s_1, \ldots, s_n)$, where $s_i \in \{1, -1\}$ for every i, we say that $x \leq_s y$ if $s_i x_i \leq s_i y_i$ for every i. For instance, the "cooperative order" is the orthant order $\leq_s$ generated by $s = (1, \ldots, 1)$. This is the order $\leq$ defined by $x \leq y$ if and only if $x_i \leq y_i$ for all $i = 1, \ldots, n$. It is not difficult to verify if a system is cooperative with respect to an orthant order; the following lemma, known as "Kamke's condition," is not hard to prove, see Smith (1995) for details (also Angeli and Sontag, 2003 in the more general context of monotone systems with input and output channels).

Lemma 1. *Consider an orthant order $\leq_s$ generated by $s = (s_1, \ldots, s_n)$. A system (1) is monotone with respect to $\leq_s$ if and only if*

$$s_i s_j \frac{\partial F_j}{\partial x_i} \geq 0, \quad i, j = 1, \ldots, n, \quad i \neq j. \tag{2}$$

To provide intuition, let us sketch the sufficiency part of the proof for the special case of the cooperative order. Suppose by contradiction that the system is not monotone, and that therefore there is a pair of initial conditions $x_0 \leq y_0$ whose solutions $x(t)$, $y(t)$ cease to satisfy $x(t) \leq y(t)$ at some point. This implies that at a certain critical moment in time t, there is some coordinate i so that $x_i(t^-) < y_i(t^-)$ but $x_i(t^+) > y_i(t^+)$. (This argument is not entirely accurate, but it gives the flavor of the proof.) Thus $x_i(t) = y_i(t)$ for some i and the derivative with respect to time of x_i is larger than that of y_i at time t, meaning that that $F_i(x) > F_i(y)$, where $x = x_i(t)$ and $y = y_i(t)$. However, this cannot happen if F_i is increasing on all the variables x_j except possibly x_i, so that $x \leq y, x_i = y_i$ implies $F_i(x) \leq F_i(y)$. An equivalent way to phrase this condition is by ask that $\partial F_i/\partial x_j \geq 0$ at all states for every $i, j, i \neq j$, which is the Kamke condition for the special case of the cooperative order. The name of the order arises because in a monotone system with respect to that order each species promotes or "cooperates" with each other.

A rephrasing of this characterization of monotonicity with respect to orthant orders can be given by looking at the *signed digraph* G associated to (1). We define the vertex set $V(G)$ and the edge set $E(G)$ of G as follows. Let $V(G) = \{1, \ldots, n\}$, and given vertices i, j, let $(i, j) \in E(G)$ and $f_E(i, j) = 1$ if both $\partial F_j/\partial x_i \geq 0$ and the strict inequality holds at least at one state. Similarly let $(i, j) \in E(G)$ and $f_E(i, j) = -1$ if both $\partial F_j/\partial x_i \leq 0$ and the strict inequality holds at least at one state. Finally, let $(i, j) \notin E(G)$ if $\partial F_j/\partial x_i \equiv 0$. Recall that we are assuming that one of the three cases must hold.

Now we can define an orthant cone using any function $f_V : V(G) \to \{-1, 1\}$, by letting $x \leq_{f_V} y$ if and only if $f_V(i) x_i \leq f_V(i) y_i$ for all i. Given f_V, we define the consistency function $g : E(G) \to \{\text{true, false}\}$ by $g(i, j) = f_V(i) f_V(j) f_E(i, j)$. Then, the following analog of Lemma 1 holds.

Lemma 2. *Consider a system (1) and an orthant cone $\leq_{f_V}$. Then (1) is monotone with respect to $\leq_{f_V}$ if and only if $g(i, j) \equiv 1$ on $E(G)$.*

Proof. Let $s_i = f_V(i), i = 1, \ldots, n$. Note that $s_i s_j \partial f_i/\partial x_j = 0$ if $(i, j) \notin E(G)$. For $(i, j) \in E(G)$, it holds that $s_i s_j \partial f_i/\partial x_j \geq 0$ if and only if $s_i s_j f_E(i, j) = 1$,

Please cite this article as: Bhaskar DasGupta et al., Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, BioSystems (2006), doi:10.1016/j.biosystems.2006.08.001

that is, if and only if $g(i, j) = 1$. The result follows from Lemma 1. □

For the next lemma, let the *parity* of a chain in G be the product of the signs $(+1, -1)$ of its individual edges. We will consider in the next result closed *undirected chains*, that is, sequences $x_{i_1}, \ldots, x_{i_r}$ such that $x_{i_1} = x_{i_r}$, and such that for every $\lambda = 1, \ldots, r-1$ either $(x_{i_\lambda}, x_{i_{\lambda+1}}) \in E(G)$ or $(x_{i_{\lambda+1}}, x_{i_\lambda}) \in E(G)$.

The following lemma (see DeAngelis et al., 1986 as well as Smith, 1988, page 101) is analogous to the fact from vector calculus that path integrals of a vector field are independent of the particular path of integration if and only if there exists a potential function. Since the result is key to the formulation of the problem being considered, we provide a simple and self-contained proof in Appendix A.

Lemma 3. *Consider a dynamical system (1) with associated directed graph G. Then (1) is monotone with respect to some orthant order if and only if all closed undirected chains of G have parity 1.*

2.1. Systems with inputs and outputs

As we discussed in the introduction, a useful approach to the analysis of biological networks consists of decomposing a given system into an interconnection of monotone subsystems. The formulation of the notion of interconnection requires subsystems to be endowed with "input and output channels" through which information is to be exchanged. In order to address this we consider *controlled* dynamical systems (Sontag, 1990) which are systems with an additional parameter $u \in \mathbb{R}^m$ and which have the form

$$\dot{x} = g(x, u). \tag{3}$$

The values of u over time are specified by means of a function $t \to u(t) \in \mathbb{R}^m$, $t \geq 0$, called an *input* or *control*. Thus each input defines a time-dependent dynamical system in the usual sense. To system (3) there is associated a *feedback function* $h : \mathbb{R}^n \to \mathbb{R}^m$, which is usually used to create the closed loop system $\dot{x} = g(x, h(x))$. Finally, if $\mathbb{R}^n$, $\mathbb{R}^m$ are ordered by orthant orders $\leq_{f_V}, \leq_q$ respectively, we say that the system is monotone if it satisfies (2) for every u, and also

$$q_k f_V(j) \frac{\partial g_j}{\partial u_k} \geq 0, \quad \text{for every } k, j \tag{4}$$

(see also Angeli and Sontag, 2003.) As an example, let us consider the following biological model of testosterone dynamics (Enciso and Sontag, 2004; Murray and Mathematical Biology, 2002):

$$\dot{x}_1 = \frac{A}{K + x_3} - b_1 x_1, \qquad \dot{x}_2 = c_1 x_1 - b_2 x_2,$$
$$\dot{x}_3 = c_2 x_2 - b_3 x_3. \tag{5}$$

Drawing the digraph of this system, it is easy to see that it is not monotone with respect to any orthant order, as follows by application of Lemma 3. On the other hand, replacing x_3 in the first equation by u, we obtain a system that is monotone with respect to the orders $\leq_{(1,1,1)}, \leq_{(-1)}$ for state and input respectively. Defining $h(x) = x_3$, the closed loop system of this controlled system is none other than (5). The paper (Enciso and Sontag, 2004) shows how, using this decomposition together with the "small gain theorem" from monotone input/output theory (Angeli and Sontag, 2003) leads one to a proof that the system does not have oscillatory behavior, even under arbitrary delays in the feedback loop, contrary to the assertion made in Murray and Mathematical Biology (2002).

We can carry out this procedure on an arbitrary system (1) with a directed graph G, as follows: given a set E of edges in G, enumerate the edges in E^C as $(i_1, j_1), \ldots, (i_m, j_m)$. For every $k = 1, \ldots, m$, replace all appearances of x_{i_k} in the function F_{j_k} by the variable u_k, to form the function $g(x, u)$. Define $h(x) = (x_{i_1}, \ldots, x_{i_m})$. It is easy to see that this controlled system (3) has closed loop (1).

Note that the controlled system (3) generated by the set E as above has, as associated digraph, the sub-digraph of G generated by E. This is because for every k, one has $\partial g_{j_k}(x, u)/\partial x_{i_k} \equiv 0$, i.e., the edge from i_k to j_k has been "erased".

