Determining the distance to monotonicity of a biological network: a graph-theoretical approach

G. Iacono[†], F. Ramezani[‡], N. Soranzo[†] and C. Altafini[†*]

[†] SISSA Int. School for Advanced Studies
via Beirut 2-4, 34014 Trieste, Italy
[‡] Max-Planck-Institut für Informatik,
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany

August 31, 2009

Abstract

We use ideas from graph theory in order to determine how distant is a given biological network from being monotone. On the signed graph representing the system, the minimal number of sign inconsistencies (i.e., the distance to monotonicity) is shown to be equal to the minimal number of fundamental cycles having a negative sign. Suitable operations aiming at computing such number are also proposed and shown to outperform all algorithms so far existing for this task.

Keywords: biological networks, monotone systems, consistency deficit, signed graphs, fundamental cycles.

1 Introduction

In a series of papers [4, 10, 20, 19], E. Sontag and colleagues have drawn to the attention of the Systems Biology community the concept of monotone systems. Such systems have very useful dynamical properties [17], like the tendency of their solution to converge to an equilibrium (a bounded trajectory generically converges to an equilibrium) and the lack of "chaotic" behavior. The link with biological systems arise from the observation that, although nonlinear and complex, these systems typically show highly predictable and ordered dynamical behavior, and a tendency to remain at equilibrium or to robustly return to it when perturbed. Sontag's suggestion is that biological systems might have evolved so as to be, if not monotone, at least near monotone [10, 20]. Monotonicity in dynamical systems is a well-studied property [7, 17], and can be stated in several alternative ways. For biological networks, a very useful way to formulate/verify it is in terms of the sign of all possible feedback loops among the variables of the system. Given an arbitrary system of Ordinary Differential Equations (ODEs), consider the graph corresponding to the matrix of signs associated to the terms of the Jacobian matrix (assumed for simplicity to be constant regardless of the point in state space we consider). Looking only at the signs of the Jacobian gives a basic indication of the effect (activatory/inhibitory) of a variable on another variable. In most biological contexts, this information is the best one can hope to obtain, as too little is known of the functional form of the ODE and of its dependence

*Corresponding author: altafini@sissa.it

on concentrations, parameters, external conditions, hidden (non-modeled) variables, etc. For the undirected graph corresponding to the symmetrization of this signed adjacency matrix, the monotonicity property corresponds to all undirected cycles having positive sign, where the sign of a cycle is computed as the product of the signs of the edges forming the cycle. Undirected cycles may correspond to "true" oriented feedback loop or to e.g. distinct paths connecting pairs of nodes [20].

It is argued in [20] that biological networks are "near monotone" in the sense that a relatively small number of sign changes in edges is enough to make the graph monotone. Closely related to this idea is the intuition that biological networks may have many more positive cycles than negative ones, which is the approach taken in [10]. While the simple verification of whether a network is monotone or less is feasible in polynomial-time, the problem of testing how distant a given network is from monotonicity (i.e. estimating the "consistency deficit" in the terminology of [20]) is an NP-hard one [4]. As the size of a network becomes of the order of the thousands of nodes, like for example in any gene regulatory network, testing exhaustively the sign of all cycles quickly becomes an untreatable problem, because the number of cycles grows exponentially. In fact, in [10] only short cycles were tested for large networks, while in [4] approximation algorithms based on semidefinite programming ideas were introduced.

The purpose of this paper is to tackle this problem from a different perspective, using tools from graph theory, namely the notion of fundamental cycles. The concept of a fundamental cycle was introduced by Kirchhoff [9]. What Kirchhoff showed is that no matter how many cycles a network contains, considering only fundamental cycles with respect to a spanning tree is enough as the rest of the cycles are obtained as linear combinations of some fundamental cycles. In terms of linear algebra, fundamental cycles form a basis of a vector space whose elements are cycles and disjoint unions of cycles. We show that in particular fundamental cycles of positive sign form a subspace which is invariant to the positivity property: any cycle of this subspace must have a positive sign, and the cycle subspaces obtained in this way correspond to monotone subsystems. For the negative fundamental cycles, this property does not hold (the reason being that elements of the subspace are not only cycles but also disjoint unions of cycles). However, the number of negative fundamental cycles corresponds to the number of sign changes that are required to render the network monotone. In fact, each fundamental cycle is uniquely associated to a chord not shared with any other fundamental cycle. By changing sign to the chords of all negative fundamental cycles we obtain a monotone graph.

As an easy byproduct, we get an upper bound on the number of inconsistencies of a network: *any* network can be rendered monotone by at most a number of sign changes equal to the cardinality of a basis of fundamental cycles. Unrelated (and usually sharper) upper bounds can also be obtained from the theory of signed graphs [18]. These bounds are quite helpful in defining a proper metric to test whether a given network can be classified as "near-monotone".

If all bases of fundamental cycles have the same cardinality, the number of positive/negative fundamental cycles in a given basis depends however on the choice of spanning tree (and can vary widely with it). Needless to say, testing all spanning trees requires computational time that grows exponentially with the size of the graph. In order to simplify the choice of a "good" spanning tree (with fewest possible negative fundamental cycles, hence as near as possible to monotonicity), we propose an algorithm which acts on cut sets and aims at maximizing the overall number of positive edges on the graph while maintaining unaltered the sign of each cycle. The *rationale* of the method is that changes of sign through a cut set leave the consistency deficit invariant. In the theory of monotone systems [17], this operation corresponds to changing sign to the order relationship in one or more orthants; in the theory of signed graphs [22] this corresponds to changing the representative element in a "switching class of equivalence" where the consistency deficit is an invariant of the equivalence relation.

Though this algorithm is heuristic, its performance is excellent both in terms of computational time and of improving previously known upper bounds on the consistency deficit [4, 8]. The signed adjacency matrix resulting from the iterated application of such an algorithm has usually many less minus signs than the original system, while being equivalent to it in terms of the consistency deficit. It is then possible to search for a lower bound on the consistency deficit assigning to each residual minus sign a negative cycle in such a way that all these negative cycles are edge disjoint. Also for this lower bound our algorithm improves the existing results of [4, 8].

As examples, we consider four large biological networks (two transcriptional regulatory networks and two signaling networks) and compute upper and lower bounds for the consistency deficit. Comparing the consistency deficit as estimated by the algorithms with the theoretically predicted maximal values for the upper bound, we obtain that the two transcriptional networks are much more monotone than the two signaling ones. We interpret this difference in terms of abundance of negative short cycles in the two signaling networks, abundance which in turns originates from the stoichiometric level of detail used in the construction of these signed networks.

The structure of the paper is as follows. In Section 2 we review the required background material on monotone systems and on graph theory; in Section 3 we formulate the notion of consistency deficit in graph-theoretical terms, and in Section 4 we provide the algorithms for the computation of such a measure. Finally, in Section 5 the algorithms are applied to biological networks taken from the literature.

2 Background material

2.1 Graph theory and cycles

A basic reference for this Section is [5]. An undirected *graph* G is an ordered pair (V, E) consisting of a set of nodes $V = \{v_1, v_2, ..., v_n\}$ and one of edges $E = \{e_1, e_2, ..., e_m\}$, such that each edge e_k is identified with an unordered pair (v_i, v_j) of vertices. The number of edges incident on a vertex v_i is called the *degree of vertex* $v_i, d(v_i)$. A *path* is defined as a finite alternating sequence of vertices and edges, beginning and ending with vertices, such that each edge is incident with the vertices preceding and following it and such that no edge and no vertex (except, perhaps for the first and last) appears more than once. A closed path is called a *cycle*. Clearly, in a subgraph which is a cycle every vertex is of degree two.

A *tree* is a connected graph without any cycle. A tree with n vertices has $n-1$ edges. A tree T is said to be a *spanning tree* of a connected graph G if T is a subgraph of G and T contains all vertices of G. If we add an edge between any two vertices of a tree, a cycle is created. This is because there already exists one path between any two vertices of a tree; adding an edge between them creates an additional path and hence a cycle. Every connected graph has at least one spanning tree. An edge in a spanning tree is called a *branch* of T and an edge of G that is not in a given spanning tree T is called a *chord*. With respect to any spanning tree, a graph G with n vertices has $n-1$ branches and $\mu = m - n + 1$ chords, where $m = |E|$ (number of edges of G).

