Bull Math Biol (2013) 75:939–966
DOI 10.1007/s11538-012-9794-1

Society for Mathematical Biology

ORIGINAL ARTICLE

Deconstruction and Dynamical Robustness of Regulatory Networks: Application to the Yeast Cell Cycle Networks

Eric Goles · Marco Montalva · Gonzalo A. Ruz

Received: 23 March 2012 / Accepted: 1 November 2012 / Published online: 28 November 2012

Abstract Analyzing all the deterministic dynamics of a Boolean regulatory network is a difficult problem since it grows exponentially with the number of nodes. In this paper, we present mathematical and computational tools for analyzing the complete deterministic dynamics of Boolean regulatory networks. For this, the notion of alliance is introduced, which is a subconfiguration of states that remains fixed regardless of the values of the other nodes. Also, equivalent classes are considered, which are sets of updating schedules which have the same dynamics. Using these techniques, we analyze two yeast cell cycle models. Results show the effectiveness of the proposed tools for analyzing update robustness as well as the discovery of new information related to the attractors of the yeast cell cycle models considering all the possible deterministic dynamics, which previously have only been studied considering the parallel updating scheme.

Keywords Boolean networks · Attractors · Update robustness · Alliances · Dynamics

1 Introduction

Boolean Regulatory Networks (*BRN*) were introduced by Kauffman (1969). Roughly, they consist in a finite directed graph, where nodes represent genes in discrete states

E. Goles · M. Montalva · G.A. Ruz (✉)
Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Av. Diagonal las Torres 2640, Peñalolén, Santiago, Chile
e-mail: gonzalo.ruz@uai.cl

E. Goles
e-mail: eric.chacc@uai.cl

M. Montalva
e-mail: marco.montalva@uai.cl

(say 0 or 1) governed by Boolean functions depending on the input edges, which represents the gene's interactions. Suppose the network has n genes, so we define the update of the genes by a function $s : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that $s(a) \leq s(b)$ if and only if gene a is updated before or at the same time as b. A deterministic dynamics is associated to the network by the application of a specific deterministic update schedule. The parallel dynamics consists in updating the whole set of nodes synchronously, i.e., $s(i) = 1$ for every gene; the sequential dynamics updates the nodes one by one in a prescribed order, i.e., s is a permutation of the set $\{1, \ldots, n\}$. Between these two schedules, there are many others, depending of the order in which the nodes are updated. Actually, there are an exponential number of updates, as mentioned in Demongeot et al. (2008), the number T_n of deterministic update schedules associated to a digraph of n vertices is equal to the well-known number of ordered partitions of a set of size n, that is,

$$T_n = \sum_{k=0}^{n-1} \binom{n}{k} T_k$$

where $T_0 \equiv 1$.

Given a particular updating scheme, for any initial condition the network converges to the steady state conformed by limit cycles and/or fixed points. A *fixed point* is a Boolean vector such that it does not change under the application of the update schedule. A *limit cycle* of period $p > 1$ is a set of p vectors $x(0), x(1), \ldots, x(p-1)$ such that the $x(t + 1 (\mathrm{mod}\, p))$ is the successor of $x(t)$. Furthermore, it is direct that fixed points, if they exist, are invariant under updates. We call an *attractor* a fixed point or a limit cycle. Given an attractor A, the *basin of attraction* of A is the set of configurations not in A that reach a configuration of A after a finite number of time steps.

Within the context of gene regulatory networks, the nodes of a *BRN* represent the genes and the Boolean functions how they interact. States 1 and 0 correspond to expressed and nonexpressed genes, respectively, and attractors are associated to cell differentiation. Given that fixed points are invariant under the application of different updating schemes, fixed points are usually associated to different cell types. For example, the Boolean network model of the floral morphogenesis in *Arabidopsis thaliana* (Mendoza and Alvarez-Buylla 1998) has 13 attractors (7 limit cycles of length two each, and 6 fixed points), but only the six fixed points are associated with different cell types: (1) sepal, (2) petal, (3) carpel, (4) stamen, (5) inflorescence, (6) mutant (unobserved cell). Less frequent is the case where limit cycles are associated to cell types due to the strong dependence of the updating scheme, nevertheless, there are some examples. In Fauré et al. (2006), a logical model of the mammalian cell-cycle network is presented which contains (in the parallel updating mode) 2 attractors: 1 fixed point and 1 limit cycle of length seven. In this case, the fixed point represents a quiescent cell (G_0 phase), whereas the limit cycle represents the cell-cycle.

One of the first studies about dynamical robustness (i.e., when we change the update mode, the attractors, basins, etc. remains more or less the same) was conducted in Elena (2009), where four complexity classes are defined that are related to whether or not a *BRN* admits fixed points and/or cycles for every updating mode.

In the computer experiments, the author considers only threshold networks (each node has a Heaviside Boolean function). Since the number of updating modes is exponential, the exhaustive analysis of the *BRN* is very difficult, therefore, networks with only $n = 3$ are considered. Examples of robust and nonrobust networks in different applications such as brain and plant morphogenesis, bulbar cardio-respiratory regulation, glycolytic/oxidative metabolic coupling, and cell cycle are studied in Demongeot et al. (2009).

Another example is the threshold Boolean network constructed in Ruz and Goles (2012) using the input-output data from a logical model of the mammalian cell-cycle network presented in Fauré et al. (2006) (based on a differential model in Novak and Tayson 2004). The robustness of the network is explored with respect to update perturbations. The results showed that the model has different limit cycles when changing from parallel to a sequential update.

From a mathematical point of view, one of the first comparisons between different updating modes was done by Robert (1986), in particular, the parallel and sequential updates are studied. More related with this paper are the results obtained in Aracena et al. (2009). Given a BRN, equivalence classes of update schedules are defined such that inside a class, networks have the same dynamical behavior. One of the first results, which showed the difficulty in comparing different updating modes was obtained in Goles and Salinas (2008), where, for monotone networks, it is shown that it is not possible that the parallel and the sequential update share limit cycles. On the other hand, in Ruz and Goles (in press), through computer simulations using threshold Boolean networks with $n = 6$ it was found that most of the time, limit cycles are destroyed when the network changes its updating mode from parallel to sequential. Robust networks were only found for limit cycle length equal to two and specific network topologies. Recent theoretical results about dynamical robustness is presented in Goles and Noual (2012), where disjunctive networks are classified according to the robustness of their dynamics with respect to changes of their update schedules. Critical Kauffman networks (indegree $= 2$) under deterministic asynchronous update are investigated in Greil et al. (2007) by introducing node-based delays. Results show that delays typically increase the attractor lengths and reduce the attractor numbers as well as the emergence of new types of attractors. Also, the basins of attraction are larger and, therefore, the path to the attractor becomes more robust.

In Serra et al. (2010), a different approach is considered where the dynamics is described by sets of attractors rather than single ones. They use the notion of ergodic sets, which considers the attractors of the network and also transitions between them via state space perturbations (single node flip). This perturbation may result in the system moving toward another attractor. Given that the ergodic set is robust with respect to noise, the authors propose to associate sets of attractors to cell types rather than single attractors as proposed in the original model by S. Kauffman.

Throughout the paper, we will refer to dynamics and updating modes, but only in a deterministic way. It is important to point out that there are other non-deterministic updating schemes (Gershenson 2004) such as *asynchronous* where a node is randomly chosen and the network updated. Another example, is the nondeterminism approach used in Richard et al. (2012) to model biosurfactants production in *Pseudomonas fluorescens*.

Since the analysis in this paper, is related to the whole set of deterministic updates, which is of exponential cardinality, we have to develop strategies and results in order to diminish the amount of computation to analyze the proposed networks. In particular, it is not a trivial problem to know if a given network has only fixed points as attractors whatever updating mode is considered. In this context, we will analyze two networks related with the yeast cell-cycle (Davidich and Bornholdt 2008; Li et al. 2004). In both papers, the authors construct a Boolean (threshold) discrete model of the yeast cell-cycle updated in parallel that agrees with the biological constraints of the yeast cell-cycle. Furthermore, they studied the robustness of the models related with small changes in the network's interconnections (Li et al. 2004) as well as the robustness of the largest basin of attraction of the model, i.e., given a vector in such basin, to study the trajectory of a perturbation consisting in flipping the value of a single randomly chosen gene (Li et al. 2004). Although in this paper, we consider the two models mentioned previously, there are other Boolean models for these networks. For example, in Irons (2009), a new model for the budding yeast cell cycle is presented that is robust since there is only one attractor for the dynamics to follow. Given that there are no other alternative attractors, the order in which the nodes are updated, does not alter the fundamental behavior of the system. In Mangla et al. (2010), Boolean models for the budding and fission yeast cell cycle are constructed to study robustness to timing variations using model-checking for the property of speed independence (a system that has a desired property in a fully asynchronous model). It was found that these models were nearly, but not totally, speed-independent. Our approach consists in studying update robustness, but for every updating mode. To achieve this, we use some theoretical tools related with the characterization of equivalent updating modes (Montalva 2011) and the notion of *alliance,* which we will introduce below. These tools allow us to dramatically decrease the number of updates to analyze. Furthermore, we see that such networks are minimal in the sense that no edge can be deleted because every one of them represents a real interaction in the respective network. The notion of minimal network was used in Elena (2009) to study the smallest network (w.r.t. the number of edges) within the set of networks with the same dynamics. Roughly speaking, it is equivalent to consider interactions which one cannot eliminate.

Also, in order to apply these tools to our study of the two yeast cell-cycle networks, we prove a sufficient condition such that the network's dynamics converges, for every updating mode, to fixed points. From that result, we are able to study extensively, the proposed networks. For the first one, say *Yeast1* (Davidich and Bornholdt 2008), we characterize every updating mode, and in particular, whether or not limit cycles appear. Actually, we will give a partition of the updates related with the existence of limit cycles. Within the previous context, since exporeous limit cycle appear, *Yeast1* is not completely robust in the sense of update changes. For the second network, *Yeast2* (Li et al. 2004), we prove that for any update the attractors are only fixed points (no limit cycles). So, in some way (there are no limit cycles for any update schedule) *Yeast2* is more robust than *Yeast1*. Concerning the basin of attraction for the parallel update of the principal fixed point, we observe that independently of the update mode this basin of attraction remains the most important one.