Denote by $\hat{G}$ the underlying undirected graph of a directed graph G obtained by ignoring the directions of the edges. Given a set $E \subseteq V(G)$ of vertices in a (directed or undirected) graph G, denote by $G(E)$ the undirected subgraph of G generated by E. The edges of both $\hat{G}$ and $G(E)$ are labeled with ± 1 using the labels in the edges of G, whenever appropriate. Let E be called *consistent* if $\hat{G}(E)$ has no closed chains with parity -1. Note that this is equivalent to the existence of f_V such that $g \equiv 1$ on E, by Lemma 4 applied to the open loop system (3). If E is consistent, then the associated system (3) itself can also be shown to be monotone: to verify condition (4), simply define each q_k so that (4) is satisfied for k, j_k. Since $\partial g_{j_k}/\partial u_k = \partial F_{j_k}/\partial x_{i_k} \not\equiv 0$, this choice is in fact unambiguous. Conversely, if (3) is monotone with respect to the orthant orders $\leq_{f_V}, \leq_q$, then in particular it is monotone for every fixed constant u, so that E is consistent by Lemma 3. We thus have the following result.

Please cite this article as: Bhaskar DasGupta et al., Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, BioSystems (2006), doi:10.1016/j.biosystems.2006.08.001

Lemma 4. *Let E be a set of edges of the digraph G. Then E is consistent if and only if the corresponding controlled system (3) is monotone with respect to some orthant orders.*

3. Statement of problem

A natural problem is therefore the following. Given a dynamical system (1) that admits a digraph G, use the procedure above to decompose it as the closed loop of a monotone controlled system (3), while minimizing the number $\|E^C\|$ of inputs. Equivalently, *find f_V such that $P(E_+) = \|E_+\|$ is maximized and $P(E_-) = \|E_-\| = \|E_+^C\|$ minimized*. This produces the following problem formulation.

Problem 1 (Undirected labeling problem (ULP)). An instance of this problem is (G, h), where $G = (V, E)$ is an undirected graph and $h : E \mapsto \{0, 1\}$. A valid solution is a vertex labeling function $f : V \to \{0, 1\}$. Define an edge $\{u, v\} \in E$ to be consistent iff $h(u, v) \equiv (f(u) + f(v)) \pmod 2$. The objective is then to find a valid solution *maximizing* $|F|$ where F is the set of consistent edges.

That ULP is a correct formulation for our problem is confirmed by the following easy equivalence.

Proposition 1. *Consider an instance (G, h) of ULP with an optimal solution having x consistent edges given by a vertex labeling function f. Let D be a set of edges of smallest cardinality that have to be removed such that for the remaining graph, that is the graph $G' = (V, E \setminus D)$ with the same vertex set V but an edge set $E \setminus D$, there exists a vertex labeling function $f' : V \to \{0, 1\}$ that makes every edge consistent. Then, $x = |E| - |D|$.*

Proof. Since f produces a solution of ULP with x consistent edges, exactly $|E| - x$ edges are inconsistent, thus $|D| \le |E| - x$, that is, $x \le |E| - |D|$. Conversely, since there is a solution with $|E| - |D|$ consistent edges, $x \ge |E| - |D|$. □

A special case of ULP, namely when $h(e) = 1$ for all $e \in E$, is the MAX-CUT problem (defined in Section 3.1). Moreover, ULP can be posed as a special type of "constraint satisfaction problem" as follows. We have $|E|$ linear equations over $GF(2)$, one equation per edge and each equation involving exactly two variables, over $|V|$ Boolean variables. The goal is to assign values to the variables to satisfy the maximum number of equations. For algorithms and lower-bound results for general cases of these types of problems, such as when the equations are over $GF(p)$ for an arbitrary prime $p > 2$, when there are an arbitrary number of variables per equation or when the goal is to minimize the number of unsatisfied equations, see references such as Amaldi and Kann (1996), Berman and Karpinski (2001), Creignou et al. (2001) and Hastad and Venkatesh (2002) and the references therein.

Another interpretation (Sontag, in preparation) of ULP is in statistical mechanics terms. Let us label edges by "±1" instead of {0, 1}, denoting by $w_{uv} = (-1)^{h(u,v)}$ the edge parities, now called "interaction energies." Similarly, let us consider ±1-valued vertex labeling functions, now called (magnetic) "spin configurations," $\sigma : V \to \{-1, +1\}$, $\sigma(v) = (-1)^{f(v)}$. An edge $\{u, v\}$ is consistent provided that $w_{uv}\sigma_u\sigma_j = 1$. A graph with ±1 weights is called an Ising spin-glass model in statistical physics. A "non-frustrated" spin-glass model is one for which there is a spin configuration for which every edge is consistent (Barahona, 1982; Cipra, 2000; De Simone et al., 1995; Istrail, 2000). This is the same as a consistent graph in our sense. Moreover, a spin configuration that maximizes the number of consistent edges is one for which the "free energy" (with no exterior magnetic field):

$$-\sum_{ij} w_{uv}\sigma_u\sigma_v$$

is minimized, a "ground state". (When $h(e) = 1$ or equivalently $w_e = -1$ for all edges, one has what is called the "anti-ferromagnetic case".) Thus, our problem amounts to finding ground states.

Given orthant orders $\le_{f_V}$ and $\le_q$ for $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively, we say that a feedback function h is *positive* if $x \le_{f_V} y$ implies $h(x) \le_q h(y)$, and that it is *negative* if $x \le_{f_V} y$ implies $h(x) \ge_q h(y)$. It can be shown that the closed loop of a monotone system with a positive feedback function is actually itself monotone, so that no system can be produced in this way that was not monotone already. But if h is a negative feedback function, then several results become available which use the methods of monotone systems for systems that are not monotone, see Angeli and Sontag (2003), Enciso and Sontag (2004) and Enciso and Sontag (2006). For the following result, let $(\mathcal{C}, \subseteq)$ be the class of consistent subsets of $E(G)$, ordered under inclusion.

Proposition 2. *Let E be a consistent set. Then E is maximal in $(\mathcal{C}, \subseteq)$ if and only if h is a negative feedback function for every f_V such that $g \equiv 1$ on E.*

Proof. Suppose that E is maximal, and let f_V be such that $g \equiv 1$ on E. Given any edge $(i_k, j_k) \in E^C$, it holds that $g(i_k, j_k) = -1$. Otherwise one could extend E by adding (i_k, j_k), thus violating maximality. That is, $f_V(i_k)f_V(j_k)f_E(i_k, j_k) = -1$. By monotonicity, it holds that $q_k f_V(j_k)\partial g_{j_k}/\partial u_k \ge 0$, and since $\partial g_{j_k}/\partial u_k =$

Please cite this article as: Bhaskar DasGupta et al., Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, BioSystems (2006), doi:10.1016/j.biosystems.2006.08.001

$\partial F_{j_k}/\partial x_{i_k}$, it follows necessarily that

$q_k f_V(j_k) f_E(i_k, j_k) = 1.$

Therefore it must hold that $q_k = -f_V(i_k)$ for each k, which implies that h is a negative feedback function.

Conversely, if f_V is such that $g \equiv 1$ on E and h is a negative feedback function, then $q_k = -f_V(i_k)$. By the same argument as above, $q_k f_V(j_k) f_E(i_k, j_k) = 1$ for all k by monotonicity. Therefore $g \equiv -1$ on E^C. Repeating this for all admissible f_V, maximality follows. □

There is a second, slightly more sophisticated way of writing a system (1) as the feedback loop of a system (3) using an arbitrary set of edges E. Given any such E, define $S(E^c) = \{i | \text{there is some } j \text{such that } (i, j) \in E^c\}$. Now enumerate $S(E^c)$ as $\{i_1, \ldots, i_m\}$, and for each k label the set $\{j | (i_k, j) \in E^c\}$ as $j_{k1}, j_{k2}, \ldots$. Then for each k, l, one can replace each appearance of x_{i_k} in $F_{j_{kl}}$ by u_k, to form the function $g(x, u)$. Then one lets $h(x) = (x_{i_1}, \ldots, x_{i_m})$ as above. The closed loop of this system (3) is also (1) as before but with the advantage that there are $|S(E^c)|$ inputs, and of course $|S(E^c)| \leq |E^c|$.

If E is a consistent and *maximal* set, then one can make (3) into a monotone system as follows. By letting f_V be such that $g \equiv 1$ on E, we define the order $\leq_{f_V}$ on $\mathbb{R}^n$. For every i_k, j_{kl} such that $(i_k, j_{kl}) \in E^C$, it must hold that $f_V(i_k) f_V(j_{kl}) f_E(i_k, j_{kl}) = -1$. Otherwise $E \cup \{(i_k, j_{kl})\}$ would be consistent, thus violating maximality. By choosing $q_k = -f_V(i_k)$, Eq. (4) is therefore satisfied. See the proof of Proposition 2. Conversely, if the system generated by E using this second algorithm is monotone with respect to orthant orders, and if h is a negative function, then it is easy to verify that E must be both consistent and maximal.