Consider a spanning tree T in a connected graph G. Adding any chord to T will create exactly one cycle. Such a cycle formed by adding a chord to a spanning tree, is called a *fundamental cycle*. A connected graph G has exactly μ fundamental cycles with respect to any spanning tree T, because each spanning tree has μ chords and each chord creates its own cycle. A given cycle may be fundamental with respect to one spanning tree but not with respect to another spanning tree of the same graph.

The *ring sum of two graphs* G_1 and G_2 (written as $G_1 \oplus G_2$) is a graph consisting of the vertex set $V_1 \cup V_2$ and edges that are either in G_1 or in G_2, but not in both. For example $G \oplus G$ is a graph without any edge.

Consider a graph G with m edges. Any subgraph of G can be represented by an m-tuple representing a subset of edges. There are 2^m such m-tuples possible, including the zero vector which represents the null graph and $(1, \ldots, 1)$ which is G itself. The ring sum between two subgraphs corresponds to the modulo 2 addition between the two tuples representing the two subgraphs. The set of all m-tuples forms a *vector space* W_G with $GF(2)$ as field. The natural basis for this vector space W_G is a set of m linearly independent vectors, each representing a subgraph consisting of one edge of G, so $\dim(W_G) = m$. The vector space operation restricted to cycles yields the following Theorem.

Theorem 1 *The ring sum of two cycles in a graph G is either a cycle or an edge disjoint union of cycles.*

A *cycle vector* is a vector in W_G representing either a cycle or a union of edge disjoint cycles in a graph G. Therefore the linear combination of two cycle vectors is also a cycle vector, so this set is closed with respect to both addition and scalar multiplication and hence it is a vector subspace of W_G, call it W_Γ. The set of cycle vectors corresponding to the set of fundamental cycles with respect to any spanning tree, forms a basis for the cycle subspace W_Γ ([5], Thm 6.6), so $\dim(W_\Gamma) = \mu$.

Assume G is connected. A *cut set* is a set of edges whose removal from G leaves G disconnected, provided removal of no proper subset of these edges disconnects G. Every cut set in a connected graph G must contain at least one branch of every spanning tree of G. We shall make use of the following Theorem.

Theorem 2 *Every cycle has an even number of edges in common with any cut set.*

A graph (V, E) is signed if its edges are endowed with a sign: $e_i \in \{\pm 1\}$. A cycle of a signed graph is positive if it contains an even number of negative edges, it is negative otherwise.

2.2 Monotone systems and positive cycles

The material of this Section is taken from [20]. The system considered is described by the time-dependent vector $x(t) = [x_1(t), ..., x_n(t)]^T$ whose components x_i may represent concentrations of chemical species such as proteins, RNA or metabolites. We assume that $x_i(t)$ are nonnegative: $x_i(t) \in \mathbb{R}_+$, $\forall t$. The model we consider is a system of autonomous differential equations describing the rate of change of each variable as a function of the concentrations of all the variables:

$$\frac{dx}{dt} = f(x), \tag{1}$$

where $f : \mathbb{R}^n_+ \to \mathbb{R}^n$ is a vector function with components f_i. The species graph G associated with the Jacobian of the system (1) is a signed graph with n nodes $v_1, ..., v_n$, one node for each species x_i. No edge is drawn from node v_j to node v_i if the partial derivative $\partial f_i / \partial x_j$ vanishes identically for any x, meaning that there is no direct effect of the jth species upon the ith species. If this derivative is not identically zero, then there are three possibilities: $\partial f_i / \partial x_j \geq 0$ for all x, or $\partial f_i / \partial x_j \leq 0$ for all x, or $\partial f_i / \partial x_j$ changes sign depending on the particular entries of the concentrations vector x. For the sake of simplicity we do not consider this third case. In the first case (activation), we draw an edge labeled $J_{ij} = +1$ from v_j to v_i, in the second case (inhibition) we draw this edge with label $J_{ij} = -1$. Following [20], self-loops (i.e., edges starting and ending on the same v_i) are ignored: $J_{ii} = 0$. Again following [20], we symmetrize

the signed adjacency matrix by imposing $J_{ji} = J_{ij}$. In all the real networks we treat in Section 5, this operation (which yields an undirected graph G) leads to very few inconsistent edges pairs which can be disregarded. Denote by J the resulting $n \times n$ symmetric sign adjacency matrix, of elements $J_{ij} \in \{0, +1, -1\}$. A *spin assignment* (or signature) σ for the graph G is a labeling of each node v_i with a number σ_i equal to $+1$ or -1. If there is an edge from node v_j to node v_i, with label $J_{ij} \in \{+1, -1\}$, we say that this edge is *consistent with the spin assignment* σ provided that

$$J_{ij}\sigma_i\sigma_j = 1.$$

We say that σ is a *consistent spin assignment for the signed graph* G if every edge of G is consistent with σ. In other words, if for a pair of vertices v_i and v_j there is a positive edge from node v_j to node v_i then v_j and v_i must have the same spin, and if there is a negative edge connecting v_j to v_i then v_j and v_i must have opposite spins. A signed graph is said to be *consistent* if there exists a consistent sign assignment for it.

Lemma 1 *A signed graph G is consistent if and only if every undirected cycle in G has a positive sign.*

Given any partial order $\leq$ defined on $\mathbb{R}^n$, the system (1) is said to be *monotone with respect to* $\leq$ if $x_0' \leq x_0''$ implies $x'(t) \leq x''(t)$ for every $t \geq 0$. Here $x'(t)$ and $x''(t)$ are solutions of system (1) with initial conditions x_0' and x_0'' respectively. A useful way to define partial orders in $\mathbb{R}^n$ is through a vector $z = (z_1, ..., z_n)$, where $z_i \in \{1, -1\}$. We say that $x \leq_z y$ if $z_i x_i \leq z_i y_i$ for every i.

Lemma 2 ***(Kamke condition)*** *Consider an order $\leq_z$ generated by $z = (z_1, ..., z_n)$. The system (1) is monotone with respect to $\leq_z$ if and only if*

$$z_i z_j \partial f_j / \partial x_i \geq 0, \qquad i, j = 1, ..., n, \quad i \neq j.$$

Consider the species graph G for the system (1) with order $\leq_z$ generated by z and let the spin assignment σ be equal to z. Then $z_i z_j \partial f_j / \partial x_i \geq 0$ if and only if

$$J_{ij} z_i z_j = 1. \tag{2}$$

In other words, the system (1) is monotone with respect to $\leq_z$ if and only if z is a consistent spin assignment for the species graph G.

Corollary 1 *Consider an order $\leq_z$ generated by $z = (z_1, ..., z_n)$. If the system (1) is monotone with respect to z then every undirected cycle in the associated graph G has positive sign.*

A dynamical system is said to be *monotone* if there exists at least one consistent spin assignment for its associated graph G.

3 Fundamental cycles and inconsistencies

Consider a dynamical system $f(\cdot)$ and its associated symmetric signed graph G of adjacency matrix J. Checking whether $f(\cdot)$ is monotone or less, i.e., whether G is consistent or less, is a simple task, verifiable in polynomial-time. A possible approach is based on considering the Laplacian $L = D - J$, where D is a diagonal matrix of elements $d(v_i)$, $i = 1, \ldots, n$. The signed graph G is consistent if and only if L is singular, i.e., 0 is an eigenvalue of L [22]. In general, for any given system $f(\cdot)$ the corresponding signed graph G will not be consistent, although in a biological context it might be "near-monotone", i.e closer to monotone than expected by

random edge assignments, as claimed in [20]. Our goal is to identify the smallest number of edges such that if we change their signs the new graph obtained is consistent, and the tool we use for this scope is an extension of the theory of fundamental cycles for signed graphs.

Lemma 3 *Let $\{F_1, F_2, ..., F_\mu\}$ be the fundamental cycles associated to a given spanning tree of G. A nonempty intersection of an arbitrary number of these fundamental cycles is a path.*

Proof. The proof is by induction on the number of fundamental cycles in the intersection. If $s = 1$, the intersection is just the fundamental cycle, which is obviously a path.