1.1 Threshold Networks, Alliances and Fixed-Points Conditions

In threshold networks, each gene updates its sate by a Heaviside function:

$$x_i' = H\left(\sum_{j=1}^{n} w_{ij}x_j - \theta_i\right) = \begin{cases} 0 & \text{if } \sum_{j=1}^{n} w_{ij}x_j - \theta_i \leq 0, \\ 1 & \text{if } \sum_{j=1}^{n} w_{ij}x_j - \theta_i > 0 \end{cases} \tag{1}$$

where $W = (w_{ij})$ is the weight matrix with integral entries associated to the network and $\theta = (\theta_i)$ the threshold vector. The graph associated to W is $G = (V, E)$ where $V = \{1, \ldots, n\}$ is the set of nodes (genes) and $E = \{(i, j)/ \text{ iff } w_{ji} \neq 0\}$ being the set of edges (gene's interactions).

In the models (Davidich and Bornholdt 2008; Li et al. 2004) to be studied in the paper, there exist genes such that $\sum_{j\neq i} w_{ij}x_j - \theta_i = 0$ implies that the value of the gene remains, but in some specific cases (nodes marked with a loop over them), if $x_i(t) = 1$, then the new value changes, i.e., $x_i' = 0$. This situation can be described as in a previous definition of the Heaviside function by nonnegative diagonal values (if the gene does not change in the tie case) and a diagonal negative value when it changes from 1 to 0.

Since we have to study every update schedule, it is necessary to develop tools that allow us to diminish the amount of computational load. For that, we introduce below alliances and equivalent classes of update schedules.

We will say that a subconfiguration $(a_{i_1}, \ldots, a_{i_p}) \in \{0, 1\}^p$, $\{i_1, \ldots, i_p\} \subseteq V$, is an *alliance* if and only if

$$a_{i_k} = H\left(\sum_{j=1}^{p} w_{i_k i_j}a_{i_j} + \sum_{j\notin\{i_1,\ldots,i_p\}} w_{i_k j}x_j - \theta_{i_k}\right)$$

$$\text{for } k \in \{1, \ldots, p\} \text{ and any } x \in \{0, 1\}^{n-p}$$

Roughly speaking, an *alliance* is a subconfiguration of binary values, associated to a subset of genes that remain fixed for any binary value of the other genes. It is direct that alliances are invariant under update changes. Furthermore, any fixed point is trivially an *alliance*. Of course one may find networks without alliances, therefore, without fixed points (see Fig. 1(c) and (d) as example), and also with alliances but with no fixed points (see Fig. 1(a) and (b) as example). The interesting case is when there exists *alliances* with an strict subset of genes, which can be seen as a strong local fixed point that remains constant independently of the values of the others genes. So, to study the whole set of updates, it is sufficient to analyze the behavior of the genes that do not belong to the alliance. Also, one could expect that the existence of alliances tends to freeze the network's dynamics (convergence to fixed points). As we will see for *Yeast1*, it is not always the case: It admits limit cycles for several updating modes.

We say that two different updating modes are *equivalent* if and only if under its application the networks exhibit exactly the same dynamical behavior. This approach was first developed for sequential updates in Mortveit and Reidys (2004). Furthermore, for regulatory networks, the first characterization was done in Aracena et al. (2009), and studied in depth in Montalva (2011).

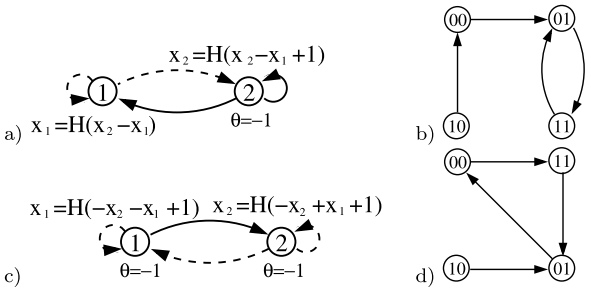

Fig. 1 In (**a**) and (**c**), the *dotted lines* represent negative interactions (weight $= -1$) while the *others* are positive interactions (weight $= 1$). (**a**) A BN of 2 nodes with threshold Boolean functions defined as x_1, x_2, and thresholds $\theta = 0$ and $\theta = -1$, respectively. (**b**) The dynamical behavior of (**a**) for the parallel update schedule. Note that $x_2 = 1$ is an alliance although the dynamic has no fixed point. (**c**) A BN as in (**a**) but with both thresholds equal to -1. (**d**) The dynamical behavior of (**c**) for the parallel update schedule. Note in this case that the network has no alliance

In this paper, we use a specific algorithm to enumerate the different equivalence classes (see Appendices A and B for examples and more details) for *Yeast1* which allows us, together with the *alliance* notion, to characterize its dynamics for the whole spectrum of deterministic update schedules. For *Yeast2*, the study is more easy because by considering an *alliance* as well as a fixed point sufficient condition, we establish that the only attractors are fixed points. The fixed point condition is the following.

Proposition 1 *Consider a Boolean network such that each gene is governed with a threshold function. Then*:

(i) *if the associated incidence graph, without considering the diagonal elements, is a directed acyclic graph* (DAG) *and* $w_{i,i} \geq 0$, *then for any updating mode the attractors are only fixed points.*
(ii) *if the associated incidence graph, without considering the diagonal elements, is a directed acyclic graph* (DAG) *and the thresholds are non negatives,* $\theta_i \geq 0$, *then for any updating mode the attractors are only fixed points.*

Proof Let us verify case (i). Clearly a *DAG* can be partitioned in $S_1, \ldots, S_p$ set of genes, such that genes in S_1 are sources and those in S_p are sinks. Furthermore,

$$\text{if } (a, b) \in E \text{ then } a = b \text{ or } (a \in S_r \text{ and } b \in S_q) \text{ with } r < q.$$

Let us study the set S_1. Consider the first step of the application of the update application. Suppose first a gene $k \in S_1$ such that $x_k = 0$ then

$$x_k(1) = H(0 - \theta_k) = \begin{cases} 0 & \text{if } \theta_k \geq 0, \\ 1 & \text{if } \theta_k < 0. \end{cases}$$

In this situation, if $\theta_k < 0$, the gene changes from 0 to 1, but at the next step: $x_k(2) = H(w_{k,k}x_k(1) - \theta_k) = H(w_{k,k} - \theta_k) \geq 0 = x_k(1)$ because $w_{k,k} \geq 0$. So, at the most,

in two steps the gene remains constant. Let us study now the case $x_k = 1$. Similar to the previous case,

$$x_k(1) = H(w_{k,k} - \theta_k) = \begin{cases} 0 & \text{if } \theta_k > w_{k,k}, \\ 1 & \text{if } \theta_k \leq w_{k,k}. \end{cases}$$

Hence, for $\theta_k \leq w_{k,k}$, the gene remains constant in the value 1. In the other case, say $\theta_k > w_{k,k} \geq 0$ then $x_k(1) = 0$, and $x_k(2) = H(w_{k,k}x_k(1) - \theta_k) = H(0 - \theta_k) = 0$ so it remains constant in the value 0. Hence, at the most, in two applications of the update the first level of the *DAG* will be fixed. By induction over the layers, we finish the proof. For (ii), the proof is similar. □

2 Fission Yeast Cell-Cycle Model (*Yeast1*)

Let us consider the Boolean network *Yeast1* studied in Davidich and Bornholdt (2008). It is composed by ten nodes as shown in Fig. 2. Using the same configuration as in Davidich and Bornholdt (2008), the green/solid edges represent positive weights (activations), the red/dashed edges represent negative weights (inhibitory). The yellow/solid loops represent self-degradation, which are modeled mathematically by a negative weight. The gene updates are computed by a variant of (1), as defined in Davidich and Bornholdt (2008):

$$x_i' = H\left(\sum_{j=1}^{n} w_{ij}x_j - \theta_i\right) = \begin{cases} 0 & \text{if } \sum_{j=1}^{n} w_{ij}x_j - \theta_i < 0, \\ 1 & \text{if } \sum_{j=1}^{n} w_{ij}x_j - \theta_i > 0, \\ x_i & \text{if } \sum_{j=1}^{n} w_{ij}x_j - \theta_i = 0. \end{cases} \tag{2}$$

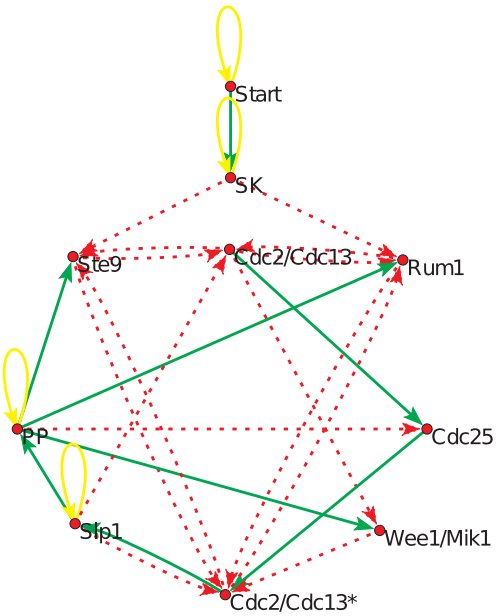

Fig. 2 The fission yeast cell-cycle threshold Boolean network. Using the same configuration as (Davidich and Bornholdt 2008), the *green/solid edges* represent positive weights (activations), the *red/dashed edges* represent negative weights (inhibitory). The *yellow/solid loops* represent self-degradation

$$W = \begin{array}{c} \\ Start \\ Sk \\ Cdc2/Cdc13 \\ Ste9 \\ Rum1 \\ Slp1 \\ Cdc2/Cdc13^* \\ Wee1/Mik1 \\ Cdc25 \\ PP \end{array} \begin{array}{c} \begin{array}{cccccccccc} Start & SK & Cdc2/Cdc13 & Ste9 & Rum1 & Slp1 & Cdc2/Cdc13^* & Wee1/Mik1 & Cdc25 & PP \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & -1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{pmatrix} \end{array} \quad \Theta = \begin{pmatrix} 0 \\ 0 \\ -0.5 \\ 0 \\ 0 \\ 0 \\ 0.5 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Fig. 3 Weight matrix and threshold vector for *Yeast1*

The weight matrix and the threshold vector used in (2) are shown in Fig. 3. To generate the same dynamics exhibited in Davidich and Bornholdt (2008), we have to consider thresholds -0.5 and 0.5 for genes $Cdc2/Cdc13$ and $Cdc2/Cdc13^*$, respectively.