Thus the problem of finding E consistent and such that $P(E_-) = \|S(E_-)\| = \|S(E^C)\|$ is smallest, when restricted to those sets that are maximal and consistent (this does not change the minimum $\|S(E^C)\|$), is equivalent to the following problem: decompose (1) into the negative feedback loop of an orthant monotone control system, using the second algorithm above, and using as few inputs as possible. This produces the following problem formulation.

Problem 2 (Directed labeling problem (DLP)). An instance of this problem is (G, h) where $G = (V, E)$ is a directed graph and $h : E \to \{0, 1\}$. A valid solution is a vertex labeling function $f : V \to \{0, 1\}$. Define an edge $(u, v) \in E$ to be consistent iff $h(u, v) \equiv (f(u) + f(v)) \pmod 2$. The objective is then to find a valid solution *minimizing* $|g(E - F)|$ where $g(C) = \{u \in V \mid \exists y \in V, (u, y) \in C\}$ for any $C \subseteq E$ and F is the set of consistent edges.

3.1. Summary of key concepts and results in approximation algorithms

For any $\gamma \geq 1$ (resp. $\gamma \leq 1$), a *γ-approximate solution* (or simply an γ-approximation) of a minimization (resp., maximization) problem is a solution with an objective value no larger than γ times (resp., no smaller that γ times) the value of the optimum, and an algorithm achieving such a solution is said to have an *approximation ratio* of γ.

In Papadimitriou and Yannakakis (1991) Papadimitriou and Yannakakis defined the class of MAX-SNP optimization problems and a special approximation-preserving reduction, the so-called *L-reduction*, that can be used to show MAX-SNP-hardness of an optimization problem. The version of the L-reduction that we provide below is a slightly modified but equivalent version that appeared in Berman and Schnitger (1992).

Definition 1. Berman and Schnitger (1992), Papadimitriou and Yannakakis (1991) Given two optimization problems Π and Π', we say that Π *L-reduces* to Π' if there are three polynomial-time procedures T_1,T_2, T_3 and two constants a and $b > 0$ such that the following two conditions are satisfied: **(1)** For any instance I of Π, algorithm T_1 produces an instance $I' = f(I)$ of Π' generated from T_1 such that the optima of I and I', OPT(I) and OPT(I'), denoted by respectively, satisfy OPT(I') $\leq a \cdot$ OPT(I). **(2)** For any solution of I' with cost c', algorithm T_2 produces another solution with a cost c'' no worse than c', and algorithm T_3 produces a solution of I of Π with cost c (possibly from the solution produced by T_2) satisfying $|c - \text{OPT}(I)| \leq b \cdot |c'' - \text{OPT}(I')|$.

An optimization problem is MAX-SNP-*hard* if any problem in MAX-SNP L-reduces to that problem. The importance of proving MAX-SNP-hardness results comes from a result proved by Arora et al. Arora et al. (1998) which shows that, assuming P≠NP, for every MAX-SNP-hard minimization (resp., maximization) problem there exists a constant $\varepsilon > 0$ such that no polynomial time algorithm can achieve an approximation ratio better than $1 + \varepsilon$ (resp., better than $1 - \varepsilon$).

A special case of the ULP problem, namely when $h(e) = 1$ for all $e \in E$, is the well-known MAX-CUT problem. An instance of this problem is an undirected graph $G = (V, E)$. A valid solution is a set $S \subseteq V$. The objective is to find a valid solution that *maximizes* the number of edges $\{u, v\} \in E$ such that $|\{u, v\} \cap S| = 1$. The MAX-CUT problem is known to be MAX-SNP-hard. For further details on these topics, the reader is referred to the excellent book by Vazirani (Vazirani, 2001).

Please cite this article as: Bhaskar DasGupta et al., Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, BioSystems (2006), doi:10.1016/j.biosystems.2006.08.001

Some terminology The following notation will be used for the remainder of the paper. Given a set S of vertices in a directed graph G, define $E_{\text{out}}(S) = \{(u, v) \in E(G) | u \in S\}$ as the set of out-bound edges of vertices in S. $\text{OPT}_P(I)$ denotes the size of an optimal solution for a problem P with instance I. Recall that the length of a circuit c is normally defined as the number of edges in the circuit. Given a weight function $w : E \mapsto \mathbb{R}$, the *length of c with respect to w* is defined as $\sum_{e \in c} w(e)$.

4. Theoretical results

Our theoretical results are summarized as follows.

Theorem 1.

(a) *For some constant $\varepsilon > 0$, it is not possible to approximate in polynomial time the ULP and the DLP problems to within an approximation ratio of $1 - \varepsilon$ and $1 + \varepsilon$, respectively, unless $P = NP$.*
(b) *For ULP, we provide a polynomial time α-approximation algorithm where $\alpha \approx 0.87856$ is the approximation factor for the MAX-CUT problem obtained in Goemans and Williamson (1995) via semidefinite programming.*
(c) *For DLP, if $d_{in}^{\max}$ denotes the maximum in-degree of any vertex in the graph, then we give a polynomial-time approximation algorithm with an approximation ratio of at most $d_{in}^{\max} \cdot O(\log |V|)$.*

Our computational results are illustrated in Section 6 by an implementation of the algorithms applied to a 13-node Drosophila segmentation network, as well as to a 200^+ node recently published network of the Epidermal Growth Factor Receptor pathway.

Remark 1. It should be noted that the complexity of ULP becomes tractable if the network is biased significantly towards excitatory connections. Obviously, if all the edges of the given graph $G = (V, E)$ are labeled 0, then it is possible to label the vertices such that all the edges are consistent. Moreover, given any graph G, it is easy to check in $O((|V| + |E|)^3)$ time if an optimal solution contains all the edges as consistent by solving a set of linear equations via Gaussian elimination. Now, suppose that at most L of the edges of G are labeled 1. Then, obviously at most L inconsistent edges exist in any optimal solution. Thus a straightforward way to solve the problem is to consider all possible subsets of edges in which at most L edges are dropped and checking, for each such subset, if there is an optimal solution that contains all the edges as consistent. The total time taken is $O(|V|^{2L} \cdot (|V| + |E|)^3)$, which is a polynomial in $|V| + |E|$ if L is a constant.

5. Proof of Theorem 1

This section provides the proof of Theorem 1, broken up into a series of technical parts.

5.1. Proof of Theorem 1(a)

Based on the discussion in Section 3.1, it suffices to show that both these problems are MAX-SNP-hard. ULP is MAX-SNP-hard since its special case, the MAX-CUT problem, is MAX-SNP-hard. To prove MAX-SNP-hardness of DLP, we need the definitions of the following two problems.

Problem 3 (Node deletion problem with bipartite property (NDBP)). An instance of this problem is an undirected graph $G = (V, E)$. A valid solution is a vertex set $S \subseteq V$, such that $G(V - S)$ is a bipartite graph. The objective is to find a valid solution *minimizing* $|S|$.

Problem 4 (Variance of node deletion problem (*VNDP*)). An instance of this problem is (G, h) where $G = (V, E)$ is a directed graph and $h : E \to \{0, 1\}$. A valid solutions is a vertex set $S \subseteq V$ with the following property: if $G_S = (V_S, E_S)$ is the graph with $V_S = V$ and $E_S = E - E_{\text{out}}(S)$, then $\widehat{G_S}$ is free of odd length circuit with respect to weight function h. The objective is to find a valid solution *minimizing* $|S|$.

First, we note that DLP is *equivalent* to VNDP. If one identifies the solution set S in UNDP with the solution set $g(E - F)$ in DLP, then the set of consistent edges F in DLP corresponds to the E_S in UNDP since every edge $(u, v) \in F$ satisfying $h(u, v) \equiv (f(u) + f(v)) \pmod 2$ is equivalent to stating that $\widehat{G_S}$ is free of odd length circuit with respect to weight function h.

Thus, to prove the MAX-SNP-hardness of DLP it suffices to prove that of VNDP. NDBP is known to be MAX-SNP-hard (Lund and Yannakakis, 1993). We provide a L-reduction from NDBP to VNDP. For an instance of VNDP with graph $G = (V, E)$, construct an instance of DLP with instance (G', h) as follows (note that G' is a digraph):

$$V' = V(G') = V \cup \{A_{u,v}, B_{u,v} | \{u, v\} \in E\},$$

$$E' = E(G')$$

$$= \{(u, A_{u,v}), (A_{u,v}, B_{u,v}), (v, B_{u,v}) | \{u, v\} \in E\},$$

and $h(e) = 1$ for all $e \in E'$ Now, the following holds:

(1) If S is a solution to NDBP, it is also a solution to the generated instance of UNDP. The reason

Please cite this article as: Bhaskar DasGupta et al., Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, BioSystems (2006), doi:10.1016/j.biosystems.2006.08.001

is as follows. Notice that every odd length (resp., even length) circuit $\mathcal{C}$ in G corresponds to an odd length (resp., even length) circuit $\mathcal{C}'$ in $\widehat{G'}$ with respect to the weight function h. Since $G(V-S)$ is a bipartite graph, it is free of odd length circuits. So for each odd length cycle $\mathcal{C}$ of G, there exists $u \in S$ such that the deletion of all out-bound edges of u in G' breaks its corresponding odd length cycle $\mathcal{C}'$.