For $s \geqslant 2$, assume the statement true for any r, $r < s$, we want to show it for $r = s$. By induction $F_1 \cap F_2 \cap ... \cap F_{s-1}$ is a path, call it H_1. Assuming $F_1 \cap F_2 \cap ... \cap F_s$ not connected, let e_1 and e_2 be two edges belonging to two disconnected paths of the intersection $F_1, ..., F_s$. Since $F_1 \cap F_2 \cap ... \cap F_s \subseteq F_1 \cap F_2 \cap ... \cap F_{s-1}$ then $e_1, e_2 \in H_1$. So there is a path between e_1 and e_2 in H_1. Since e_1 and e_2 are also in F_s, there must exist another path between them in F_s such that it does not contain the chord of F_s, call it H_2. Now omitting all chords from the graph, we must obtain a graph without cycles, however $H_1 \cup H_2 \cup e_1 \cup e_2$ contains a cycle and hence we have a contradiction. In fact since each fundamental cycle has its own chord, chords are not in the intersection of fundamental cycles, so after omitting all chords H_1 remains. Furthermore each fundamental cycle has only one chord and since H_2 does not contain the chord of F_s, H_2 still exists after we have deleted all chords. Therefore we have two different paths between e_1 and e_2, hence a cycle and a contradiction. We must conclude that $F_1 \cap F_2 \cap ... \cap F_s$ is a path. ■

The elements of the vector space W_Γ are cycles *and* disjoint unions of cycles. As a matter of fact, the ring sum of two cycles can consist of more than one cycle. As a consequence, while the sign of a cycle vector obtained in this ring sum is always univocally defined, in general it does not provide any information on the sign of the (edge disjoint) cycles in which the cycle vector can be decomposed. Lemma 3 suggests that for fundamental cycles, the situation is to some extent simplified. In fact, we show in the following that subspaces composed entirely of positive cycles can be obtained only by looking at the signs of the basis elements.

Lemma 4 *The ring sum of two positive or two negative nondisjoint fundamental cycles is a positive cycle and the ring sum of one positive and one negative nondisjoint fundamental cycles is a negative cycle.*

Proof. The Lemma is proved for the case of both positive cycles, the other cases being similar. By assumption F_1 and F_2 have a nontrivial intersection, then by Lemma 3 this intersection is a path, call it H_1. Hence there exist paths H_2 and H_3 such that $F_1 = H_1 \cup H_2$ and $F_2 = H_1 \cup H_3$, from which $F_1 \oplus F_2 = H_2 \cup H_3$ is a cycle. With regard to the sign of H_1, we have two cases.

1. If H_1 has positive sign, then H_2 and H_3 must have positive sign and so must $F_1 \oplus F_2$.
2. If H_1 has negative sign, then H_2 and H_3 must have negative sign. Also in this case $F_1 \oplus F_2$ must have positive sign.

■

The following theorem shows that part of the result of Lemma 4 extends to an arbitrary number of positive fundamental cycles.

Theorem 3 *The ring sum of s positive fundamental cycles is a positive cycle or a disjoint union of positive cycles.*

Proof. The proof is by induction. Letting $C = \oplus_{i=1}^{r} F_i$, if $r = 2$ then Lemma 4 holds and the cycle is positive. By induction, we assume that $F_1 \oplus ... \oplus F_r$, $r < s$, is a positive cycle or a disjoint union of positive cycles, and we prove it for the case $r = s$. Since the set of all cycle vectors forms a subspace in W_G, this set is closed under the operation of ring sum and therefore the ring sum of arbitrary cycles or disjoint union of cycles is either a single cycle or a disjoint union of cycles. Consider the two cases.

1. Let $C = \oplus_{i=1}^{s} F_i$ be a single cycle. Letting $B = \oplus_{i=1}^{s-1} F_i$, the inductive hypothesis is that B is a positive cycle or a disjoint union of positive cycles, and we have $C = B \oplus F_s$, where F_s by assumption is a positive cycle. Denoting the edges in common between B and F_s by A, since F_s has positive sign, A and $(F_s - A)$ must have the same sign. In the ring sum we omit the common edges, i.e., we have $C = (B - A) \cup (F_s - A)$. The sign of the cycle C is the product of the signs of the edges in $(B - A)$ multiplied by the product of the edges in $(F_s - A)$. But this is the same as the product of the signs of the edges in $(B - A)$ multiplied by the product of the edges in A, that is the product of edges in B. As B is composed of positive cycles the sign of B is positive, hence C must have a positive sign.

2. Let $C = \oplus_{i=1}^{s} F_i$ be a disjoint union of ℓ cycles $C = C_1 \oplus ... \oplus C_\ell$, $\ell \geq 2$. Then for each C_i, $i = 1, .., \ell$, we can write $C_i = F_{i_1} \oplus ... \oplus F_{i_q}$ where $q < s$ and $C_i \cap C_j = \emptyset \; \forall \; 1 \leqslant i \neq j \leqslant \ell$. Denote the chord of each $F_i, 1 \leq i \leq s$, by e_i. Each C_i must contain some chords belonging to $e_1, ..., e_s$, since if we have a cycle C_i without any of the chords $e_1, ..., e_s$, after omitting all the chords C_i remains a cycle and hence we have a contradiction. Also, if C_i contains the chords $e_{i_1}, ..., e_{i_q}$, $q < s$, then it is uniquely identified by $F_{i_1} \oplus ... \oplus F_{i_q}$ (the ring sum of fundamental cycles corresponding to its chords). $F_{i_1} \oplus ... \oplus F_{i_q}$, by Theorem 1, is a cycle or a disjoint union of cycles and so $(F_{i_1} \oplus ... \oplus F_{i_q}) \oplus C_i$ is also a cycle or a disjoint union of cycles, and both of them contain only the chords $e_{i_1}, ..., e_{i_q}$, therefore $C_i = F_{i_1} \oplus ... \oplus F_{i_q}$, otherwise we would have a cycle without any chords which is another contradiction. So $F_1 \oplus .... \oplus F_s$ is the disjoint sum $C_1 \oplus ... \oplus C_\ell$ in which each C_i can be written as a sum of q fundamental cycles, $C_i = F_{i_1} \oplus ... \oplus F_{i_q}$ where $q < s$ and by induction each C_i has positive sign.

■

Corollary 2 *If there exists a spanning tree such that every cycle in the fundamental basis it induces has positive sign, then the system is monotone.*

Proof. From the previous Theorem, a fundamental basis of cycles having positive signs implies that every cycle (and every disjoint union of cycles) has positive sign. ■

Theorem 3 essentially affirms that the cycle subspace generated by positive fundamental cycles is invariant with respect to the positivity property, i.e., each cycle (also those obtained as disjoint unions of other cycles) in this subspace must have a positive sign. In general, nothing can be said about the sign of the ring sum of fundamental cycles if some of them have negative sign. In fact, as long as this ring sum yields a single cycle, then arguments such as Lemma 4 still hold, but they are in general not applicable for a disjoint union of cycles (in a disjoint union of cycles there could be two disjoint cycles each having a negative sign, but still the cycle vector would be positive).

Given G with μ fundamental cycles of which ν have positive sign and $\mu - \nu$ have negative signs, one way to render the entire system monotone is to change the sign of the last $\mu - \nu$ fundamental cycles. This can be achieved simply by changing sign on the $\mu - \nu$ chords that identify the fundamental cycles having negative sign.

Corollary 3 *Any signed graph G having a μ-dimensional fundamental cycle basis characterized by ν cycles having positive sign and $\mu - \nu$ having negative sign, can be rendered consistent by exchanging the signs of the $\mu - \nu$ chords corresponding to the fundamental cycles having negative sign.*

Of course, the worst case is when all fundamental cycles are negative, i.e., any G can be rendered consistent with at most μ sign changes. From the theory of signed graphs, see [18], we also have another worst-case upper bound on the consistency deficit, $\eta = (m - \sqrt{m})/2$. Hence we have the following Proposition.