Considering the four possible combinations of initial values for *Start* and *SK*, it is direct that $Start = 0$ is obtained at the most in 1 time step, and $SK = 0$ is obtained at the most in two time steps. Therefore, for $t > 2$, $Start = SK = 0$ always. So, for our analysis, we consider only the subnetwork composed by the remaining genes:

$$\left\{Cdc2/Cdc13, Ste9, Rum1, Slp1, Cdc2/Cdc13^*, Wee1/Mik1, Cdc25, PP\right\}$$

Also, we denote the current update vector as follows:

$$x(t) = \big(Cdc2/Cdc13(t), Ste9(t), Rum1(t), Slp1(t), \\ Cdc2/Cdc13^*(t), Wee1/Mik1(t), Cdc25(t), PP(t)\big)$$

In the analysis that follows, we will present two alliances of the network, named y_1 and y_2. To find them, we have observed common patterns in the fixed points exhibit in Davidich and Bornholdt (2008) since the network is small and keeping in mind the definition of alliance. We have seen that, although there are more than two alliances, these can be deduced from y_1 and y_2. Thus, in Proposition 2, we will prove that y_1 and y_2 are, in fact, alliances. Hence, in Corollary 1, we will prove that if one of such alliances is reached in the whole dynamic, then the successive configurations converge necessarily to fixed points. The latter will be especially useful to justify theoretically, each of the fixed points of this network.

In a second part of this analysis, we will study the influence of the update modes, in the alliance nodes, over the type of attractors that can exist in the network. To do that, we will begin by defining the respective partitions produced by considering all the possible ways to update the three nodes of the alliance, then we will prove through theoretical (alliance results) and numerical arguments (equivalence classes tools summarized in the appendix sections) that, in some partitions, some nodes remain fixed (Proposition 3). The latter will help us to deduce structural properties of the attractors that can exist in the dynamics (Corollary 2) as well as the partitions where they belong to.

Finally, with the numerical data obtained from the application of the equivalence classes tools on each partition, we will detail the average size of each attraction basin, results that will be discussed and shown in Table 2.

In what follows, we will study all the deterministic dynamics of *Yeast1* through the following propositions.

Proposition 2 *The subconfigurations with components*

$$y_1 = \big(Cdc2/Cdc13, Ste9, Cdc2/Cdc13^*\big) = (0,1,0),$$
$$y_2 = \big(Cdc2/Cdc13, Rum1, Cdc2/Cdc13^*\big) = (0,1,0)$$

are alliances *for Yeast1*.

Proof From the definition of alliances, one may prove that

$$y_1(t) = \big(Cdc2/Cdc13(t), Ste9(t), Cdc2/Cdc13^*(t)\big)$$

and

$$y_2(t) = \big(Cdc2/Cdc13(t), Rum1(t), Cdc2/Cdc13^*(t)\big)$$

remain fixed for the alliances $y_1(0) = y_2(0) = (0,1,0)$ and for any updating mode. Let s be an update schedule. We may suppose w.l.o.g that

$$s(Cdc2/Cdc13) \le s(Ste9) \le s(Rum1) \le s\big(Cdc2/Cdc13^*\big). \tag{3}$$

Thus, the respective local functions in these nodes are:

$$Cdc2/Cdc13(t+1) = H\big(-Ste9(t_2) - Rum1(t_3) - Slp1(t_4) + 0.5\big), \tag{4}$$
$$Ste9(t+1) = H\big(-Cdc2/Cdc13(t_1) - Cdc2/Cdc13^*(t_5) + PP(t_8)\big), \tag{5}$$
$$Rum1(t+1) = H\big(-Cdc2/Cdc13(t_1) - Cdc2/Cdc13^*(t_5) + PP(t_8)\big), \tag{6}$$
$$\begin{aligned} Cdc2/Cdc13^*(t+1) = H\big(&-Slp1(t_4) - Wee1/Mik1(t_6) - Rum1(t_3) \\ &- Ste9(t_2) + Cdc25(t_7) - 0.5\big) \end{aligned} \tag{7}$$

where t_j, $j \in \{1,\dots,8\}$ is equal to t or $t+1$ depending of the update schedule s used. Then, doing the respective calculations for $y_1(t+1)$, we have that

(3)$\Rightarrow t_2 = t$ in (4). Hence,

$$\left.\begin{aligned} Cdc2/Cdc13(t+1) &= H\big(-\underbrace{Ste9(t)}_{=1,\text{ by hyp.}} - Rum1(t_3) - Slp1(t_4) + 0.5\big), \\ &= H\big(-1 - Rum1(t_3) - Slp1(t_4) + 0.5\big) \\ &= H\big(\underbrace{-Rum1(t_3) - Slp1(t_4)}_{\le 0} - 0.5\big) \\ &\le H(-0.5) \\ &= 0. \end{aligned}\right\} \tag{8}$$

Thus,

$$Cdc2/Cdc13(t) \overbrace{=}^{\text{hyp.}} 0 \overbrace{=}^{(8)} Cdc2/Cdc13(t+1) \ \wedge\ (3) \Rightarrow t_5 = t \text{ in } (5) \tag{9}$$

So,

$$\left.\begin{aligned} Ste9(t+1) &= H\big(-\underbrace{Cdc2/Cdc13(t_1)}_{=0,\text{ by (9)}} - \underbrace{Cdc2/Cdc13^*(t)}_{=0,\text{ by hyp.}} + PP(t_8)\big), \\ &= H\big(PP(t_8)\big) \\ &= 1, \quad \text{because } PP(t_8) \geq 0 \wedge Ste9(t) = 1. \end{aligned}\right\} \tag{10}$$

$$\text{Analogous to } Cdc/Cdc13, \text{ we have that } Ste9(t) \overbrace{=}^{\text{hyp.}} 1 \overbrace{=}^{(10)} Ste9(t+1) \tag{11}$$

So,

$$\left.\begin{aligned} Cdc2/Cdc13^*(t+1) &= H\big(-Slp1(t_4) - Wee1/Mik1(t_6) - Rum1(t_3) \\ &\quad - \underbrace{Ste9(t_2)}_{=1,\text{ by (11)}} + Cdc25(t_7) - 0.5\big) \\ &= H\Big(\underbrace{-Slp1(t_4) - Wee1/Mik1(t_6) - Rum1(t_3)}_{\leq 0} \\ &\quad \underbrace{-1 + Cdc25(t_7) - 0.5}_{\leq -0.5}\Big) \\ &\leq H(-0.5) \\ &= 0. \end{aligned}\right\} \tag{12}$$

Therefore, $y_1(t+1) = (0, 1, 0)$ for any updating mode. Similarly, for $y_2(t+1)$ we have that

$$\begin{aligned} Rum1(t+1) &= H\big(-Cdc2/Cdc13(t_1) - Cdc2/Cdc13^*(t_5) + PP(t_8)\big) \\ &= Ste9(t+1) \\ &= 1 \end{aligned}$$

i.e., $y_2(t+1) = (0, 1, 0)$ for any update schedule. □

Note that $y_3 = (Cdc2/Cdc13, Ste9, Rum1, Cdc2/Cdc13^*) = (0, 1, 1, 0)$ is also an alliance that can be obtained considering both, y_1 and y_2.

2.1 Networks Associated to *Alliances*

Since *alliances* by definition remain fixed, in order to study the dynamics of the whole network we only need to analyze the subnetworks associated with the genes that do not belong to the alliance. As we see in Fig. 4 they are trees, so we may give the following property.

Corollary 1 *For the alliances,* $y_1(0) = (Cdc2/Cdc13(0), Ste9(0), Cdc2/Cdc13^*(0)) = (0, 1, 0)$ *and* $y_2(0) = (Cdc2/Cdc13(0), Rum1(0), Cdc2/Cdc13^*(0)) =$

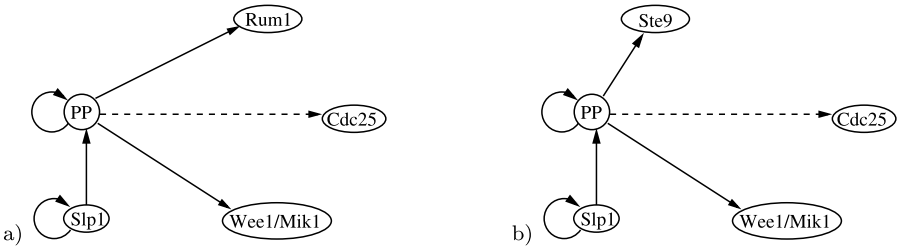

Fig. 4 (**a**) Associated network for the alliance $y_1(0) = (0, 1, 0)$. (**b**) Associated network for the alliance $y_2(0) = (0, 1, 0)$

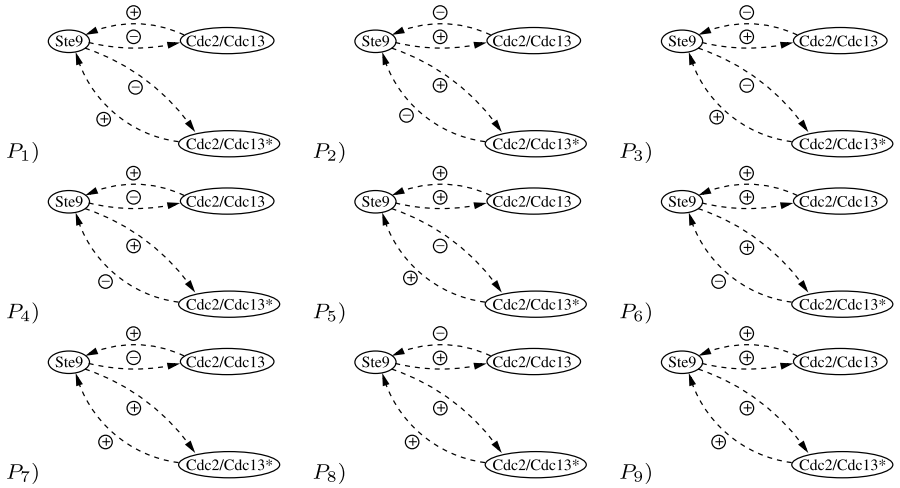

Fig. 5 The update subdigraphs associated with the partitions $P_1(Ste9), \ldots, P_9(Ste9)$, respectively, where an edge (i, j) is labeled by $\oplus$ if and only if $s(i) \geq s(j)$, otherwise, the edge is labeled by $\ominus$ (see Montalva 2011 for more details)