(2) If S' is a solution to UNDP, then we can construct a solution S of NDBP in the following manner: for each $x \in S'$:

if $x = A_{u,v}$, add u to T; if $x = B_{u,v}$, add v to T;

if $x = u$ or $x = v$, add $x$to T.

It is now easy to see that since the graph $\widehat{G_{S'}}$ is free of odd length circuit with respect to h, $G(V-S)$ has no odd length circuit either.

Hence, we have $\mathrm{OPT}_{\mathrm{UNDP}}(G', h) \leq \mathrm{OPT}_{\mathrm{NDBP}}(G)$. Moreover, given a solution S' of UNDP, we are able to generate a solution S of NDBP such that

$$||S| - \mathrm{OPT}_{\mathrm{NDBP}}(G)| \leq ||S'| - \mathrm{OPT}_{\mathrm{UNDP}}(G', h)|.$$

Thus, our reduction satisfies Definition 1 of a L-reduction with $a = b = 1$.

5.2. Proof of Theorem 1(b)

Our algorithm for ULP uses the semidefinite programming (SDP) technique used by Goemans and Williamson in Goemans and Williamson (1995); hence we use notations and terminologies similar to that used in the paper (readers not very familiar with this technique are also referred to the excellent explanation of this technique in the book by Vazirani Vazirani (2001)). For each vertex $v \in V$, we have a real vector $x_v \in \mathbb{R}^{|V|}$ with $||x_v||_2 = 1$. Then, we can generate from ULP the following vector program (where $\cdot$ denotes the vector inner product):

Solve the following vector program via SDP methods:

$$\text{maximize } \frac{1}{2}\sum_{h(u,v)=1}(1-x_u \cdot x_v) + \frac{1}{2}\sum_{h(u,v)=0}(1+x_u \cdot x_v)$$

$$\text{subject to : for each } v \in V : x_v \cdot x_v = 1 \text{for each } v \in V : x_v \in \mathbb{R}^{|V|}.$$

Select a uniformly random vector r in the $|V|$-dimensional unit sphere and set

$$f(v) = \begin{cases} 0 \text{ if } r \cdot x_v \geq 0 \\ 1 \text{ otherwise} \end{cases}$$

This proof of the claimed approximation performance of the above vector program is obtained by adapting the proof in Section 26.5 of Vazirani (2001) for the MAX-2SAT problem to deal with fact that, in our problem, $a_{ij} = b_{ij} = 1/2$ as opposed to a different set of values in Vazirani (2001). Since there are some subtleties in adapting that proof for readers unfamiliar with this approach, we provide a sketch of the proof in Appendix A. The procedure can be derandomized via methods of conditional probabilities (*e.g.*, see Mahajan and Ramesh (1995)).

5.3. Proof of Theorem 1(c)

For an instance of (G, h) of DLP, construct instance $(G' = (V', E'), h')$ as follows:

$$V' = V \cup \{C_{u,v} | (u,v) \in E \,\&\, h(u,v) = 0\},$$

$$E' = \{e | e \in E \,\&\, h(e) = 1\} \cup \{(u, C_{u,v}), \times (C_{u,v}, v) | (u,v) \in E \,\&\, h(u,v) = 0\},$$

and

$$h'(e) = 1 \text{for all} e \in E'.$$

Note that every odd (resp., even) length circuit in G with respect to weight function h corresponds to an odd (resp., even) length circuit in G' with respect to weight function h', and vice versa. Let F is a set of consistent edges in (G, h) with a vertex labeling function f. Now, observe the following:

(1) F' is a set of consistent edges in (G', h') with a vertex labeling function f' with $f'(x) = f(x)$ for $x \in V' \cap V$ and $f'(C_{u,v}) = f(u) = f(v)$ for an edge $(u,v) \in F$ with $h(u,v) = 0$; thus, an edge (u,v) in F correspond to an edge (u,v) in F' if $h(u,v) = 1$ and correspond to a pair of edges $(u, C_{u,v})$, $(C_{u,v}, v)$ in F' if $h(u,v) = 0$.

(2) If $(u,v) \in E - F$ is an inconsistent edge in (G, h), then the edge $(C_{u,v}, v)$ in G' can always be made consistent by choosing $f'(C_{u,v}) = f(v)$.

Thus, if F'' is the set of consistent edges obtained from F following rules **(1)** and **(2)** above, then $|g(E' - F'')| =$

$|g(E-F)|$ and thus $\mathrm{OPT}_{\mathrm{DLP}}(G', h') = \mathrm{OPT}_{\mathrm{DLP}}(G, h)$. Consider the NDBP problem on $\widehat{G'}$. Any solution to DLP on (G', h') with vertex labeling function f' and set of consistent edges F' cannot contain an odd cycle of consistent edges and thus provides a solution to NDBP on $\widehat{G'}$ of size $|g(E'-F')|$. Thus,

$$\mathrm{OPT}_{\mathrm{NDBP}}(\widehat{G'}) \leq \mathrm{OPT}_{\mathrm{DLP}}(G', h') = \mathrm{OPT}_{\mathrm{DLP}}(G, h).$$

$\mathrm{OPT}_{\mathrm{NDBP}}(\widehat{G'})$ can be approximated in polynomial time to within an approximation ratio of $O(\log|V'|)$ (Lund and Yannakakis, 1993), *i.e.*, we can find a solution $S_{\mathrm{NDBP}}(\widehat{G'})$ in polynomial time such that

$$\begin{aligned} |S_{\mathrm{NDBP}}(\widehat{G'})| &\leq O(\log|V'|) \cdot \mathrm{OPT}_{\mathrm{NDBP}}(\widehat{G'}) \\ &\leq O(\log|V|) \cdot \mathrm{OPT}_{\mathrm{DLP}}(G, h). \end{aligned}$$

Now,

$$\begin{aligned} S_{\mathrm{DLP}}(G, h) &= S_{\mathrm{NDBP}}(G') \\ &\quad \times \cup \{u \mid \exists v \in S_{\mathrm{NDBP}}(G'), (u, v) \in E\}, \end{aligned}$$

is obviously a solution to DLP on (G, h). Recall that $d_{in}^{\max}$ denotes the maximum in-degree of any vertex in G. Thus,

$$\begin{aligned} |S_{\mathrm{DLP}}(G, h)| &\leq d_{in}^{\max} \cdot |S_{\mathrm{NDBP}}(G')| \\ &\leq d_{in}^{\max} \cdot O(\log|V|) \cdot \mathrm{OPT}_{\mathrm{DLP}}(G, h). \end{aligned}$$

6. Examples of applications of the ULP algorithm

We have implemented the SDP-based algorithm for calculating approximate solutions of the undirected labeling problem using Matlab, and we illustrate this algorithm with two applications to biological systems. The first application concerns the relatively small-scale 13-variable digraph of a model of the Drosophila segment polarity network. A second application involves a digraph with 300+ variables associated to the human Epidermal Growth Factor Receptor (EGFR) signaling network. This model was published recently and built using information from 242 published papers. Finally, we provide an example involving a yeast gene regulatory network.

6.1. Drosophila segment polarity

An important part of the development of the early Drosophila (fruit fly) embryo is the differentiation of cells into several stripes (or *segments*), each of which eventually gives rise to an identifiable part of the body such as the head, the wings, the abdomen, etc. Each segment then differentiates into a posterior and an anterior part, in which case the segment is said to be *polarized*. (This differentiation process continues up to the point when all identifiable tissues of the fruit fly have developed.) Differentiation at this level starts with differing concentrations of certain key proteins in the cells; these proteins form striped patterns by reacting with each other and by diffusion through the cell membranes.

A model for the network that is responsible for segment polarity (von Dassow et al., 2000) is illustrated in Fig. 5. As explained above, this model is best studied when multiple cells are present interacting with each other. But it is interesting at the one-cell level in its own right—and difficult enough to study that analytic tools seem mostly unavailable. The arrows with a blunt end are interpreted as having a negative sign in our notation.