Proposition 1 *Any signed graph G can be rendered consistent with at most* $\min(\mu, \eta)$ *sign changes in its edges.*

The two values for the upper bound are unrelated: for very sparse networks (with average connectivity of a node < 2) then $\mu < \eta$, viceversa for more dense networks. While the value of μ is always attainable in a network, it is not clear from the literature in which cases η is achievable as a worst-case upper bound.

In general the sign associated with a basis of fundamental cycles is not invariant to changes of basis (i.e. of spanning tree). Therefore if we can find a fundamental cycle basis with fewer negative cycles, we have to do fewer changes of sign in order to obtain a monotone system. The following Proposition is the starting point of a series of algorithms aiming at "simplifying" the graph by changing its signs in a suitable equivalence class in which the monotonicity properties and the number of inconsistencies are preserved.

Proposition 2 *Exchanging the sign of the edges through a cut set preserves the sign of each cycle of a given graph.*

Proof. From Theorem 2, every cycle of G intersects a cut set in an even number of edges and hence a sign change through an entire cut set does not alter the sign of a cycle. ∎

In the literature about signed graphs, operations of sign change through a cut set are called "switching equivalences" [22], while in the statistical physics literature they are called "gauge transformations" [21].

Proposition 2 admits a simple linear-algebraic interpretation. It is known ([17] Lemma 2.1) that a symmetric matrix J is monotone with respect to an order $\leq_\sigma$ if and only if $P_\sigma J P_\sigma$ with $P_\sigma = \mathrm{diag}(\sigma)$ has all nonnegative off-diagonal elements, see (2), where the signature σ of the order corresponds to a spin assignment to the nodes of σ. A cut set s induces a bipartite partition of the vertices V of G into the subsets V_1 and V_2. Switching sign through the cut set s corresponds to changing sign in the spin assignment of one (and only one) of the subsets of vertices, say V_1. For example, for a graph partitioning V into $V_1 = \{v_1, \ldots, v_k\}$ and $V_2 = \{v_{k+1}, \ldots, v_n\}$, $\sigma = \{\underbrace{-1, \ldots, -1}_{k \text{ times}}, 1, \ldots, 1\}$. After the switching through s (i.e. the multiplication $P_\sigma J P_\sigma$), any edge (v_i, v_j) such that $i, j \leqslant k$ or $i, j > k$ maintains the same sign, while (v_i, v_j) such that $i \leqslant k$ and $j > k$ switches sign. Denote by $G(\sigma)$ the signed graph G with spin assignment σ (G without an explicit σ obviously corresponds to $G(\{1, \ldots, 1\})$, i.e. the "all spins up" assignment). For the cut set s, let d_s^+ , d_s^- be, respectively, the number of $+$ and $-$ signs through s. Likewise, $d^+_{G(\sigma)}$ and $d^-_{G(\sigma)}$ are the total number of $+$ and $-$ signs on the edges of $G(\sigma)$. If we select a cut set s such that $d_s^+ < d_s^-$, then, from Proposition 2, switching sign through s results in a new signed graph with the same monotonicity properties as the original one, but with a lower number of $-$ signs overall. If we can find a spin assignment σ such that $d^-_{G(\sigma)}$ is a global minimum, then the residual $-$ signs of $G(\sigma)$ will all be unremovable violations of consistency. Denoting $\mathcal{S}$ the set of all cut sets of G, we have in fact the following.

Proposition 3 *Under sign changes through cut sets, $d^-_{G(\sigma)}$ is minimal with respect to all σ if and only if $d^+_s \geqslant d^-_s \ \forall\ s \in \mathcal{S}$.*

Proof. By contradiction, assume that in correspondence of the minimum number of $-$ signs $d^-_{G(\sigma)}$ there exists $\bar{s} \in \mathcal{S}$ such that $d^+_{\bar{s}} < d^-_{\bar{s}}$. Changing sign through $\bar{s}$ the number of $-$ signs on $G(\sigma)$ decreases and hence it cannot be minimal. For the other direction the claim follows from the definition of minimal ("minimal under sign changes through cut sets"). In fact, assuming that the number of $-$ sign is not minimal means that there must exist one or more cut sets that allow to decrease that total number. But this is impossible if $d^+_s \geqslant d^-_s \ \forall\ s \in \mathcal{S}$. ∎

Denote $\delta = \min_\sigma d^-_{G(\sigma)}$ the consistency deficit, i.e., the global minimum number of sign inconsistencies of the graph G (also called "distance to monotonicity" or "frustration index" in the context of spin systems).

Needless to say, testing sign changes through all $s \in \mathcal{S}$ is computationally hard, hence it is not a viable method to compute δ. In the following Section we provide a few heuristic algorithms to do this calculation in a reasonable time.

4 Computing a minimal number of sign inconsistencies: algorithms

Computing how close a given signed network is to a monotone network is an NP-hard problem, equivalent to the well-known MAX-CUT problem [4] or to the problem of finding the ground state of a frustrated spin system in statistical physics [20]. Like in [4] and [8], we formulate it as a problem of identifying the minimal number of edges whose removal leave the graph consistent. In the following, we briefly review the approach of [4] and [8] and then formulate a series of algorithms for the computation of an upper and lower bound of δ.

In [4] the authors use a polynomial-time approximation algorithm derived from the work of Goemans and Williamson [6]; this algorithm, based on semi-definite programming, is characterized by a performance guarantee $\alpha = 0.87856$, which means that the solutions delivered by the program are always at least α times the optimal value (*i.e.* the maximum possible size of the balanced edge subset). The authors were in this way able to obtain upper and lower bounds for δ. In particular, in a graph $G = (V, E)$, let B be the subset of edges of the balanced subgraph of G found by the algorithm. Then the optimal δ is limited to the interval $[\delta^-, \delta^+]$, where $\delta^- = |E| - |B|/\alpha$ and $\delta^+ = |E| - |B|$. Unfortunately, the computational speed of the implementation provided by [4] is quite low, hence the software is unable to calculate a truly large number of randomizations within a reasonable time. Furthermore, the performance guarantee α results in a lower bound which is not sufficiently tight, and worsens for near-monotone networks (when the upper bound gets closer to the "true" one, i.e., $|B|$ is large).

In [8], the problem is tackled from a different perspective. Their "balanced subgraph" approach uses a fixed parameter system and a data reduction scheme. This last works by substituting portions of the starting graph G with simpler parts called gadgets, exhibiting the same behavior of the original counterpart (in terms of δ). In this way, their algorithm is claimed to be capable of drastically reducing the size of the starting graph, eventually obtaining a much simpler one which is then solved by an iterative compression procedure. The final result of [8] is an upper bound for δ that is slightly better than the one of [4] and that is calculated in a computational time which depends on both the size and the topology of the network. With the software implementation provided in [8] we were unable to obtain an estimate of δ in large networks, see Section 5. In addition, for other networks our algorithms (described below)

improve considerably the bounds found in [8], which, of course, questions the exactness claimed in the paper [8], or at least the correctness of the implementation provided with it.

4.1 Algorithms for the upper bound

The starting point of our approach is the idea, enunciated in Proposition 2, that switching sign to the edges of a cut set leaves δ invariant, and hence that such switching equivalences can be used to look for a representative $G(\sigma)$ in the equivalence class of δ which is easier to treat, namely a $G(\sigma)$ having a minimal number of $-$ signs. The following (heuristic) algorithm acts on cut sets partitioning the graph into subgraphs $V_1 = \{v_i\}$ and $V_2 = \{v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n\}$. For a vertex $v_i \in V$, let $d^+(v_i)$ be the number of positive edges adjacent to v_i and $d^-(v_i)$ the number of negative edges adjacent to v_i.