$(0, 1, 0)$, *the dynamics of* $x_1(t) = (0, 1, Rum1(t), Slp1(t), 0, Wee1/Mik1(t), Cdc25(t), PP(t))$ *and* $x_2(t) = (0, Ste9(t), 1, Slp1(t), 0, Wee1/Mik1(t), Cdc25(t), PP(t))$ *converges to fixed points for any updating mode, respectively.*

Proof The subnetworks associated to the alliances $y_1(0)$ and $y_2(0)$ are shown in Fig. 4(a) and (b) respectively, i.e., both are trees. Then the proof becomes direct from Proposition 1. □

Let us consider S_n as the set of updates for a given network. In our case, we have $n = 8$. Furthermore, given $z \in \{Ste9, Rum1\}$, we define the following sets (see Fig. 5):

$$P_1(z) = \left\{s \in S_n : \ s(z) < s(Cdc2/Cdc13) \wedge s(z) < s\left(Cdc2/Cdc13^*\right)\right\},$$

$$P_2(z) = \left\{s \in S_n : \ s(z) > s(Cdc2/Cdc13) \wedge s(z) > s\left(Cdc2/Cdc13^*\right)\right\},$$

$$P_3(z) = \left\{s \in S_n : \ s(Cdc2/Cdc13) < s(z) < s\left(Cdc2/Cdc13^*\right)\right\},$$

$$P_4(z) = \{s \in S_n : s(Cdc2/Cdc13^*) < s(z) < s(Cdc2/Cdc13)\},$$
$$P_5(z) = \{s \in S_n : s(Cdc2/Cdc13^*) > s(z) = s(Cdc2/Cdc13)\},$$
$$P_6(z) = \{s \in S_n : s(Cdc2/Cdc13^*) < s(z) = s(Cdc2/Cdc13)\},$$
$$P_7(z) = \{s \in S_n : s(Cdc2/Cdc13) > s(z) = s(Cdc2/Cdc13^*)\},$$
$$P_8(z) = \{s \in S_n : s(Cdc2/Cdc13) < s(z) = s(Cdc2/Cdc13^*)\},$$
$$P_9(z) = \{s \in S_n : s(Cdc2/Cdc13) = s(z) = s(Cdc2/Cdc13^*)\},$$
$$P(z) = P_1(z) \cup P_2(z) \cup P_3(z) \cup P_4(z) \cup P_7(z) \cup P_8(z).$$

Note that $S_8 = \bigcup_{i=1}^{9} P_i(Ste9) = \bigcup_{i=1}^{9} P_i(Rum1)$. So $\{P_i(z)\}_{i=1}^{9}$ is a partition of S_8. Furthermore, we have the following result.

Proposition 3 *Let the network of Yeast1 with update schedule s and let $z \in \{Ste9, Rum1\}$. If $s \in P(z)$, then the gene z becomes fixed.*

Proof We will prove by contradiction. Consider $z = Ste9$ and suppose that *Ste*9 does not become fixed. Hence, *Ste*9 changes it states from 0 to 1 (and vice versa as well), i.e.,

$$\exists t \geq 0, \quad Ste9(t) = 0 \wedge Ste9(t+1) = 1 \tag{13}$$

There are different cases according to the set $P_i(Ste9)$, which s belongs in $P(Ste9)$. In the following four cases, we will give an analytic proof while for the last two cases, the proof will be numerical, by exhaustive analysis.

Case 1: $s \in P_1(Ste9)$. Then $Cdc2/Cdc13(t+1) = 0$ and $Cdc2/Cdc13^*(t+1) = 0$ because $t_2 = t+1$ in both (4) and (7), continuing its analysis as in (8) and (12), respectively, due to (13). So, at step $t+1$, we have the alliance $y_1(t+1) = (Cdc2/Cdc13(t+1), Ste9(t+1), Cdc2/Cdc13^*(t+1)) = (0,1,0)$ that will converge to a fixed point due to Corollary 1, which contradicts our assumption that *Ste*9 is not fixed.

Case 2: $s \in P_2(Ste9)$. It is analogous to *Case1* but now, the calculations are done for the step $t+2$.

Case 3: $s \in P_3(Ste9)$. As before, $t_2 = t+1$ in both (4) and (7), but when we calculate $Cdc2/Cdc13(t+2)$ and $Cdc2/Cdc13^*(t+1)$, respectively, which implies, together with (13), that its values are zero. Since at step $t+2$ in (5) we have that $t_1 = t+2$ and $t_5 = t+1$, the calculations for $Ste9(t+2)$ are like in (10). Thus, it is easy to check that $Cdc2/Cdc13^*(t+2) = 0$, producing the same contradiction of *Case 1* but, in the step $t+2$.

Case 4: $s \in P_4(Ste9)$. It is similar to *Case 3* but now, $t_2 = t+1$ in $Cdc2/Cdc13^*(t+2)$ and $Cdc2/Cdc13(t+1)$ respectively, which implies, together with (13) that $Ste9(t+2) = 1$ and consequently $Cdc2/Cdc13(t+2) = 0$, producing the same contradiction of *Case 3*.

The analysis of the above four cases becomes identical for $z = Rum1$, because Eq. (5) is equal to Eq. (6).

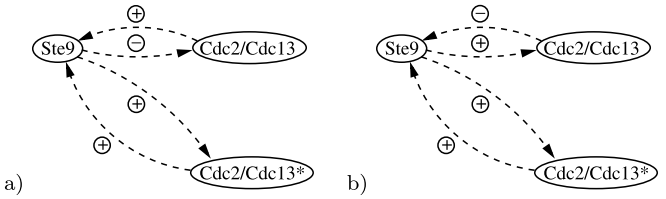

Fig. 6 Update subdigraphs associated with the nodes z, $Cdc2/Cdc13$ and $Cdc2/Cdc13^*$ for the Cases 5 and 6 in the proof of Proposition 3, $z \in \{Ste9, Rum1\}$. (**a**) Update subdigraph of *Case 5:* $s \in P_7(z)$. (**b**) Update subdigraph of *Case 6:* $s \in P_8(z)$

Table 1 Decomposition of all the different deterministic dynamics associated to the fission yeast cell-cycle model. $|D(X)|$ is the number of different dynamics obtained after running the algorithms developed in Montalva (2011) for the enumeration of the equivalence classes associated with *Yeast1*. $|D^{cl}(X)|$ is the number of different dynamics in $D(X)$ having at least one limit cycle

X	$\lvert D(X)\rvert$	$\lvert D^{cl}(X)\rvert$
$P_1(Ste9)$	3345	1085
$P_2(Ste9)$	3361	1015
$P_3(Ste9)$	1350	494
$P_4(Ste9)$	1712	486
$P_5(Ste9)$	1275	638
$P_6(Ste9)$	1376	666
$P_7(Ste9)$	1376	501
$P_8(Ste9)$	1016	403
$P_9(Ste9)$	539	225
Total	15350	5513

Case 5: $s \in P_7(z)$, $z \in \{Ste9, Rum1\}$. We analyze this case using the algorithmic tools developed in Montalva (2011) for the enumeration of update digraphs. Note that there is only one update subdigraph associated with the nodes z, $Cdc2/Cdc13$ and $Cdc2/Cdc13^*$ (see Fig. 6(a)). Hence, to show that z becomes fixed, we must do an exhaustive analysis of all the dynamics generated by update digraphs of the original network containing the above as update subdigraph. Since there are dynamics that only have fixed points (and here, clearly z becomes always fixed), we focus on those that have at least one limit cycle (because here, the state of z eventually can change from 0 to 1 and vice versa). Thus, the numerical results shows that there are 1,376 update digraphs associated with the original network that have as update subdigraph those shown in Fig. 6(a), all of them generating a different dynamic and only 501 of such dynamics having at least one limit cycle (see the row of the particular case $P_7(Ste9)$ in Table 1), in all of them $z = 0$, i.e., z becomes fixed.

Case 6: $s \in P_8(z)$, $z \in \{Ste9, Rum1\}$. It is similar to *Case 5,* but now the numerical results shows that there are 1,376 update digraphs associated with the original network that have as update subdigraphs of those shown in Fig. 6(b), only 1,016 of them generating a different dynamic and only 403 of such dynamics having at least one limit cycle (see row of the particular case $P_8(Ste9)$ in Table 1), in all of them $z = 0$, i.e., z becomes fixed. □

Corollary 2 *If* $Ste9(t) = constant = 1$ ($Rum1(t) = constant = 1$), *then every configuration converges to the alliance* y_1 (y_2) *and consequently the whole dynamic has no limit cycles for each update schedule.*

Proof The alliance y_1 (y_2) is obtained directly replacing $Ste9 = 1$ ($Rum1 = 1$) in the definitions of $Cdc2/Cdc13$ and $Cdc2/Cdc13^*$, respectively. The absence of limit cycles is due to Corollary 1. □

In this way, if *Yeast1* has at least one limit cycle C for some $s \in P(z)$, $z \in \{Ste9, Rum1\}$, then the counter-reciprocal of Corollary 2 implies necessarily that C is such that $z(t) = 0$, for all $t > t'$, for some $t' \geq 0$. Thus, in *Yeast1* the only possibility to find limit cycles, with z not constant, are in partitions not belonging to $P(z)$, i.e., $P_5(z)$, $P_6(z)$ or $P_9(z)$. Hence, using the algorithms related with the *equivalent* classes (Montalva 2011), we can determine exactly the number of different dynamics with at least one limit cycle. To do that, let $X = P_i(z)$ for some $z \in \{Ste9, Rum1\}$, for some $i \in \{1, \ldots, 9\}$. We define $D(X)$ as the set of different dynamics associated to the update schedules in X and $D^{cl}(X)$ as the set of dynamics in $D(X)$ having at least one limit cycle. Thus, the computational results for the *Yeast1* network are summarized in Table 1.

We remember that the number of update schedules associated to a network of n nodes is $T_n = \sum_{k=0}^{n-1} \binom{n}{k} T_k$, where $T_0 \equiv 1$. In our case, $T_8 = 545835$ but with the previous analysis and the equivalent classes algorithm we only have to check about 3 % of the T_8 update schedules. More precisely, there are only 15,350 update schedules with different dynamics and between them, only 5,513 with limit cycles.