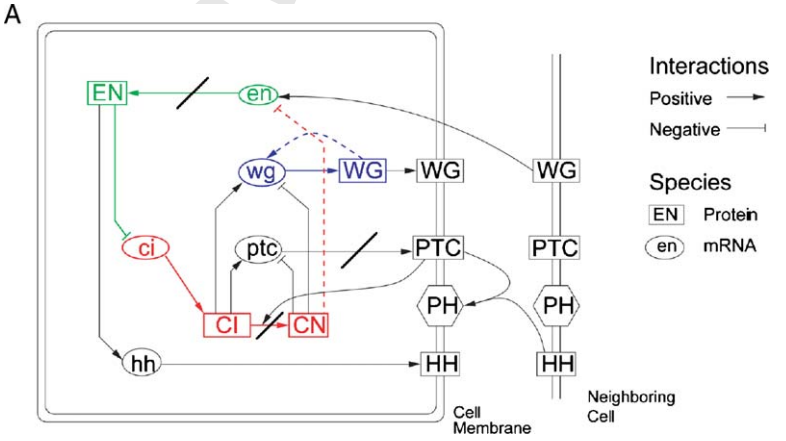

Fig. 5. The network associated to the Drosophila segment polarity, as proposed in von Dassow et al. (2000), Courtesy of N. Ingolia and PLoS. The three edges that have been crossed have been chosen in order to let the remaining edges form an orthant monotone system.

Please cite this article as: Bhaskar DasGupta et al., Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, BioSystems (2006), doi:10.1016/j.biosystems.2006.08.001

Furthermore, the concentrations of the membrane-bound and inter-cell traveling compounds PTC, PH, HH and WG (membrane) on all cells have been identified in the one-cell model (so that, say, HH→ PH is now in the digraph). Finally, PTC acts on the reaction CI→ CN itself by promoting it without being itself affected, which in our notation means PTC$\rightarrow^{+}$ CN *and* PTC$\rightarrow^{-}$ CI.

The implementation. The Matlab implementation of the algorithm on this digraph with 13 nodes and 20 edges produced several partitions with as many as 17 consistent edges. One of these possible partitions simply consists of placing the three nodes ci, CI and CN in one set and all other nodes in the other set, whereby the only inconsistent edges are CL$\rightarrow^{+}$ wg, CL$\rightarrow^{+}$ ptc, and PTC$\rightarrow^{+}$ CN. But note that it is desirable for the resulting open loop system to have as simple remaining loops as possible after eliminating all inconsistent edges. In this case, the remaining directed loops

$$\mathrm{EN} \xrightarrow{-} \mathrm{ci} \xrightarrow{+} \mathrm{CI} \xrightarrow{+} \mathrm{CN} \xrightarrow{-} \mathrm{en} \xrightarrow{+} \mathrm{EN}$$

$$\mathrm{EN} \xrightarrow{-} \mathrm{ci} \xrightarrow{+} \mathrm{CI} \xrightarrow{+} \mathrm{CN} \xrightarrow{-} \mathrm{wg} \xrightarrow{+}$$

$$\mathrm{WG} \xrightarrow{+} \mathrm{WG\,(membrane)} \xrightarrow{+} \mathrm{en} \xrightarrow{+} \mathrm{EN}$$

can still cause difficulties.

A second partition which generated 17 consistent edges is that in which EN, hh, CN, and the membrane compounds PTC, PH, HH are on one set, and the remaining compounds on the other. The edges cut are ptc$\rightarrow^{+}$ PTC, CI$\rightarrow^{+}$ CN and en$\rightarrow^{+}$ EN, each of which eliminates one or several positive loops. By writing the remaining consistent digraph in the form of a cascade, it is easy to see that the only loop whatsoever remaining is wg ↔ WG; this makes the analysis proposed in Enciso and Sontag (2006) easier.

In this relatively low dimensional case we can prove that in fact OPT = 17, as the results below will show.

Lemma 5. *Any partition of the nodes in the digraph in Fig. 5 generates at most 17 consistent edges.*

Proof. From Lemma 3, a simple way to prove this statement is by showing that there are three disjoint cycles with odd weighted length in the network associated to Fig. 5 (disjoint in the sense that no edge is part of more than one of the cycles). Such three disjoint cycles exist in this case, and they are CI-CN-wg, CI-ptc-PTC, CN-en-EN-hh-HH-PH-PTC. □

It is surprising that a realistic biological system with as many as 13 variables and 20 edges can be transformed into a monotone system after the deletion of only 3 nodes. It is conceivable that this restricts the possible dynamics of the system. This is especially the case given that the open loop digraph has almost no closed oriented paths (except for $WG \leftrightarrow wg$), which is evidence that the dynamics of the control system under constant inputs may be especially simple, e.g. such that all solutions converge towards a unique equilibrium.

6.1.1. Multiple copies

It was mentioned above that the purpose of this network is to create striped patterns of protein concentrations along multiple cells. In this sense, it is most meaningful to consider a *coupled* collection of networks as it is given originally in Figs. 6 and 5. Consider a row of k cells, each of which has independent concentration variables for each of the compounds, and let the cell-to-cell interactions be as in Fig. 5 with cyclic boundary conditions (that is, the kth cell is coupled with the first in the natural way). We show that the results can be extended in a very similar manner as before.

Given a partition f_V of the one-cell network considered above, let $\hat{f}_V$ be the partition of the k-cell network defined by $\hat{f}_V(\mathrm{en}_i) := f_V(\mathrm{en})$ for every i, etc. Thus $\hat{f}_V$ consists of k copies of the partition f_V in a natural way.

Lemma 6. *Let f_V be a partition of the nodes of the 1-cell network with n consistent edges. Then with respect*

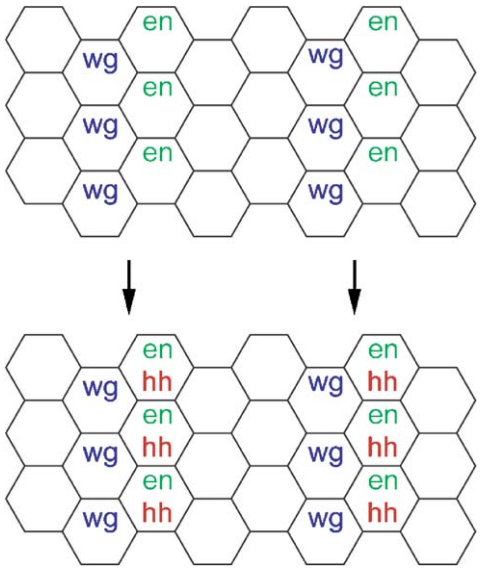

Fig. 6. A diagram of the Drosophila embryo during early development. Each hexagon represents a cell containing a copy of the network in Fig. 6, and neighboring cells interact to form a collective behavior. In this example, an initial striped pattern of the genes en and wg induces the production of the gene hh, but only in those cells that are producing en. This will further strengthen the pattern of stripes and help differentiate the various tissues. Courtesy of N. Ingolia and PLoS (Ingolia, 2004).

Please cite this article as: Bhaskar DasGupta et al., Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, BioSystems (2006), doi:10.1016/j.biosystems.2006.08.001

to the partition $\hat{f}_V$, there are exactly kn consistent edges for the k-cell coupled model.

Proof. Consider the network consisting of *k isolated* copies of the network, that is, *k* groups of nodes each of which is connected exactly as in the one-cell case. Under the partition $\hat{f}_V$, this network has exactly *kn* consistent edges. To arrive to the coupled network, it is sufficient to replace all edges of the form (HH_i, PH_i) by (HH_{i+1}, PH_i) and (WG_i, en_i) by (WG_{i+1}, en_i), $i = 1, \ldots, k$ (where we identify $k + 1$ with 1). Since by definition $\hat{f}_V(HH_{i+1}) = \hat{f}_V(HH_i)$ and $\hat{f}_V(WG_{i+1}) = \hat{f}_V(WG_i)$, the consistency of these edges does not change, and the number of consistent edges therefore remains constant. □

In particular, OPT$\geq 17k$ for the coupled system. The following result will establish an upper bound for OPT.

Lemma 7. *Any partition of the nodes in the digraph in the k-cell coupled network generates at most 17k consistent edges.*

Proof. Consider the signed graph in Fig. 7, which is a sub-digraph of the network associated to Fig. 5. Since the inter-cell edges (WGmem,en) and (HH,PH) are not in this graph, it follows that there are *k* identical copies of it in the *k*-cell model. If it is shown that at least three edges need to be cut in each of these *k* sub-digraphs, the result follows immediately.

Consider the negative cycle ci-CI-wg-CN-en-EN, which must contain at least one inconsistent edge for any given partition. The remaining edges of the subgraph form a tetrahedron with four negative parity triangles, which cannot all be cut by eliminating any single edge. If follows that no two edges can eliminate all negative parity cycles in this signed graph, and that therefore $20k - 3k = 17k$ is an upper bound for the number of consistent edges in the *k*-cell network.