Algorithm 1 *Consider a vertex v_i, $i = 1, \ldots, n$, in a signed graph G.*

- *If $d^-(v_i) > d^+(v_i)$, change sign to all the edges adjacent to v_i, so now $d^+(v_i) > d^-(v_i)$.*
- *Repeat for all i until $d^-(v_i) \leq d^+(v_i)$, $\forall\, i = 1, \ldots, n$ (or until there is no improvement in the total number of positive edges in $G(\sigma)$).*

The set of all edges adjacent to a given vertex v_i is a cut set partitioning G into the isolated vertex v_i and $n-1$ other vertices, hence Proposition 2 applies. Consequently, all switches of signs in cut sets inducing graph-partitioning of the type $\{1 \text{ vertex}\}/\{n-1 \text{ vertices}\}$ do not alter the sign of any cycle of the graph. Of course any $\{k \text{ vertices}\}/\{n-k \text{ vertices}\}$ cut set could be used instead. In particular, if the signature σ_i corresponds to the partition $\{v_i\}/\{v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n\}$, the iterative procedure of Algorithm 1 corresponds to a matrix multiplication of P_{σ_i}: $P_\sigma = P_{\sigma_{i_1}} P_{\sigma_{i_2}} \ldots P_{\sigma_{i_k}}$ where the order is irrelevant as P_{σ_i} commutes with P_{σ_j}. Hence the new signed graph $G(\sigma)$ of adjacency matrix $P_\sigma J P_\sigma$ has the same consistency properties as the original G of adjacency matrix J, but for it the violations of consistency are easier to identify. In fact, its edges with a minus sign are typically much less than in the original graph. Therefore constructing a suitable fundamental cycle basis with minimal number of negative cycles is an easier task. For example, whenever possible we can choose a spanning tree which does not pass through edges with negative sign. In the "best" cases, edges with negative sign will be chords and hence associated with only one of the fundamental cycles. Provided we associate high weights to the edges having a negative sign after the application of Algorithm 1, any algorithm for a minimal spanning tree [5] will select a spanning tree with minimal number of minus signs. From Corollary 2, the cycle subspace associated with the set of fundamental cycles having positive sign corresponds to the monotone subsystem of the original system.

This greedy algorithm, inspired from the idea of "gauge transformation" of Ising spin systems [21] and from the literature on "switching equivalence" of signed graphs [23], shows an extremely fast computational speed, and the solutions delivered are comparable to those obtained by [8].

In order to improve slightly on the termination point of Algorithm 1 (in particular in the case $d^+(v_i) = d^-(v_i)$ for some i), the following can be used.

Algorithm 2 *Consider a vertex v_i, $i = 1, \ldots, n$, in a graph $G = (V, E)$. If $d^-(v_i) \leq d^+(v_i)$, $\forall\, i = 1, \ldots, n$ then:*

- *Create the set $W = \{w \in V \mid d^-(w) = d^+(w)\}$.*
- *Consider the subset $W' \subset W$ such that $\forall w_i, w_j \in W' \Rightarrow (w_i, w_j) \notin E$ (i.e., W' contains all the vertices that have $d^+ = d^-$ and that are not directly linked by an edge).*

- *Find the vertex $v \in V \setminus W'$ which maximizes the function $f(v) = 2 * |W'_v| + d^-(v) - d^+(v)$, where $W'_v = \{w' \in W' | (v, w') \in E$ and (v, w') is a positive edge$\}$.*
- *If the maximized $f(v)$ is > 0, then switch all the vertices $w' \in W'_v$; now, since $f(v)$ is positive it must be $d^-(v) > d^+(v)$ and the graph G can again be processed by Algorithm 1.*

In other terms, such an algorithm is able to manipulate a graph $G(\sigma_1)$ in which there are no vertices suitable for a convenient sign inversion in order to obtain a graph $G(\sigma_2)$ in which there is at least one vertex suitable for sign inversion. It does so by excluding from the count of d^+ and d^- of a vertex the edges having the other vertex in the set W' of vertices whose edge signs can be freely switched without increasing the total number of negative edges.

The combined use of Algorithms 1-2, though fast, yield a solution which is only locally optimal: this is a well-recognized limit of greedy-like algorithms [2], especially in classes of global optimization problems characterized by a large number of different local minima like the *max-cut* or *balanced-subgraph* problems, to which finding δ is equivalent. If a random perturbation (i.e a sign switch through a random cut set) is applied to $G(\sigma)$, its signature σ is moved away from the local minimum, and re-running Algorithms 1-2 typically results in a new minimum, with different $d^-_{G(\sigma)}$. The following rule of thumb concerns the size of the randomization step: the larger the applied cut set is, the greater is the distance between the signatures of the local minimum and the new initial condition. Thus, a possible algorithm iterating Algorithms 1-2 operates as follows:

Algorithm 3 *Consider a graph $G = (V, E)$ with signature σ.*

- *Process G with Algorithms 1-2 and save the resulting final signature.*
- *Apply to G a random sign change on a randomly chosen small cut set, then process again G with Algorithms 1-2.*
- *Repeat last step for τ times: if a better minimum cannot be found within the τ iterations, increase the size of random cut set. When a better minimum is found, restore the starting size of random cut set.*

Starting from the current value of the minimum, Algorithm 3 gradually moves away from it, increasing the size of the cut set through which switching the signs. With such a method, the space of the solutions is gradually explored while seeking for a global minimum. As the key concept inspiring Algorithm 3 is a stochastic-like exploration of the space of the solutions, it belongs to the category of randomized global optimization heuristic algorithms.

4.2 An algorithm for the lower bound

After the application of Algorithm 3 to G for enough iterations, the spin assignment σ of G will have a reduced amount of negative edges, although it will be in general $d^-_{G(\sigma)} \geqslant \delta$. It is unlikely for biological regulatory networks to be exactly monotone, if not because they contain particular negative subnetworks (like the incoherent feedforward loops of [11]) which cannot be eliminated, implying that $\delta > 0$.

Consider $G(\sigma)$, with the minimal number of $-$ signs obtained after the application of Algorithm 1-3. In order to bound δ from below, the idea is to iteratively assign to each remaining negative edge a cycle made only of positive edges so as to obtain a negative cycle edge-disjoint from the previously assigned negative cycles. As long as we can associate to each negative edge one such cycle, and all these cycles have no edge in common, then we are guaranteed that no cut set can improve $d^-_{G(\sigma)}$.

Proposition 4 *In a collection of edge-disjoint cycles presenting each one and only one negative edge, no cycle can be rendered positive by sign changes through any cut set.*

Proof. For a single cycle, the Proposition follows directly from Theorem 2. As long as the cycles are edge-disjoint, also its generalization to a collection of cycles is straightforward. ∎

The algorithm (also heuristic) for the lower bound aims at assigning as many such edge-disjoint negative cycles as possible to the negative edges of $G(\sigma)$.

Algorithm 4 *Consider a graph $G = (V, E)$ and denote E^- the subset of E containing only negative edges.*

- *Create the subgraph $G' \subset G$ such that $G' = (V, E \setminus E^-)$.*
- *Select an edge $(v, w) \in E^-$ and find the shortest possible path in G' connecting the vertices v, w.*
- *Remove from the graph G' all the edges used in the path.*
- *Repeat until all the $|E^-|$ negative edges are connected with a path in G'. If, for some $i = 1, \ldots, |E^-|$, there does not exist a path in G' connecting (v_i, w_i), start removing the $i - 1$ previous assigned paths in G', until (v_i, w_i) can be connected.*
- *Stop if a program loop is detected.*

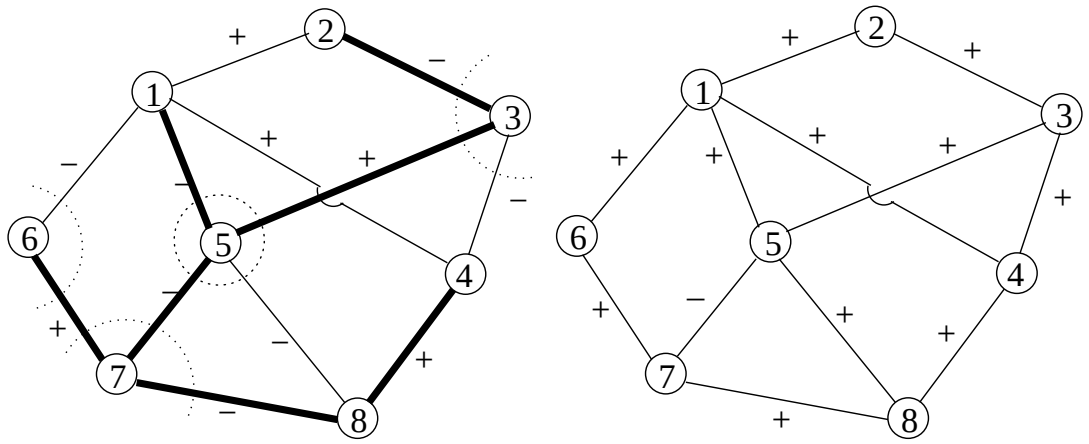
In the literature, the problems of constructing edge-disjoint cycles are called *cycle packing* problems, and are known to be NP-hard [3]. Of course, the success rate of Algorithm 4 is greatly improved by the pre-application of Algorithms 1-3 which (typically) drastically reduce the number of residual negative signs in G. Algorithm 4 outperforms simpler methods such as the recursive deletion of the shortest negative cycles (in the original graph), see Supplementary Table S1.