Examples of the complete dynamics of *Yeast1*, with and without *Start* and *SK*, for some specific updates, are shown in Figs. 7, 8, and 9. In general, from the three

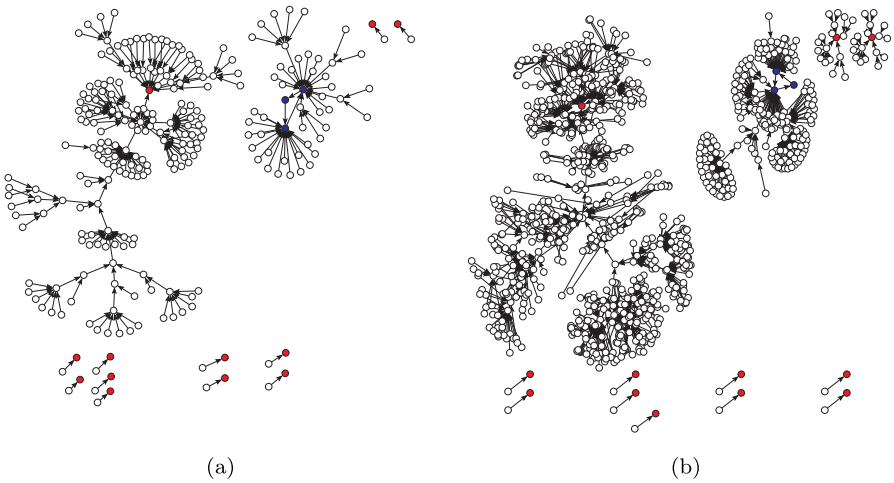

Fig. 7 (**a**) State transition graph for *Yeast1*, without *Start* and *SK*, using the parallel updating scheme. (**b**) State transition graph for the complete *Yeast1*, using the parallel updating scheme. The *twelve red circles* represent the fixed point states, the *three blue circles* represent the states that belong to the limit cycle

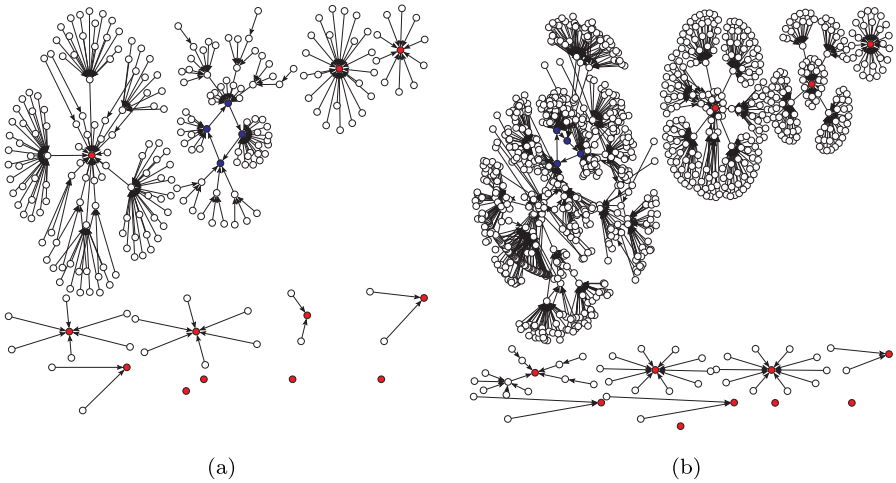

Fig. 8 (**a**) State transition graph for *Yeast1*, without *Start* and *SK*, using the block-sequential updating mode: $s(Wee1/Mik1) = s(Cdc25) < s(Cdc2/Cdc13) = s(Cdc2/Cdc13^*) = s(Ste9) < s(Rum1) = s(Slp1) = s(PP)$. (**b**) State transition graph for the complete *Yeast1*, using the block-sequential updating mode: $s(Wee1/Mik1) = s(Cdc25) < s(Cdc2/Cdc13) = s(Cdc2/Cdc13^*) = s(Ste9) < s(Rum1) = s(Slp1) = s(PP) = s(Start) = s(SK)$. The *twelve red circles* represent the fixed point states, the *four blue circles* represent the states that belong to the limit cycle

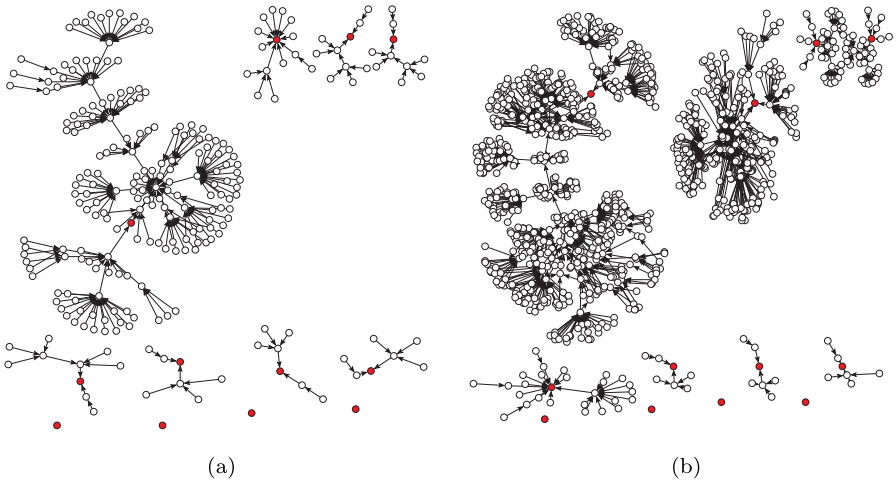

Fig. 9 (**a**) State transition graph for *Yeast1*, without *Start* and *SK*, using the block-sequential updating mode: $s(Ste9) < s(PP) < s(Wee1) = s(Cdc25) < s(Cdc2/Cdc13) = s(Rum1) = s(Slp1) = s(Cdc2/Cdc13^*)$. (**b**) State transition graph for the complete *Yeast1*, using the block-sequential updating mode: $s(Ste9) < s(PP) < s(Wee1) = s(Cdc25) < s(Start) = s(SK) = s(Cdc2/Cdc13) = s(Rum1) = s(Slp1) = s(Cdc2/Cdc13^*)$. The *twelve red circles* represent the fixed-point states; there are no limit cycles

Table 2 (0, 0, 0, 1, 1, 0, 0, 0, 1, 0) and (0, 0, 0, 1, 1, 0, 0, 1, 0, 0) are the fixed points of the *Yeast1* network with minimum and maximum basin average size percentage (BASP), respectively, and it was calculated over the 15,350 different dynamics shown in Table 1. Since all the dynamics with a limit cycle showed that contains only one each, we added the BASP for the limit cycles that was calculated over the 5,513 different dynamics having a limit cycle. Observe that the values are consistent with those shown in Table 3

Attractor	BASP
(0, 0, 0, 1, 1, 0, 0, 0, 1, 0)	2 %
(0, 0, 0, 1, 1, 0, 0, 1, 0, 0)	51 %
Limit cycle	39 %

Table 3 Attractors of *Yeast1* under the parallel updating mode. There are two types of attractors, fixed points (FP) and limit cycles (LC)

Attractor	Type	Basin size	*Start*	*SK*	*Cdc*2/ *Cdc*13	*Ste*9	*Rum*1	*Slp*1	*Cdc*2/ *Cdc*13∗	*Wee*1/ *Mik*1	*Cdc*25	*PP*
1	FP	762	0	0	0	1	1	0	0	1	0	0
2	LC	208	0	0	0	0	0	0	0	0	1	1
	LC	0	0	0	0	0	0	1	0	0	1	0
	LC	0	0	0	1	1	1	0	1	1	0	0
3	FP	18	0	0	0	0	1	0	0	1	0	0
4	FP	18	0	0	0	1	0	0	0	1	0	0
5	FP	2	0	0	0	1	0	0	0	0	0	0
6	FP	2	0	0	0	1	0	0	0	0	1	0
7	FP	2	0	0	0	1	0	0	0	1	1	0
8	FP	2	0	0	0	0	1	0	0	0	0	0
9	FP	2	0	0	0	0	1	0	0	0	1	0
10	FP	2	0	0	0	0	1	0	0	1	1	0
11	FP	2	0	0	0	1	1	0	0	0	0	0
12	FP	2	0	0	0	1	1	0	0	0	1	0
13	FP	2	0	0	0	1	1	0	0	1	1	0

figures, it is clear that different unique limit cycles appear and in some cases (Fig. 9) there are no limit cycles.

On the other hand, an analysis of the attractors with minimum and maximum basin of attraction of the network is shown in Table 2 in order to compare with the original results of (Davidich and Bornholdt 2008) which are shown in Table 3. In this table, the fixed point (0, 0, 0, 1, 1, 0, 0, 1, 0, 0) has the largest attraction basin, with 762 of the $2^{10} = 1024$ configurations (i.e., a 74 %), there are nine fixed points with the smallest attraction basin, each one of them of size 2 (i.e., a 0.2 %) and the only limit cycle has attraction basin of size 208 (i.e., a 20 %). Comparing these results with ours shown in Table 2, we can validate the maximum attraction basin with an average size of 51 % and we can specify that the fixed point (0, 0, 0, 1, 1, 0, 0, 0, 1, 0) has the smallest attraction basin with an average size of 2 %, both percentages calculated over the 15,350 different dynamics, while the average size of the attraction basin in the 5,513 dynamics, with a limit cycle, increases to 39 %.

3 Cell-Cycle Network of the Budding Yeast

In this section, we will carry out a similar analysis as the one done with *Yeast1*, i.e., to determine the dynamics for every deterministic updating mode. Actually, we will prove that every update admits only fixed points. To achieve this, two propositions are proven as well as the use of Proposition 1.

Let us consider *Yeast2* given in Fig. 10, which follows the same configuration as (Li et al. 2004) where green/solid edges represent positive weights, the red/dashed edges represent negative weights, and the yellow/solid loops represent self-degradation, which are modeled mathematically by a negative weight.

This model also uses Eq. (2) to compute the output of each gene, with a weight matrix and a threshold vector shown in Fig. 11.