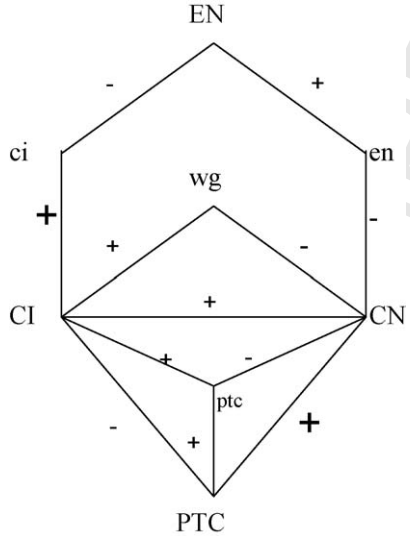

Fig. 7. A sub-digraph of the network in Fig. 5, using the notation defined in the previous sections. Note that this sub-digraph does not include any of the two edges (WGmem,en) and (HH,PH), which connect the networks of different cells in Fig. 5; this will be important in the proof of Lemma 7.

Corollary 1. *For the k-cell linearly coupled network described in* Fig. 5, *it holds* OPT $= 17k$.

Proof. Follows from the previous two results. □

6.2. EGFR signaling

The protein called *epidermal growth factor* is frequently stored in epithelial tissues such as skin, and it is released when rapid cell division is needed (for instance, it is mechanically triggered after an injury). Its function is to bind to a receptor on the membrane of the cells, aptly called the *epidermal growth factor receptor*. The EGFR, on the inner side of the membrane, has the appearance of a scaffold with dozens of docks to bind with numerous agents, and it starts a reaction of vast proportions at the cell level that ultimately induces cell division.

In their May 2005 paper (Oda et al., 2005), Oda et al. integrate the information that has become available about this process from multiple sources, and they define a network with 330 known molecules under 211 chemical reactions. The network itself is available from supplementary material in SBML format (*Systems Biology Markup Language*, http://www.sbml.org), and will most likely be subject to continuous updates. *The implementation.* Each reaction in the network classifies the molecules as reactants, products, and/or modifiers (enzymes). This information was imported into Matlab using the Systems Biology Toolbox. The digraph *G* that is used for this analysis has many more edges than the digraph considered in the digraph displayed in Oda et al. (2005). The reason for this is as follows: if molecules *A* and *B* are both reactants in the same reaction, then the presence of *A* will have an indirect inhibiting effect on the concentration of *B*, since it will accelerate the reaction which consumes *B* (assuming *B* is not also a product). Therefore a negative edge must also appear from *A* to *B*, and vice versa. Similarly, modifiers have an inhibiting effect on reactants.

We thus define *G* by letting sign$(i, j) = 1$ if there exists a reaction in which *j* is a product and *i* is either a reactant or a modifier. We let sign$(i, j) = -1$ if there exists a reaction in which *j* is a reactant, and *i* is also either a reactant or a modifier. Similarly sign$(i, j) = 0$ if the nodes *i*, *j* are not simultaneously involved in any

given reaction, and sign(i, j) is undefined (NaN) if the first two conditions above are both satisfied.

In a few of the reactions of this network there is a modifier or a reactant involved which has an inhibitory effect in the reaction. The effect of this compound on the remaining participants of the reaction is the opposite from that described above. Determining which compounds were inhibitors in the reaction was difficult given the nature of this dataset. Therefore the digraph was corrected by hand in this implementation by looking at the annotations given for each reaction.

An undefined edge can be thought of as an edge that is *both* positive and negative, and it can be dealt with, given an arbitrary partition, by deleting exactly one of the two signed edges so that the remaining edge is consistent. Thus, in practice, one can consider undefined edges as edges with sign 0, and simply add the number of undefined edges to the number of inconsistent edges in the end of each procedure, in order to form the total number of inputs. This is the approach followed here; there are exactly seven such entries in the digraph G.

The results. After running the algorithm several hundred times for this problem, and choosing that partition which produced the highest number of consistent edges, the induced consistent set contained 636 out of 855 edges (ignoring the edges on the diagonal and the 7 undefined edges). See supplementary material for the relevant Matlab functions that carry out this algorithm. A procedure analogous to that carried out for system (5) allows to decompose the system as the feedback loop of a controlled monotone system using $855 - 636 = 219$ inputs. Since the induced consistent set is maximal by definition, Proposition 2 guarantees that the function h is a negative feedback.

Contrary to the previous application, many of the reactions involve several reactants and products in a single reaction. This induces a denser amount of negative and positive edges: even though there are 211 reactions, there are 855 (directed) edges in the 330×330 graph G. It is very likely that this substantially decreases OPT for this system.

The approximation ratio of the SDP algorithm is guaranteed to be at least 0.87 for some r, which gives the estimate $\mathrm{OPT} \leq\approx 636/0.87 \approx 731$ (valid to the extent that r has sampled the right areas of the 330-dimensional sphere, but reasonably accurate in practice).

One procedure that can be carried out to lower the number of inputs is a hybrid algorithm involving *out-hubs*, that is, nodes with an abnormally high out-degree. Recall from the description of the DLP algorithm that all the out-edges of a node x_i can be potentially cut at the expense of only one input u, by replacing all the appearances of x_i in $f_j(x)$, $j \neq i$, by u. We considered the k nodes with the highest out-degrees, and eliminated all the out-edges associated to these hubs from the reaction digraph to form the graph G_1. Then we run the ULP algorithm on G_1 to find a partition f_V of the nodes and a set of m edges that can be cut to eliminate all remaining negative closed chains. Finally, we put back on the digraph those edges that were taken in the first step, and which are consistent with respect to the partition f_V. The result is a decomposition of the system as the negative feedback loop of a controlled monotone system, using at most $k + m$ edges.

An implementation of this algorithm with $k = 60$ yielded a total maximum number of inputs $k + m = 136$. This is a significant improvement over the 226 inputs in the original algorithm. Clearly, it would be worthwhile to investigate further the problem of designing efficient algorithms for the DLP problem to generate improved hybrid algorithmic approaches. The approximation ratios in Theorem 1(c) are not very satisfactory since $d_{in}^{\max}$ and $\log |V|$ could be large factors; hence future research work may be carried out in designing better approximation algorithms.

We conclude with another, more tentative way to drastically reduce the number of inputs necessary to write this system as the negative closed loop of a controlled monotone system. The idea is to make suitable changes of variables in the original system using the mass conservation laws. Such changes of variables are discussed in many places, for example in Volpert et al. (2000), Angeli and Sontag (2003). In terms of the associated digraph, the result of the change of variables is often the elimination of one of the closed chains. The simplest target for a suitable change of variables is a set of three nodes that form part of the same chemical reaction, for instance two reactants and one product, or one reactant, one product and one modifier. It is easy to see that such nodes are connected in the associated digraph by an odd length triangle of three edges.

In order to estimate the number of inputs that can potentially be eliminated by suitable changes of variables, we counted pairwise disjoint, odd length triangles in the digraph of the EGFR network. Using a greedy algorithm to find and tag disjoint negative feedback triangles, we found a maximal number of them in the subgraph associated to each of the 211 chemical reactions. Special care was taken so that any two triangles from different reactions were themselves disjoint. After carrying out this procedure we found 196 such triangles in the EGFR network. This is a surprisingly high number, considering that each of these triangles must have been opened in the ULP algorithm implementation above and that therefore

Please cite this article as: Bhaskar DasGupta et al., Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, BioSystems (2006), doi:10.1016/j.biosystems.2006.08.001

each triangle must contain 1 of the 226 edges cut. To the extent to which most of these triangles can be eliminated by suitable changes of variables, this can yield a much lower number of edges to cut, and it could provide a way to thus stress the underlying structure of the system.

6.3. A yeast regulatory network

As a final example, we run our algorithm on the yeast *Saccharomyces cerevisiae* gene regulatory network from Milo et al. (2002), downloaded from Anon (2006). This network has 690 nodes and 1082 edges, of which 221 are negative and 861 are positive (we labeled the one "neutral" edge as positive; the conclusions will not change if we labeled it negative instead, or we deleted this one edge).

Our algorithm (with 200 randomizations) provides an answer of 43 inconsistent edges, for the best partition found. In other words, it shows that *deleting a mere 4% of edges makes the network consistent*.

Also interesting is the following fact. The original graph has 11 components: a large one of size 664, one of size 5, three of size 3, and six of size 2. All of these components remain connected after edge deletion. The edges deleted all belong to the largest component, and they are incident on a total of 65 nodes in this component.

To better appreciate if this small number of deletions might arise by chance, we also run our algorithm on random graphs having 690 nodes and 1082 edges (chosen uniformly), of which 221 edges (chosen uniformly) are negative. We found that, for such random graphs, about 12.6% (136.6 ± 5) of edges have to be removed in order to achieve consistency. Thus, the number of deletions needed in the biological network is roughly 15 standard deviations away from the mean for random graphs.