The combined application of Algorithms 1-3 and Algorithm 4 usually leaves a gap between lower and upper bound for δ: $\delta_{\text{low}} \leqslant \delta \leqslant \delta_{\text{up}}$. However, as we will see in the next Section, this gap is much tighter than the one proposed in [4]. At the same time, the overall computational time is greatly reduced, enabling us to treat also networks which are considerably larger than those discussed in [4, 8].

5 Examples

The 8-node network shown in Fig. 1 is useful in order to explain the methodology introduced in the paper. By simply inspecting the graph of Fig. 1 (a), it is not easy to draw a conclusion on how close to monotone the network is. Neither the number of $-$ signs on the edges, nor the number of fundamental cycles having negative sign in a randomly chosen spanning tree is indicative of the minimal number of signs required to achieve monotonicity. For example, with the choice of spanning tree of Fig. 1 (a), 4 out of 5 fundamental cycles of the basis are negative. However, when we apply the procedure described in Algorithm 1, we are left with a single negative fundamental cycle. As only one edge maintains the negative sign, finding the single inconsistency is obviously straightforward.

Four large scale biological networks were tested with Algorithms 1-4 and the results were compared to those of [4] and [8]. All the networks have been previously rendered symmetric by removing when needed symmetric-incompatible or sign-ambiguous edges pairs, as done in previous studies, see Table 1 for details.



(a) Original graph G.

(b) The graph $\tilde{G}$, after the application of Algorithm 1.

Figure 1: Example of application of Algorithm 1. The network represented has 8 nodes and 12 edges with signs given in (a). Choosing a spanning tree like in (a) leads to a fundamental cycle basis in which 4 out of 5 cycles have negative sign. Applying a sign change along the 4 cut sets shown as dotted segments in (a) (i.e., $\sigma = \{1, 1, -1, 1, -1, -1, -1, 1\}$) yields a graph $G(\sigma)$ with only a single $-$ sign, see (b). Hence only one element of the basis needs to have negative sign in the "optimal" choice of spanning tree. In this case the network has distance "1 edge" from monotonicity.

- *EGFR* Epidermal growth factor receptor pathway is a network consisting of 330 nodes and 852 edges. It was created by [13].
- *Macrophage* network is the molecular interaction map of a macrophage obtained by [12]. It has of 697 nodes and 1582 edges.
- *Yeast* network: gene regulatory network of *S.cerevisiae* originally developed in [11]. It contains 690 nodes and 1082 edges (representing transcription factor - binding site interactions).
- *E.coli* network is the gene regulatory network of the *E.coli*. Our version, downloaded from *RegulonDB* database (http://regulondb.ccg.unam.mx), version 6.3, for *E.coli* [15] is an updated and expanded version of the one originally developed in [11, 16] and used in [10]. It has 1475 nodes and 3320 edges. The ~ 200 arcs labeled with both signs (dual actions) are disregarded.

Table 1: Networks used in this study and their original signed edges

	n	m	incompat./symm. edges pairs	pos./neg. edges
EGFR	330	852	4/65	515/264
macrophage	697	1582	1/155	947/478
yeast	690	1082	1/0	860/220
E.coli	1476	3228	8/10	1879/1336

The first three of these networks are considered also in [8] (the first is also in [4] and the third also in [10]). As for the *E.coli* transcriptional network, our version is almost double in size with respect to the other three (and with respect also to the version considered in [10]).

As a way of understanding how Algorithm 1-4 work, in Fig. 2 the local minima for δ_{up} on 10^4 iterations are shown for the *EGFR* and *E.coli* networks. Upon reaching a local minimum of Algorithm 1-2, Algorithm 3 applies a random sign change on a randomly chosen cut set. The size of the cut set is progressively increased as long as the δ_{up} reached by Algorithm 1-2 remains worse than the best value of δ_{up} achieved so far. Fig. 2 shows that when the size of the cut set gets large, the local minimum achieved suddenly drops. All the times this happens, the new value of δ_{up} is the same in each network (in the Figure: 193 for the *EGFR* and 371 for the *E.coli*), suggesting that the local optimum might be a global optimum.

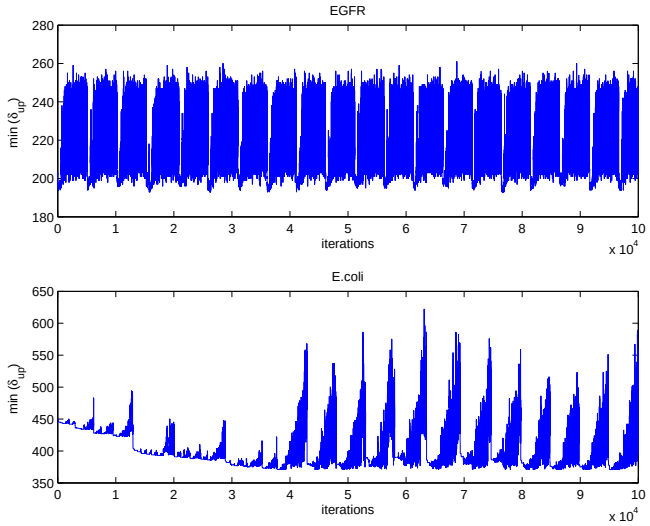


Figure 2: Upper bound δ_{up} achieved in 10^4 iterations of Algorithms 1-3 on the *EGFR* (top) and on the *E.coli* networks (bottom). Whenever a local minimum of Algorithms 1-2 is reached, a random sign switch on a randomly chosen cut set is applied. The size of this cut set is progressively increased as long as the new local minimum remains worse than the best one achieved so far. When the local minimum is improved, the size of the cut set is reset in order to look for improvements in a neighborhood. In both plots the same local minimum is always achieved (in *E.coli* after a transient) hence suggesting that the δ_{up} achieved might be the globally optimal δ.

Looking more in detail at the most complex of our networks (*E.coli*), its graph is composed of a large connected component (of 1376 genes) and of other 22 small connected components which have only (few) positive cycles and need no further analysis. The large connected component has 3150 edges (1848 + and 1362 −) and many millions of cycles. A fundamental cycle basis has 1775 cycles. Computing a minimal spanning tree with a basic Prim algorithm [14], we obtain that 970 cycles have positive sign and 805 have negative sign. Already with the application of a single run of Algorithm 1, we are left with 1317 cycles having positive sign and only 458 having negative sign. This is a local minimum in a landscape which is known to be very rough. The entire landscape can be explored changing randomly the initial σ in $P_\sigma J P_\sigma$ and applying Algorithms 1-2 to the resulting signed graph (equivalent to G in terms of distance to monotonicity). The upper bound found for the distance to monotonicity in the *E.coli* network in correspondence of $4.7 \cdot 10^6$ iterations is 371 (corresponding to 371 negative edges left in the graph). As a matter of fact, in the $4 \cdot 10^4$ times the algorithm converged to 371, the bulk of the edges have a concordant sign, see Fig. 3. In particular, a spanning tree completely formed by positive edges always exists, meaning that the 371 negative signs can always be relegated

to chords, whose removal leave the system monotone and the largest connected component still connected. Furthermore, on the subgraph with inconsistent edges removed, we can observe that the vast majority (1343 out of 1404) of the positive fundamental cycles are also edge connected i.e., they share pairwise at least an edge. Therefore, from Theorem 3, the network has a very large cycle subspace in which all the cycles formed by ring sums of the basis elements are guaranteed to be positive, meaning that the subspace is monotone. Hence we can conclude that the transcriptional network of *E.coli* behaves *de facto* as a monotone system.