It is direct that in one step *Cln*3 becomes fixed at value 0, so genes *MBF* and *SBF* will remain fixed, if not, there are $t_1, t_2 > 1$ such that:

(1) $MBF(t_1) = 1$ (similarly, $SBF(t_1) = 1$) and $MBF(t_1 + 1) = 0$ ($SBF(t_1 + 1) = 0$)
(2) $MBF(t_2) = 0$ (similarly, $SBF(t_2) = 0$) and $MBF(t_2 + 1) = 1$ ($SBF(t_2 + 1) = 1$)

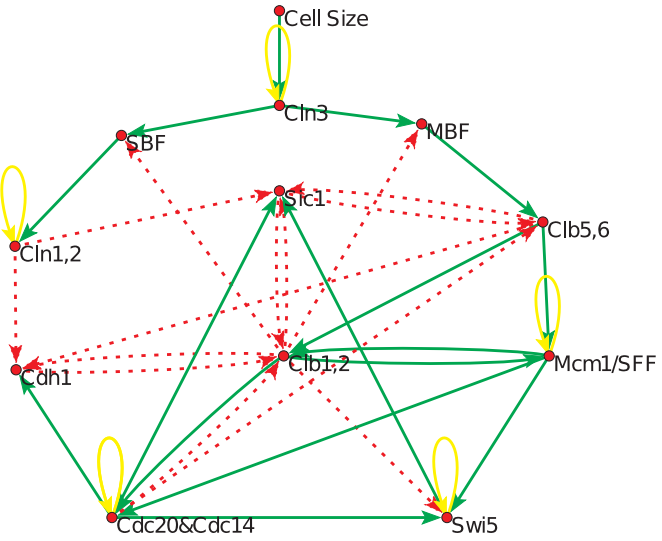

Fig. 10 The budding yeast cell-cycle threshold Boolean network. Using the same configuration as (Li et al. 2004), the *green/solid edges* represent positive weights (activations), the *red/dashed edges* represent negative weights (inhibitory). The *yellow/solid loops* represent self-degradation

$$W = \begin{array}{c} \\ Cln3 \\ MBF \\ SBF \\ Cln1,2 \\ Cdh1 \\ Swi5 \\ Cdc20 \\ Clb5,6 \\ Sic1 \\ Clb1,2 \\ Mcm1 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} Cln3 & MBF & SBF & Cln1,2 & Cdh1 & Swi5 & Cdc20 & Clb5,6 & Sic1 & Clb1,2 & Mcm1 \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 \end{pmatrix} \end{array} \quad \Theta = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Fig. 11 Weight matrix and threshold vector for *Yeast2*

but the second point implies necessarily that $Cln3 = 1$, which contradicts the fact that $Cln3$ becomes fixed at value 0. Therefore, the analysis continues similar to the previous section but now focusing in what happens when $MBF = 0$ and $MBF = 1$. On the other hand, it is direct to check that the vector $Y = (MBF(t), Clb5,6(t), Clb1,2,(t), Mcm1/SFF(t), Swi5(t), Cdc20/Cdc14(t)) = (0,0,0,0,0,0)$ is an *alliance*.

Our principal result is the following.

Proposition 4 *For any update schedule, Yeast2 admits only fixed points.*

To prove the previous proposition we will analyze two cases: $MBF = 0$ and $MBF = 1$. In the first case, we will also use the Proposition 1 (fixed point dynamical behavior). For the case $MBF = 1$, we analyze every update schedule for a bounded number of dynamical equivalence classes. The next proposition studies the case $MBF = 0$.

Proposition 5 (Convergence to the Alliance Y) *Suppose* $MBF(t) = 0$ *for any* $t \geq 0$ *then there exists* $T > 0$ *such that,* $Swi5(t) = Cdc20/Cdc14(t) = Clb5,6(t) = Clb1,2(t) = Mcm1/SFF(t) = 0, \forall t > T$.

Proof It is easy to see from the network structure that $Clb5,6(t) = 0$ for some time $t \geq 0$ implies $Clb5,6(t) = 0, \forall t \geq t'$, i.e. node $Clb5,6$ remains stable at value 0. Let us suppose now that there exists t^* such that $Clb5,6(t) = 1, \forall t > t^*$, then necessarily, $Sic(t) = Cdc20/Cdc14(t) = 0, \forall t \geq t^*$.

On the other hand, $Clb5,6(t) = 1, \forall t > t^*$ implies that $Mcm1/SFF(t) = 1, \forall t > t^*$ and this latest implies that $Cdc20/Cdc14(t) = 1, \forall t > t^*$, which is a contradiction.

So, regardless of the value of $Clb5,6(0)$ and the update schedule, there exists a time t_1 such that $Clb5,6(t) = 0, \forall t > t_1$.

Let us now analyze the genes $Clb1,2$ and $Mcm1/SFF$. It is direct that if for some t, $Clb1,2(t) = Mcm1/SFF(t) = 0$, since $MBF = 0$, these genes remain at this value independently of the update schedule. Now we have to analyze the other cases:

(1) $Clb1,2(t) = 0$ and $Mcm1/SFF(t) = 1$ for some $t \geq t_1$. Let us consider first the case gene $Clb1,2$ is updated before $Mcm1/SFF$, i.e., $s(Clb1,2) < s(Mcm1/SFF)$. So, in the next step, say t', $Mcm1/SFF(t') = 0$ and necessarily we must have $Clb1,2(t') = 1$ (if not, we will have both at zero so it remains forever). It is important to point out that regardless of the update mode of the other genes, since at each step at least one of the two genes $Clb1,2$ and $Mcm1/SFF$ considered is at value 1, then for some $t^* > t'$ $Cdc20/Cdc14(t^*) = 1$. From this fact, it is direct that there exists $t > t'$ such that $SBF(t) = 0$, $Cln1,2(t) = 0$, $Swi5(t) = 1$ and $Cdh1(t) = 1$. From that, we conclude that for some $t > 0$, $Clb1,2(t) = 0$, so also $Mcm1/SFF(t) = 0$. The case $s(Clb1,2) > s(Mcm1/SFF)$ is analogous.

(2) Suppose now that both genes are updated in parallel, i.e., $s(Clb1,2) = s(Mcm1/SFF)$. Since at each step at least one of the two genes has to be at value 1, then as in the previous case the gene $Cd20/Cdc14$ becomes constant at value 1. So, as in the first case, at some step $t > 0$ the gene $Clb1,2$ will be fixed at value zero, so will be $Mcm1/SFF$.

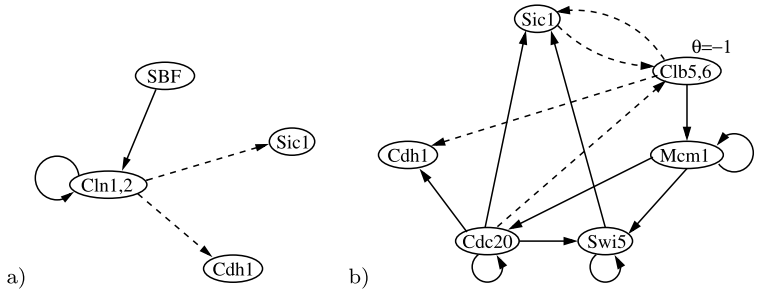

Fig. 12 (**a**) Reduced cell-cycle network of the budding yeast when $MBF = 0$. (**b**) Reduced cell-cycle network of the budding yeast when $MBF = 1$

From the previous analysis, we have proved that there exists $t^* > 0$ such that $Clb5,6(t^*) = Clb1,2(t^*) = Mcm1/SFF(t^*) = 0$. From that, it is direct that in some step $t > t^*$ $Cdc20/Cdc14(t) = Swi5(t) = 0$, which proves the property. □

It is important to point out that when we are in the same conditions as in the previous proposition, we reach the alliance:

$MBF(t) = Swi5(t) = Cdc20/Cdc14(t) = Clb5,6(t) = Clb1,2(t) = Mcm1/SFF(t) = 0$, for $t > T$, and the subnetwork to analyze is the tree shown in Fig. 12(a). Furthermore, every local application is a Heaviside function with non- negative diagonal interactions. So, Proposition 1 assures that such network admits only fixed points, that we have called A, B, C, D, and E in Table 4 and marked with italic font.

To complete the analysis of the original network, it is also necessary to analyze what happens when $MBF = 1$ is fixed. To do so, we use again the equivalence class analysis tools to prove that the network also has no limit cycles, whatever update schedule is used. We begin this analysis doing some useful observations.

Note that $T_{11} = 1.622.632.573$ and each dynamic has $2^{11} = 2048$ configurations. So, in order to improve the computational time used in the calculation of such classes, we reduce the network observing that if $MBF = 1$ is fixed, then necessarily $Clb1,2 = 0$ is also fixed; moreover, the effect of $MBF = 1$ over $Clb5,6$ is equivalent to a threshold $\theta = -1$ over it. Since $Cln3 = 0$, then we can divide the rest of the analysis in two cases; node $SBF = \text{const} = 0$ and $SBF = \text{const} = 1$.

In the first one, necessarily node $Cln1,2 = 0$ is fixed, so, at this point, the network it is reduced in 5 nodes ($Cln3 = 0$, $MBF = 1$, $Clb1,2 = 0$, and $SBF = 0$), leaving the network shown in Fig. 12(b). Since we know from (Li et al. 2004) that the original network has 7 fixed points and none of them has $MBF = SBF = 1$, then the second case to analyze is discarded. Thus, we use the equivalence class tools for the first case and the fixed points obtained that we have called F and G in Table 4 and marked with bold font.

Since the equivalence class tools allow us to have the full spectrum of different dynamics associated with the network, we will answer, among other things, if effectively the fixed point $G1$ of (Li et al. 2004) has the largest basin of attraction between all the possible dynamics of the network. This answer is shown in Table 4.