It would appear that both the topology (i.e., the underlying graph) and the actual sign assignments contribute to this near-consistency of the yeast network. To justify this remark, we performed the following numerical experiment. We randomly changed the signs of 50 positive and 50 negative edges, thus obtaining a network that has the same number of positive and negative edges, and the same underlying graph, as the original yeast network, but with 100 edges, picked randomly, having different signs. Now, one needs 8.2% (88.3 ± 7.1) deletions, an amount in-between that obtained for the original yeast network and the one obtained for random graphs. Changing more signs, 100 positives and 100 negatives, leads to a less consistent network, with 115.4 ± 4.0 required deletions, or about 10.7% of the original edges, although still not as many as for a random network.

Appendix A. More details on SDP algorithm

In this appendix, we provide details regarding the proof of the SDP algorithm for Theorem 1(b) described in Section 5.2. The proof method is similar to that used in better-known problems. For simplicity, we do not describe the derandomization methods and provide a proof for the expected approximation ratio only. Define the following notations for convenience:

- The vertex set V of the graph for ULP is simply $\{1, 2, \ldots, |V|\}$;
- f_{OPT} is an optimal vertex labeling for ULP with F_{OPT} being the set of consistent edges;
- $\mathrm{SDP}_{\mathrm{OPT}}$ is the maximum value of the objective value of the vector program

$$\begin{aligned}&\text{maximize} \frac{1}{2}\sum_{h(u,v)=1}(1 - x_u \cdot x_v) + \frac{1}{2}\sum h(u,v)\\ &\qquad = 0(1 + x_u \cdot x_v)\\ &\text{subject to : for each} v \in V : x_v \cdot x_v = 1\\ &\text{for each} v \in V : x_v \in \mathbb{R}^{|V|}\end{aligned}$$

It is easy to see that $\mathrm{SDP}_{\mathrm{OPT}} \geq |F_{\mathrm{OPT}}|$ as follows. For every $v \in V$ if $f_{\mathrm{OPT}}(v) = 0$ then set

$$x_v = (1, \underbrace{0, 0, \ldots, 0}_{|V|-1}),$$

whereas if $f_{\mathrm{OPT}}(v) = 1$ then set

$$x_v = (-1, \underbrace{0, 0, \ldots, 0}_{|V|-1});$$

this provides a solution for the vector program with an objective value of precisely $|F_{\mathrm{OPT}}|$. Thus, it suffices if we prove our claim on the approximation ratio relative to $\mathrm{SDP}_{\mathrm{OPT}}$.

Next, note that the vector program can indeed be solved by a SDP approach. Let $Y \in \mathbb{R}^{|V|\times|V|}$ be an unknown real matrix with $y_{i,j}$ denoting the (i, j)th element of Y. It is not difficult to see (via Cholesky decomposition for real symmetric matrices) that the above vector program is equivalent to the followingsemidefinite

Please cite this article as: Bhaskar DasGupta et al., Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, BioSystems (2006), doi:10.1016/j.biosystems.2006.08.001

programming problem:

$$\text{maximize } \frac{1}{2}\sum_{h(u,v)=1}(1-y_{u,v})+\frac{1}{2}\sum_{h(u,v)=0}(1+y_{u,v})$$

$$\text{subject to : for each } v \in V : y_{v,v}=1$$

Y is a positive semidefinite matrix

Such a problem can be solved in polynomial time within an additive error of any constant $\varepsilon > 0$ via ellipsoid, interior-point or convex-programming methods (Alizadeh, 1995; Grötschel et al., 1988; Nesterov and Nemirovskii, 1989, 1994; Vaidya, 1989).

Let $\theta_{u,v}$ denote the angle between the two vectors $x_u, x_v \in \mathbb{R}^{|V|}$ in an optimal solution of the vector program. Then, using standard trigonometric results,

$$\text{SDP}_{\text{OPT}} = \frac{1}{2}\sum_{h(u,v)=1}(1-\cos\theta_{u,v}) + \frac{1}{2}\sum_{h(u,v)=0}(1+\cos\theta_{u,v}). \quad \text{(A.1)}$$

Let W be the expected value of the number of consistent edges of ULP after we have performed the randomized rounding step, namely the step:

select a uniformly random vector r in the $|V|$-dimensional unit sphere;

$$set f(v) = \begin{cases} 0 \text{ if } r \cdot x_v \geq 0 \\ 1 \text{ otherwise} \end{cases}$$

Then, via linearity of expectation, it follows that

$$E[W] = \sum_{h(u,v)=1}\Pr[f(u)\neq f(v)] + \sum_{h(u,v)=0}\Pr[f(u)=f(v)]. \quad \text{(A.2)}$$

Because the vector r was chosen randomly, it is true that

$$\Pr[f(u)\neq f(v)] = \frac{\theta_{u,v}}{\pi} \quad \text{and} \quad \Pr[f(u)=f(v)] = 1-\frac{\theta_{u,v}}{\pi}. \quad \text{(A.3)}$$

Thus,

$$E[W] = \sum_{h(u,v)=1}\frac{\theta_{u,v}}{\pi} + \sum_{h(u,v)=0}\left(1-\frac{\theta_{u,v}}{\pi}\right) \geq \Delta\cdot\left[\frac{1}{2}\sum_{h(u,v)=1}(1-\cos\theta_{u,v}) + \frac{1}{2}\sum_{h(u,v)=0}(1+\cos\theta_{u,v})\right] = \Delta\cdot\text{SDP}_{\text{OPT}} \quad \text{(A.4)}$$

where

$$\Delta = \min\left\{\frac{2}{\pi}\min_{0\leq\theta\leq\pi}\frac{\theta}{1-\cos\theta}, \min_{0\leq\theta\leq\pi}\frac{2-\frac{2\theta}{\pi}}{1+\cos\theta}\right\}$$

can be shown to satisfy $\Delta > 0.87856$ using elementary calculus.

A.1. Proof of lemma 3

Proof. Suppose that the system is monotone with respect to $\leq_{f_V}$, that is,

$$f_V(i)f_V(j)f_E(i,j) = 1 \text{ for all } i, j, \quad i \neq j.$$

(by Lemma 2). Let $V(G) = A \cup B$, where $i \in A$ if $f_V(i) = 1$, and $i \in B$ otherwise. Note that by hypothesis $f_E(i,j) = 1$ if $x_i, x_j \in A$ or if $x_i, x_j \in B$. Also, $f_E(i,j) = -1$ if $x_i \in A$, $x_j \in B$ or vice versa. Noting that every closed chain in G must cross an even number of times between A and B, it follows that every closed chain has parity 1.

Conversely, let all closed chains in G have parity 1. We define a function f_V as follows: consider the partition of $V(G)$ induced by letting $i \sim j$ if there exists an undirected open chain joining i and j. Pick a representative i_k of every equivalence class, and define $f_V(i_k) = 1, k = 1, \ldots, K$. Next, given an arbitrary vertex i and the representative i_k of its connected component, define $f_V(i)$ as the parity (+1 of −1) of any undirected open chain joining i_k with i. To see that this function is well defined, note that any two chains joining i and j can be put together into a closed chain from i_k to itself, which has parity 1 by hypothesis. Thus the parity of both open chains must be the same.

Let now i, j be arbitrary different vertices. If $\partial F_j/\partial x_i \equiv 0$, then (2) is satisfied for i, j; otherwise there is an edge joining i with j. By construction of the "potential" function f_V, it holds that if $f_V(i) = f_V(j)$ then $f_E(i,j) = 1$, i.e., $\partial F_j/\partial x_i \geq 0$, and so (2) holds as well. If $f_V(i) \neq f_V(j)$, then $f_E(i,j) = -1$, i.e. $\partial F_j/\partial x_i \leq 0$. In that case (2) also holds, and the proof is complete. □

Please cite this article as: Bhaskar DasGupta et al., Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, BioSystems (2006), doi:10.1016/j.biosystems.2006.08.001

Appendix B. Supplementary data

Supplementary data associated with this article can be found, in the online version, at 10.1016/j.biosystems.2006.08.001.

References

Alizadeh, F., 1995. Interior point methods in semidefinite programming with applications to combinatorial optimization. SIAM J. Optimiz. 5, 13–51.

Amaldi, E., Kann, V., 1996. On the approximability of some NP-hard minimization problems for linear systems, ECCC Report TR96–015, (available electronically from http://eccc.uni-trier.de/eccc-reports/1996/TR96–015/).

Angeli, D., Ferrell Jr., J.E., Sontag, E.D., 2004. Detection of multistability, bifurcations, and hysteresis in a large class of biological positive-feedback systems. Proc. Natl. Acad. Sci. U.S.A. 101, 1822–1827.