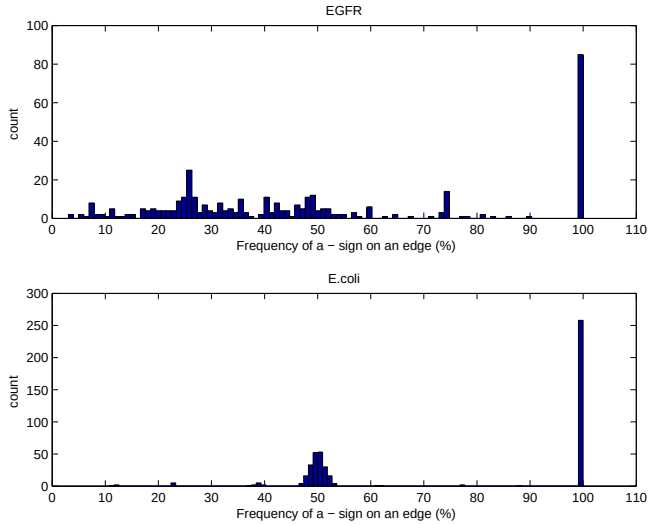


Figure 3: Frequency of assignment of the residual negative signs to the edges in the signed graph $G(\sigma)$ having a minimal number of $-$ signs for the *EGFR* (top) and *E.coli* network (bottom). For the *EGFR* network, only 85 of the 193 edges are unanimously assigned to the same edges (100% means unanimous assignment to an edge of a $-$ sign in all trials) while the other 108 (56%) are assigned with various probabilities to 377 edges. For *E.coli*, (in $4 \cdot 10^4$ successful trials in $4.7 \cdot 10^6$ attempts), most (258 out of 371) negative signs are always assigned to the same edges, as can be seen by the higher peak on the left of the histogram. Only 30% of the 371 negative signs are located on different edges in our trials. However, these 113 negative edges are distributed only on a total of 231 edges.

Even a large network such as *E.coli* is treated efficiently by our algorithms in a fairly limited amount of time (a few minutes on an ordinary PC for $\sim 10^6$ iterations of Algorithms 1-3). Analogous performances are obtained with the other networks under study: the algorithm is always extremely fast and efficient, regardless of the size and the topology of the networks. In each of these networks the best estimate for δ_{up} is reached in almost every cycle of iterations, thus indicating a high reliability of the algorithms.

In the second step of the analysis, the networks with reduced negative edges are tested with Algorithm 4 for the lower bound δ_{low}. The program receives as input the list of negative edges remaining after the simplification, and tries to verify how many of them can be labeled as unremovable by assigning to each of them a disjoint negative cycle (in which all other edges are positive). For the *EGFR* network 186 out of 193 edges can be confirmed as unremovable, meaning that for the true δ we have $186 \leq \delta \leq 193$. For the *macrophage* network 302 out of 332 edges are confirmed as unremovable, i.e., $302 \leq \delta \leq 332$. For the remaining networks the results are notably better: 365 out of 371 edges can be confirmed for the *E.coli* network, while all the 41 edges are confirmed for the *yeast* regulatory network, meaning that in this case 41 is the

true value of the consistency deficit. The results obtained on the four networks are summarized

Table 2: Estimation of the consistency deficit δ from the literature and from our Algorithms. The "*" correspond to a case in which the algorithm of [8] did not converge.

	Ref. [4]	Ref. [8]	Algorithm 1-4
EGFR	$124 \leqslant \delta \leqslant 219$	$\delta \leqslant 210$	$186 \leqslant \delta \leqslant 193$
macrophage	$204 \leqslant \delta \leqslant 383$	$\delta \leqslant 374$	$302 \leqslant \delta \leqslant 332$
yeast	$0 \leqslant \delta \leqslant 43$	$\delta \leqslant 41$	$41 \leqslant \delta \leqslant 41$
E.coli	$0 \leqslant \delta \leqslant 385$	*	$365 \leqslant \delta \leqslant 371$

and compared with the literature in Table 2. In all cases, we are able to improve the existing results for both lower and upper bounds. Explicit partial order vectors σ achieving the δ_{up} values for the 4 networks are provided in the Supplementary Files. For the lower bounds, in the Supplementary Table S1 the performances of Algorithm 4 are also compared with those of a simpler algorithm based on progressively deleting the shortest negative cycles on the original graph. In the last column of Table 4, the ratio between δ_{up} and δ_{low} is shown. For all 4 networks it is higher than 90%, meaning that at least 90% of the remaining negative edges are confirmed to belong to edge-distinct negative cycles, i.e., to give rise to non-eliminable negative cycles. The gap between lower and upper bound appears to be a function of the density and of the consistency of the network rather than of its size. The best results are obtained for those networks in which the consistency deficit is lower, as reported in Table 4. This happens because when δ is low, the ratio between positive/negative edges is high, i.e. there are more positive edges available to pack the negative edges in disjoint cycles.

Table 3: The theoretical worst-case upper bound of Proposition 1 for the consistency deficit. The tighter value is shown in boldface.

	μ	η
EGFR	452	**376**
macrophage	751	**704**
yeast	**401**	528
E.coli	1775	**1581**

In Table 3, we compare the two theoretical worst-case upper bounds for the consistency deficit mentioned in Proposition 1. Only for the *yeast* transcriptional network $\mu < \eta$, meaning that the ratio m/n is particularly low for it. The bound $\min(\mu, \eta)$ allows to evaluate the consistency deficit of the network relative to its values of n and m and hence to its number of "independent" cycles. From Table 4, we have that for the transcriptional networks of *yeast* and *E.coli* only 10.22% and 23.47% of the possible independent cycles are negative, while for the *macrophage* and *EGFR* networks the percentages grow to 47.16% and 51.13%. In other words, for these two last networks the independent inconsistent cycles are roughly half the maximum possible number, meaning that these networks cannot be classified as "near-monotone". Notice that when comparing δ_{up} with the total number of edges (first column of Table 4) the relative percentages of negative cycles are much below 50% also for these last two networks. However, this is not a significant statistics from the point of view of establishing monotonicity.

To further compare the differences between transcriptional and signaling networks, we performed an analysis of the ratio $\delta_{\text{up}}/\min(\mu, \eta)$ while varying the number of initial negative edges

Table 4: Statistics for δ_{up}: percentage of negative cycles with respect to m and with respect to the upper bound of Table 3. Last column: number of negative cycles confirmed by the lower bound of Algorithm 4.

	δ_{up}/m (%)	$\delta_{\text{up}}/\min(\mu,\eta)$ (%)	$\delta_{\text{low}}/\delta_{\text{up}}$ (%)
EGFR	24.71	51.13	96.37
macrophage	22.94	47.16	90.96
yeast	3.76	10.22	100
E.coli	11.17	23.47	98.38

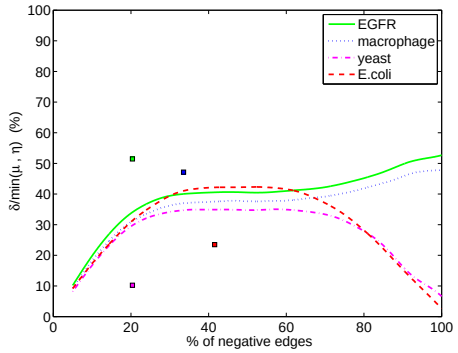


Figure 4: Value of the ratio δ/δ_{max} with varying number of negative edges randomly assigned to each of the 4 networks (mean over 1000 random instances for each of 20 different percentages of negative edges between 5% and 100%). All plots begin growing linearly for low concentration of negative signs, then reach a plateau. For the transcriptional networks (*yeast* and *E.coli*) the graph is essentially symmetric around 50% of negative edges. For signaling networks, instead, the second half of the plot is fairly different, with δ increasing. This difference is largely due to the cycles of length 3 introduced in the construction of these networks. The 4 boxes represent the true values of δ/δ_{max} (see Table 4).