Two updating schedules are shown as examples. Figure 13(a), without considering $Cln3$, and Fig. 13(b), with $Cln3$, shows the state transition graph for the parallel

Table 4 The attractors of the cell-cycle network of the budding yeast. The network has no limit cycle. The simulations performed also confirm that $G1$ is the largest attractor with an average basin of 79.5 % between the full spectrum of the dynamics of the network

Attractor	Type	BS	*Cln*3	*MBF*	*SBF*	*Cln*1, 2	*Cdh*1	*Swi*5	*Cdc*20/*Cdc*14	*Clb*5, 6	*Sic*1	*Clb*1, 2	*Mcm*1/*SFF*
A = G1	PF	1764 (86 %)	0	0	*0*	*0*	*1*	0	0	0	*1*	0	0
B	PF	151 (7.4 %)	0	0	*1*	*1*	*0*	0	0	0	*0*	0	0
F	PF	109 (5.3 %)	0	1	0	0	**1**	**0**	**0**	**0**	**1**	0	**0**
C	PF	9 (0.4 %)	0	0	*0*	*0*	*0*	0	0	0	*1*	0	0
G	PF	7 (0.3 %)	0	1	0	0	**0**	**0**	**0**	**0**	**1**	0	**0**
D	PF	7 (0.3 %)	0	0	*0*	*0*	*0*	0	0	0	*0*	0	0
E	PF	1 (0.1 %)	0	0	*0*	*0*	*1*	0	0	0	*0*	0	0

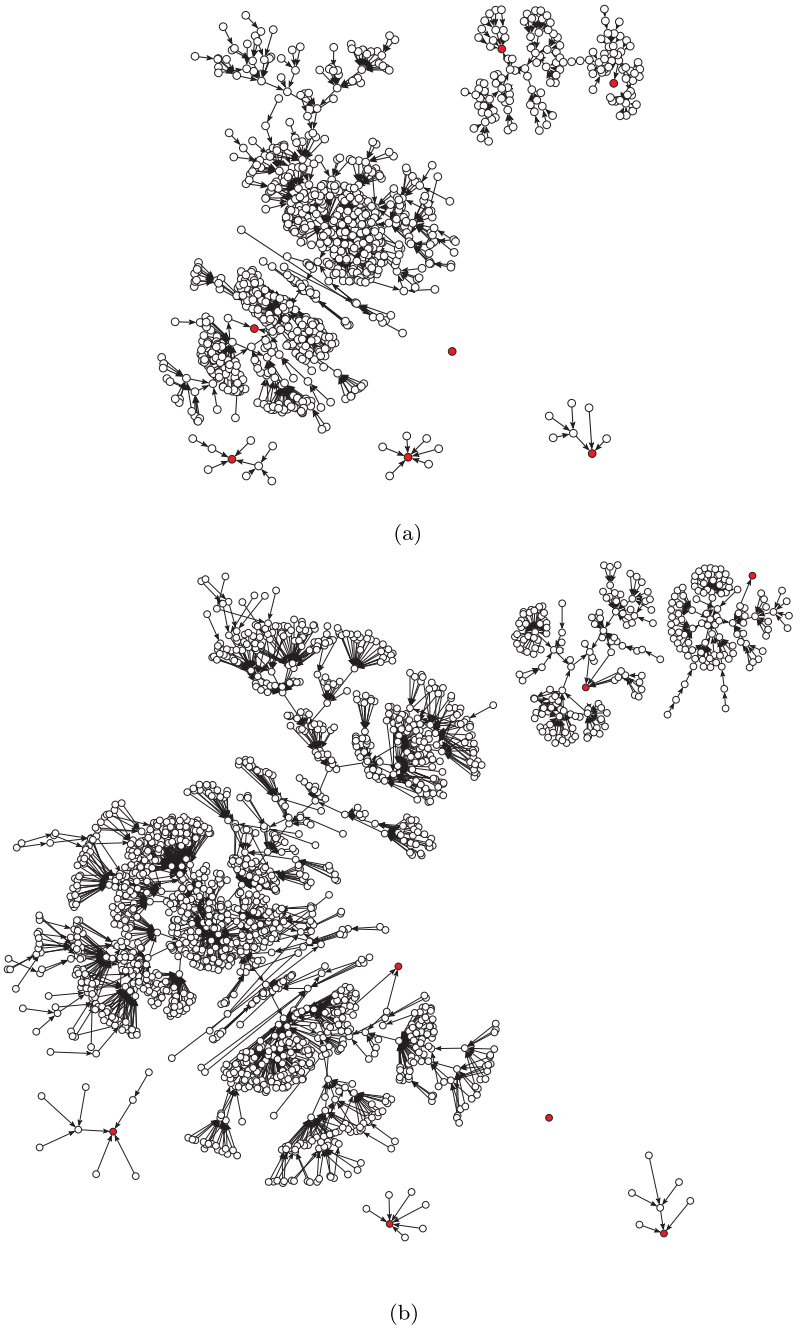

Fig. 13 (**a**) State transition graph for *Yeast2*, without *Cln3*, using the parallel updating scheme. (**b**) State transition graph for the complete *Yeast2*, using the parallel updating scheme. The *seven red circles* represent the fixed point states, there are no limit cycles

updating mode, whereas, Fig. 14(a), without *Cln*3, and Fig. 14(b), with *Cln*3, corresponds to the state transition graph for another update. There are no cycles, and the same fixed point, in both updating modes, have the largest basin of attraction.

4 Conclusion

We have developed mathematical and computational tools for the robustness analysis of two particular Boolean networks of biological interest; the fission yeast cell-cycle network (*Yeast1*) and the budding yeast cell cycle network (*Yeast2*), which have previously been analyzed only for the parallel updating mode. These tools (alliances and equivalent classes) allow the study of all the possible deterministic dynamics of these networks, which is difficult due to the highly exponential nature of the problem.

Results showed that for the *Yeast1* model, there are 15,350 different dynamics, from which 5,513 contain only one limit cycle each, with cycle lengths in the range of 2 to 4 (the original model presented in Davidich and Bornholdt (2008) showed a limit cycle of length 3 under the parallel updating mode). Also, the fixed point with the largest basin of attraction using the parallel updating mode maintains this property for all the different updating modes. Therefore, although the *Yeast1* network has different limit cycles for different updating modes, it could be considered robust in the sense that the basin of attraction for that fixed point for all the deterministic dynamics does not vary that much. In the case of *Yeast2*, theoretical results show that for all the deterministic dynamics, there are no limit cycles. Similar to *Yeast1*, the fixed point with the largest basin of attraction in the parallel updating mode, has the largest basin of attraction for all the different dynamics. Therefore, one could conclude that the *Yeast2* network is more robust in the sense that no limit cycles appear and one fixed point has the largest basin of attraction throughout all the dynamics. For the other six fixed points, computational results show that their basin of attraction varies significantly for different updating modes. The analysis carried out on the two biological models, showed the usefulness and potential of the proposed techniques, but they also enabled the discovery of new information about these models, such as the number of different dynamics they have, how many limit cycles can appear, limit cycle lengths, alliances for each model, the exact average of the basin of attraction for all the dynamics, updating schemes that can assure the stability of some nodes. While the tools used for equivalence classes are general for any network, the alliances are useful when the given network has at least one. In general, given that all the deterministic dynamics are obtained, any information of the two networks studied, under different updating schemes, can be computed.

Although the tools for Boolean network robustness analysis presented in this paper were applied to the *Yeast1* and *Yeast2* model, future research will consider the use of these techniques to study other networks such as the mammalian cell cycle network (Fauré et al. 2006).

Acknowledgements The authors would like to thank Conicyt-Chile under grant Fondecyt 1100003 (E.G.), Fondecyt 11110088 (G.A.R.), Fondecyt 3130466 (M.M.), Basal (Conicyt)-CMM (E.G. M.M.), and ANILLO ACT-88 (E.G., M.M., G.A.R.) for financially supporting this research. E. Goles would like to thank TIM3, CNRS, Sophia-Antipolis, France, where part of this work was conducted.

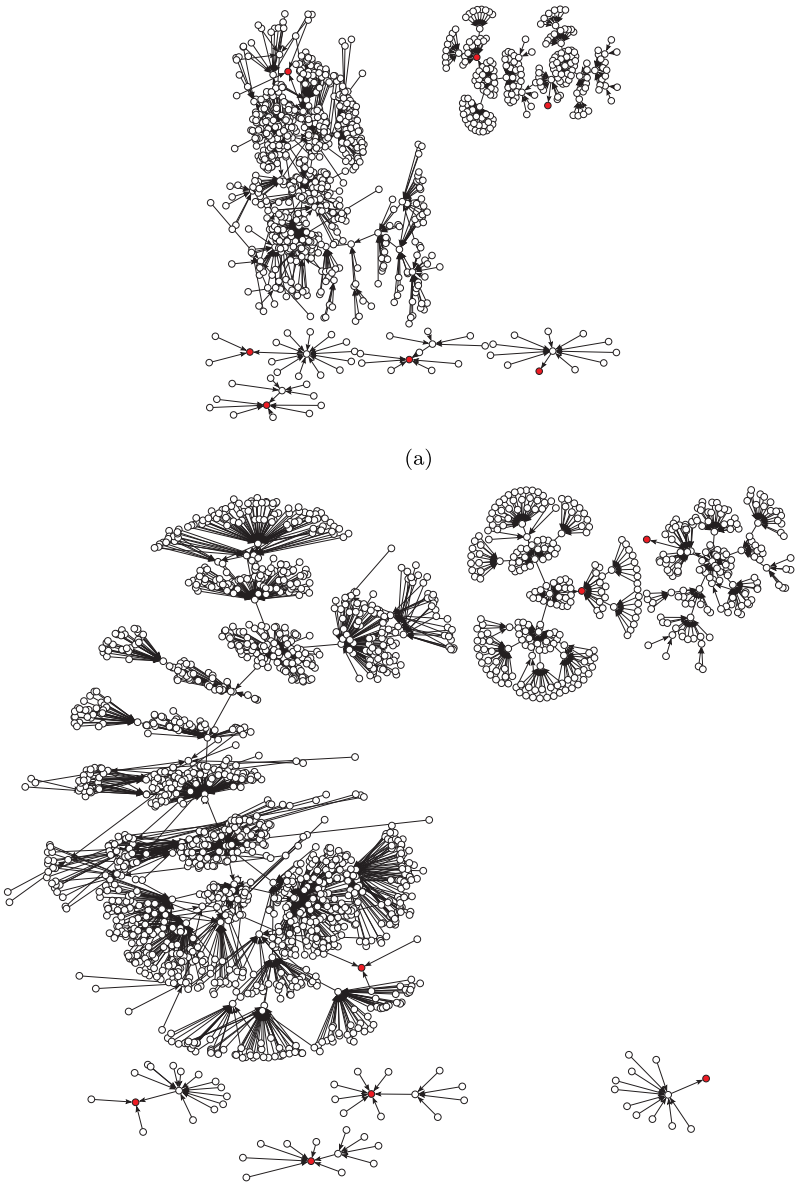

Fig. 14 (**a**) State transition graph for *Yeast2*, without *Cln3*, using the block-sequential updating mode: $s(Swi5) = s(Cdc20\&Cdc14) = s(Clb5,6) < s(MBF) = s(Sic1) < s(SBF) = s(Clb1,2) = s(Mcm1/SFF) < s(Cln1,2) < s(Cdh1)$. (**b**) State transition graph for the complete *Yeast2*, using the block-sequential updating mode: $s(Swi5) = s(Cdc20\&Cdc14) = s(Clb5,6) < s(MBF) = s(Sic1) < s(Cln3) = s(SBF) = s(Clb1,2) = s(Mcm1/SFF) < s(Cln1,2) < s(Cdh1)$. The *seven red circles* represent the fixed point states; there are no limit cycles

Appendix A: The Equivalence Classes Tools

Let G be a digraph and s an update schedule over $V(G)$. Consider the label function lab_s defined by

$$\forall (j,i) \in A(G), \quad lab_s(j,i) = \begin{cases} \oplus & \text{if } s(j) \geq s(i), \\ \ominus & \text{if } s(j) < s(i). \end{cases}$$

The pair (G, lab_s) is called *update digraph* (see Figs. 16, 5, and 6 as examples). In (Aracena et al. 2009) was defined equivalence classes $[s]_G$ of deterministic update schedules yielding the same update digraph, i.e.,

$$[s]_G = \{s' \text{ update schedule: } (G, lab_s) = (G, lab_{s'})\}$$

It was proven that two update schedules in the same class (i.e., two update schedules having the same update digraph) yield exactly the same dynamical behavior. The above implies that the number of equivalence classes of the interaction digraph in a network gives us the maximum number of possible different dynamics that can exist. From this arises the interest in developing enumeration algorithms for these classes. In this context, the following enumeration algorithm was developed in Aracena et al. (submitted) (see Algorithms 1, 2 and 3).