Angeli, D., De Leenheer, P., Sontag, E.D., 2004. A small-gain theorem for almost global convergence of monotone systems. Syst. Contr. Lett. 51, 185–202.

Angeli, D., Sontag, E.D., 2003. Monotone control systems. IEEE Trans. Autom. Contr. 48, 1684–1698.

Angeli, D., Sontag, E.D., 2004. Multistability in monotone I/O systems. Syst. Contr. Lett. 51, 185–202.

Angeli, D., Sontag, E.D., 2004. An analysis of a circadian model using the small-gain approach to monotone systems. Proceedings of the IEEE Conference Decision and Control, Paradise Island, Bahamas, December 2004. IEEE Publications 575–578.

Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M., 1998. Proof verification and hardness of approximation problems. J. ACM 45 (3), 501–555.

Barahona, F., 1982. On the computational complexity of Ising spin glass models. J. Phys. A. Math. Gen. 15, 3241–3253.

Berman, P., Karpinski, M., 2001. Efficient amplifiers and bounded degree optimization, ECCC Report TR01-053, July (available electronically from http://eccc.uni-trier.de/eccc-reports/2001/TR01–053/).

Berman, P., Schnitger, G., 1992. On the complexity of approximating the independent set problem. Inform. Comput. 96, 77–94.

Cinquin, O., Demongeot, J., 2002. Positive and negative feedback: striking a balance between necessary antagonists. J. Theor. Biol. 216, 229–241.

Cipra, B.A., 2000. The Ising Model Is NP-Complete, SIAM News, vol. 33, Number 6, July/August.

Creignou, N., Khanna, S., Sudan, M., 2001. Complexity classifications of Boolean constraint satisfaction problems, SIGACT News, 32 (4), Whole Number 121, Complexity Theory Column 34, 24–33, November.

De Leenheer, P., Angeli, D., Sontag, E.D., 2005. On predator-prey systems and small-gain theorems. J. Math. Biosci. Eng. 2, 25–42.

De Leenheer, P., Malisoff, M., 2006. A small-gain theorem for monotone systems with multivalued input-state characteristics. IEEE Trans. Automat. Contr. 51, 287–292.

De Simone, C., Diehl, M., Junger, M., Mutzel, P., Reinelt, G., Rinaldi, G., 1995. Exact ground states of Ising spin glasses: new experimental results with a branch and cut algorithm. J. Stat. Phys. 80, 487–496.

von Dassow, G., Meir, E., Munro, E.M., Odell, G.M., 2000. The segment polarity network is a robust developmental module. Nature 406, 188–192.

DeAngelis, D.L., Post, W.M., Travis, C.C., 1986. Positive Feedback in Natural Systems. Springer-Verlag, New York.

Enciso, G.A., Smith, H.L., Sontag, E.D., in press. Non-monotone systems decomposable into monotone systems with negative feedback, J. Diff. Eq.

Enciso, G., Sontag, E., 2004. On the stability of a model of testosterone dynamics. J. Math. Biol. 49, 627–634.

Enciso, G., Sontag, E.D., 2005. Monotone systems under positive feedback: multistability and a reduction theorem. Syst. Contr. Lett. 54, 159–168.

Enciso, G., Sontag, E., 2006. Global attractivity, I/O monotone small-gain theorems, and biological delay systems. Discr. Contin. Dynam. Syst. 14, 549–578.

Gedeon, T., Sontag, E.D., 2005. Oscillation in multi-stable monotone system with slowly varying positive feedback, submitted for publication (abstract in Sixth SIAM Conference on Control and its Applications, New Orleans, July).

Goemans, M., Williamson, D., 1995. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J. ACM 42 (6), 1115–1145.

Grötschel, M., Lovász, L., Schrijver, A., 1988. Geometric Algorithms and Combinatorial Optimization. Springer-Verlag, New York, NY.

Hastad, J., Venkatesh, S., 2002. On the advantage over a random assignment. Proceedings of the 34th Annual ACM Symposium on Theory of Computing, 43–52.

Hirsch, M., 1985. Systems of differential equations that are competitive or cooperative. II. Convergence almost everywhere. SIAM J. Math. Anal. 16, 423–439.

Hirsch, M., 1983. Differential equations and convergence almost everywhere in strongly monotone flows. Contemp. Math. 17, 267–285.

http://www.weizmann.ac.il/mcb/UriAlon/Papers/networkMotifs/yeastData.mat.

Ingolia, N., 2004. Topology and robustness in the drosophila segment polarity network. Publ. Libr. Sci. 2 (6), 0805–0815.

Istrail, S., 2000. Statistical mechanics, three-dimensionality and np-completeness. I. Universality of intractability of the partition functions of the ising model across non-planar lattices. Proceedings of the 32nd ACM Symposium on the Theory of Computing (STOC00), Portland, Oregon, May 21–23, 2000. ACM Press 87–96.

Klamt, S., 2006. Generalized concept of minimal cut sets in biochemical networks. Biosystems 83, 233–247.

Lewis, J., Slack, J.M., Wolpert, L., 1977. Thresholds in development. J. Theor. Biol. 65, 579--590.

Lund, C., Yannakakis, M., 1993. The approximation of maximum subgraph problems. Proceedings of the International Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science, 700, Springer-Verlag.

Mahajan, S., Ramesh, H., 1995. Derandomizing semidefinite programming based approximation algorithms. Proceedings of the 37th Annual IEEE symposium on Foundations of Computer Science, 162–169.

Ma'ayan, A., Iyengar, R., Sontag, E.D., in preparation. Sign-consistency loops detected in biochemical regulatory networks may provide dynamical stability.

Meinhardt, H., 1978. Space-dependent cell determination under the control of morphogen gradient. J. Theor. Biol. 74, 307--321.

Please cite this article as: Bhaskar DasGupta et al., Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, BioSystems (2006), doi:10.1016/j.biosystems.2006.08.001

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Alon, D.U., 2002. Network motifs: simple building blocks of complex networks. Science 298, 824–827.

Monod, J., Jacob, F., 1961. General conclusions: telenomic mechanisms in cellular metabolism, growth, and differentiation. Cold Spring Harbor Symp. Quant. Biol. 26, 389--401.

Murray, J.D., 2002. Mathematical Biology. I. An introduction. Springer, New York.

Nesterov, Y., Nemirovskii, A., 1989. Self-Concordant Functions and Polynomial Time Methods in Convex Programming. Central Economic and Mathematical Institute, USSR Academy of Science.

Nesterov, Y., Nemirovskii, A., 1994. Interior Point Polynomial Methods in Convex Programming. Society of Industrial and Applied Mathematics, Philadelphia, PA.

Oda, K., Matsuoka, Y., Funahashi, A., Kitano, H., 2005. A comprehensive pathway map of epidermal growth factor receptor signaling. Mol. Syst. Biol. doi:10.1038/msb4100014.

Papadimitriou, C.H., Yannakakis, M., 1991. Optimization, approximation, and complexity classes. J. Comput. Syst. Sci. 43 (3), 425–440.

Plathe, E., Mestl, T., Omholt, S.W., 1995. Feedback loops, stability and multistationarity in dynamical systems. J. Biol. Syst. 3, 409--413.

Remy, E., Mosse, B., Chaouiya, C., Thieffry, D., 2003. A description of dynamical graphs associated to elementary regulatory circuits. Bioinformatics 19 (Suppl. 2), ii172--ii178.

Smith, H.L., 1995. Monotone Dynamical Systems. AMS, Providence, R.I.

Smith, H.L., 1988. Systems of ordinary differential equations which generate an order-preserving flow: a survey of results. SIAM Rev. 30, 87–111.

Snoussi, E.H., 1998. Necessary conditions for multistationarity and stable periodicity. J. Biol. Syst. 6, 3–9.

Sontag, E.D., 1990. Mathematical Control Theory: Deterministic Finite Dimensional Systems, 2nd ed. 1998. Springer, New York.

Sontag, E.D., 2004. Some new directions in control theory inspired by systems biology. Syst. Biol. 1, 9–18.

Sontag, E.D., 2005. Molecular systems biology and control. Eur. J. Contr. 11, 396–435.

Sontag, E.D., in preparation. Consistency of indirect effects in biological networks.

Thomas, R., 1978. Logical analysis of systems comprising feedback loops. J. Theor. Biol. 73, 631--656.

Vaidya, P., 1989. A new algorithm for minimizing convex functions over convex sets. Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science 338–343.

Vazirani, V.V., 2001. Approximation Algorithms. Springer-Verlag, Berlin.

Volpert, A.I., Volpert, V.A., Volpert, V.A., 2000. Traveling Wave Solutions of Parabolic Systems, volume 140 of Translations of Mathematical Monographs, AMS.

Please cite this article as: Bhaskar DasGupta et al., Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, BioSystems (2006), doi:10.1016/j.biosystems.2006.08.001