randomly assigned to each of the four networks. The results are shown in Fig. 4. Clearly, the *macrophage* and the *EGFR* networks exhibit a substantially different behavior with respect to the transcriptional ones in the right tail of the graphs, where the number of initial negative edges approaches the 100%. This different behavior, with the (relative) consistency deficit growing for the two signaling networks and decreasing for the transcriptional ones, is necessarily arising from topological differences, since the negative edges are assigned randomly in the same manner for all the networks. Investigating more in detail how the signaling networks are constructed [4], we can observe that they are assembled largely from the interconnection of reactions of the type:

$$X_1 + \ldots + X_p \rightleftharpoons Z_1 + \ldots + Z_q \tag{3}$$

where X_i and Z_i are chemical species (nodes of our network). The simplest possible such reaction is bimolecular

$$X_1 + X_2 \rightleftharpoons Z_1 \tag{4}$$

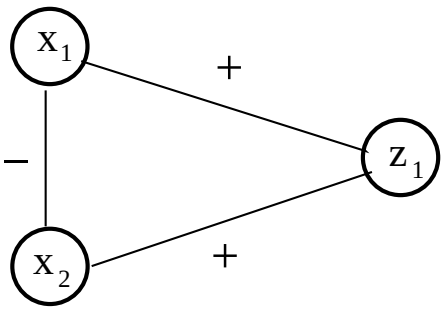


Figure 5: Signed graph for the ODE (5).

and its associated kinetics is

$$\begin{aligned} \frac{dx_1}{dt} &= -f_1(x_1, x_2) + f_2(z_1) \\ \frac{dx_2}{dt} &= -f_1(x_1, x_2) + f_2(z_1) \\ \frac{dz_1}{dt} &= f_1(x_1, x_2) - f_2(z_1) \end{aligned} \tag{5}$$

for some functions $f_1(\cdot)$, $f_2(\cdot)$. According to the procedure described in Section 2.2, we derive from these kinetics the following symmetric adjacency matrix:

$$J = \begin{bmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

i.e., the graph of Fig. 5, which corresponds to a simple negative cycle, implying that the system (5) is not monotone. For this particular case, the lack of order can be easily understood from the dynamics of the system (4)-(5) with respect to an equilibrium state. When more product Z_1 is added, both the reagents X_1 and X_2 increase, owing to the backward dissociation reaction. However, when more of one of the reagents, e.g. X_1, is added to the equilibrium state, the concentration of the other reagent X_2 decreases because the excess of X_1 depletes it. Therefore, an order relationship in the sense of Sec. 2.2 for the reagents X_1 and X_2 does not exist. Thus in this simple prototype system the lack of monotonicity is due to the process of compound formation/destruction from multiple reagents. Since the *macrophage* and the *EGFR* signaling networks arise from the interconnection of reactions such as (3), this causes both of them to have a higher consistency deficit δ with respect to the transcriptional networks. In addition, referring to Fig. 4, it is easy to explain the behavior of signaling networks when the amount of negative edges reaches the 100% in terms of abundance of odd length (usually length 3) cycles.

Since the original networks are essentially directed (especially the transcriptional networks), only a small fraction of the undirected fundamental cycles just mentioned correspond to directed cycles. However, the spin assignment interpretation is not restricted to complete directed cycles, but includes also e.g. 2 converging directed half cycles (e.g. feedforward loops) and also more complex concatenations of directed paths [20, 16, 1].

6 Conclusions and outlook

Testing how far or how close a biological network (represented as a signed graph) is from being monotone is a NP-hard problem, equivalent to well-known problems such as the MAX-CUT.

For it, in this paper, we proposed efficient heuristic algorithms, able to treat also large networks in a limited computational time. The output of the algorithms is an interval inside which the optimal value (i.e., the consistency deficit) must lie. For the four biological networks we analyze in this paper, we obtain that the two gene regulatory networks are indeed near-monotone i.e., close to monotone, while the two signaling networks have a number of inconsistencies which is roughly half the worst-case theoretical upper bound known for these networks. Our suggestion is that the higher consistency deficit δ characterizing signaling networks is due to the dynamics of compound formation/destruction inherent to this kind of networks. However, further work is required to asses more properly the distance to monotonicity of biological networks. In fact, our considerations are primarily based on a comparison with the worst-case upper bound, which in principle may not be attainable for certain topologies. Validation on other networks is also necessary before any claim can be reasonably established.

7 Acknowledgments

The work was sponsored in part by a grant from Illy Caffé, Trieste. We would like to thank E. Sontag and A. Canton for discussions on the topic of the work.

References

[1] U. Alon. Network motifs: theory and experimental approaches. *Nat Rev Genet*, 8(6):450–461, 2007.

[2] A. Bang-Jensen, G.Gutin. When the greedy algorithm fails. *Discrete optimization*, 1:121–127, 2004.

[3] A. Caprara, A. Panconesi, and R. Rizzi. Packing cycles in undirected graphs. *J. Algorithms*, 48(1):239–256, 2003.

[4] B. DasGupta, G. A. Enciso, E. Sontag, and Y. Zhang. Algorithmic and complexity results for decompositions of biological networks into monotone subsystems. *Biosystems*, 90(1):161–178, 2007.

[5] N. Deo. *Graph theory with applications to engineering and computer science*. Prentice-Hall, Englewood Cliffs, N. J., 1974.

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

[7] M. Hirsch and H. L. Smith. Monotone dynamical systems. In *Handbook of differential equations, ordinary differential equations*. Elsevier, Amsterdam, 2005.

[8] F. Hüffner, N. Betzler, and R. Niedermeier. Separator-based data reduction for signed graph balancing. *Journal of Combinatorial Optimization*, page (to appear), 2009.

[9] G. Kirchhoff. Über die Auflösung der Gleichungen, auf welche man bei der untersuchung der linearen Verteilung galvanischer ströme geführt wird. *Ann. Phys. Chem.*, 72:497–508, 1847.

[10] A. Ma'ayan, R. Iyengar, and E. Sontag. Proximity of intracellular regulatory networks to monotone. *IET Systems Biology*, 2:103–112, 2008.

[11] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[12] K. Oda, T. Kimura, Y. Matsuoka, A. Funahashi, M. Muramatsu, and H. Kitano. Molecular interaction map of a macrophage. *AfCS reports*, 2(14), 2004.

[13] K. Oda, Y. Matsuoka, A. Funahashi, and H. Kitano. A comprehensive pathway map of epidermal growth factor receptor signaling. *Mol Syst Biol*, 1:2005, 2005.

[14] R. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.

[15] H. Salgado, S. Gama-Castro, M. Peralta-Gil, E. Diaz-Peredo, F. Sanchez-Solano, A. Santos-Zavaleta, I. Martinez-Flores, V. Jimenez-Jacinto, C. Bonavides-Martinez, J. Segura-Salazar, A. Martinez-Antonio, and J. Collado-Vides. RegulonDB (version 5.0): *Escherichia coli* K-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Res.*, 34(Database issue):D394–D397, 2006.

[16] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of Escherichia coli. *Nat. Genet.*, 31(1):64–68, 2002.

[17] H. L. Smith. Systems of ordinary differential equations which generate an order preserving flow. A survey of results. *SIAM Review*, 30(1):87–113, 1988.

[18] P. Solé and T. Zaslavsky. A coding approach to signed graphs. *SIAM J. Discrete Math.*, 7(4):544–553, 1994.

[19] E. Sontag, A. Veliz-Cuba, R. Laubenbacher, and A. S. Jarrah. The effect of negative feedback loops on the dynamics of boolean networks. *Biophys J*, 95(2):518–526, 2008.

[20] E. D. Sontag. Monotone and near-monotone biochemical networks. *Systems and Synthetic Biology*, 1:59–87, 2007.

[21] G. Toulouse. Theory of the frustration effect in spin glasses : I. *Communications on Physics*, 2:115, 1977.

[22] T. Zaslavsky. Signed graphs. *Discrete Appl. Math.*, 4(1):47–74, 1982.

[23] T. Zaslavsky. Bibliography of signed and gain graphs. *Electr. J. Combinatorics*, DS8, 1998.