Notation and definitions:

- A *partial update schedule* is an update schedule over the vertices of some $G' \subseteq G$. A *block* of s is the set $B_i = \{v \in V(G') : s(v) = i\}$, $1 \leq i \leq m$. The number of blocks of s is denoted by $nb(s)$ and s is denoted by $s = (j \in B_1)(j \in B_2) \cdots (j \in B_{nb(s)})$.
- For $X \subseteq V(G) \setminus \bigcup_{i=1}^{nb(s)} B_i$, the operation $*$ is defined as follows:

$$s * X = (j \in B_1)(j \in B_2) \cdots (j \in B_{nb(s)})(j \in X)$$

  where, $\emptyset * X = (j \in X)$.
- For $G = (V, A)$ a digraph and $C, D \subseteq V$ we have that $G_{(C,D)} = (C \cup D, A(G) \cap ((C \cup D) \times (C \cup D)))$ and $lab_{(C,D)} : A(G_{(C,D)}) \longrightarrow \{\oplus, \ominus\}$ is defined by

$$lab_{(C,D)}(u,v) = \begin{cases} \ominus, & u \in C \wedge v \in D, \\ \oplus, & \text{otherwise} \end{cases}$$

For more details, see (Aracena et al. submitted).

Algorithm 1 `EqClass(G)`

Require: G, the interaction digraph of a BN
Ensure: `UD`, the set of all equivalence classes of G composed by one representative update schedule for each class
`UD` ← `DigraphUD(∅,∅,V)`
return `UD`

Algorithm 2 `DigraphUD`(s, A, B)

Require: A, B subsets of vertices of a digraph G and s a partial update schedule of a subdigraph of G
Ensure: `UD`, a set of partial update schedules for G
`UD`$\leftarrow \emptyset$;
$U \leftarrow B_{nb(s)}$;
if $U = A = \emptyset$ **then**
`UD` $=$ `UD` $\cup \{s_B = (j \in B)\}$;
for all $A_0 \subset B$ such that $A_0 \neq \emptyset$ with decreasing size **do**
$B_0 = B - A_0$;
`UD` $=$ `UD` $\cup$ `DigraphUD`$(\emptyset, A_0, B_0)$;
end for
else
if `MoveTest`$(U, A) = 0$ **then**
if `MoveTest`$(A, B) = 0$ **then**
`UD` $=$ `UD` $\cup \{(s * A) * B\}$;
end if
if $|B| > 1$ **then**
for all $A_1 \subset B$ such that $A_1 \neq \emptyset$ with decreasing size **do**
$B_1 = B - A_1$;
`UD` $=$ `UD` $\cup$ `DigraphUD`$(s * A, A_1, B_1)$;
end for
end if
end if
end if
return `UD`

Algorithm 3 `MoveTest`(C, D)

Require: C, D subsets of vertices of a digraph G
Ensure: An index 1 if it is possible to move nodes from D to C without changing the update digraph induced by $C \cup D$, an index 0 in other case
if $C = \emptyset$ **then**
return 0
else
if $\exists H \subseteq D$ such that $(G_{(C,D)}, lab_{(C,D)}) = (G_{(C \cup H, D-H)}, lab_{(C \cup H, D-H)})$ **then**
return 1
else
return 0
end if
end if

Appendix B: Example Using Alliances and the Equivalence Class Tools

Figure 15(a) is an example of a Boolean network that has 3 nodes with threshold Boolean functions. The idea is to illustrate how the notion of alliance and the math-

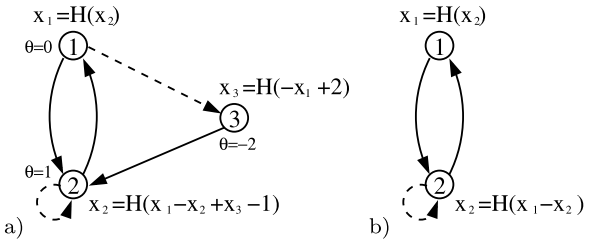

Fig. 15 (**a**) A BN of 3 nodes with threshold Boolean functions defined as x_1, x_2, x_3, and thresholds $\theta = 0$, $\theta = 1$, and $\theta = -2$, respectively. *Dotted lines* represent negative interactions (weight $= -1$) while the others are positive interactions (weight $= 1$). (**b**) The BN of (**a**) is reduced after considering the alliance $x_3 = 1$. Note that the effect of $x_3 = 1$ over x_2 is equivalent to write $x_2 = H(x_1 - x_2)$ while x_1 remains the same

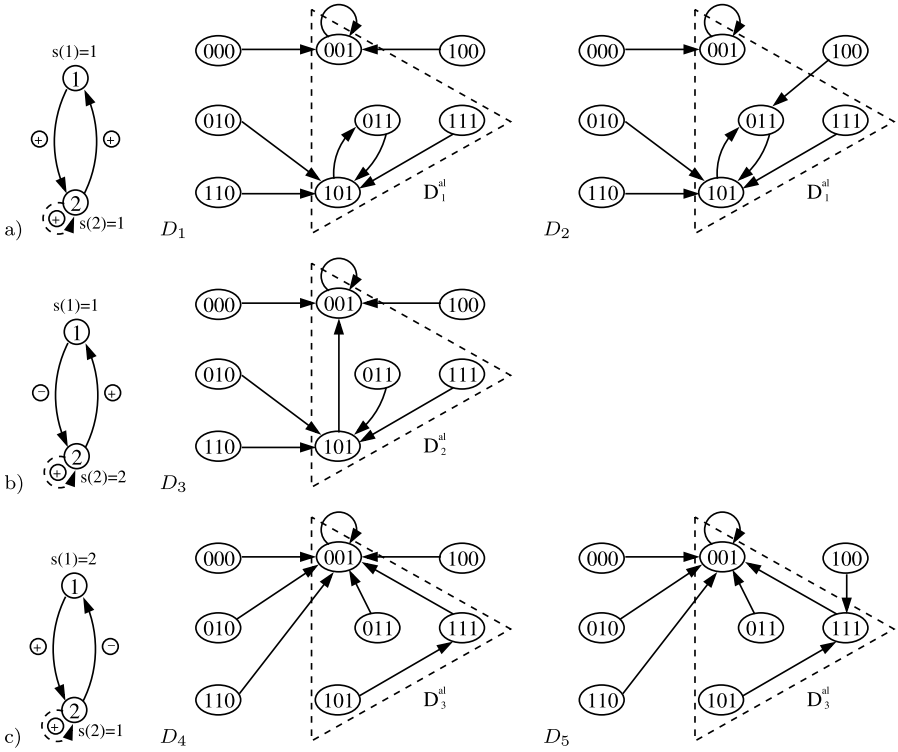

Fig. 16 In (**a**), (**b**) and (**c**) are specified the update schedules, its associated update digraphs and its dynamics (shown in *dashed triangles* and denoted by D_1^{al}, D_2^{al} and D_3^{al} respectively) inside of the different dynamics of the original network: $D_1, \ldots, D_5$

ematical tools for the enumeration of equivalence classes, in a given digraph, can be useful for knowing the type of attractors of the network as well as the different dynamics that can be obtained when it is updated in every deterministic way.

We begin by noting that $x_3 = H(-x_1 + 2) = 1$ is an alliance because $-x_1 + 2 \geq 1$, regardless the value of x_1 and the update schedule used. Moreover, any of the $2^3 = 8$ possible configurations will converge to another configuration such that necessarily $x_3 = 1$ (see Fig. 16). In particular, those belonging to some attractor (fixed point or limit cycle). Thus, to know all the possible attractors for this network it is sufficient to analyze the dynamics of the reduced network shown in Fig. 15(b).

Thus, the reduced network in Fig. 15(b) has $T_2 = 3$ update schedules, each one defining a different update digraph (update schedules and update digraphs are shown in Fig. 16(a), (b) and (c)). Moreover, these 3 update schedules generate 3 different dynamics: D_1^{al}, D_2^{al}, and D_3^{al} which are specified in dashed triangles inside of the 5 different dynamics associated with the original network: $D_1, \ldots, D_5$. These are obtained in a similar way than the reduced network mentioned above (see Fig. 16).

References

Aracena, J., Goles, E., Moreira, A., & Salinas, L. (2009). On the robustness of update schedules in Boolean networks. *Biosystems*, *97*, 1–8.

Aracena, J., Demongeot, J., Fanchon, E., & Montalva, M. (submitted). *On the number of different dynamics in Boolean networks with deterministic update schedules*. Preprint, Universidad de Concepción, Chile.

Davidich, M. I., & Bornholdt, S. (2008). Boolean network model predicts cell cycle sequence of fission yeast. *PLoS ONE*, *3*(2), e1672.

Demongeot, J., Elena, A., & Sené, S. (2008). Robustness in regulatory networks: a multi-disciplinary approach. *Acta Biotheor.*, *56*(1–2), 27–49.

Demongeot, J., Amor, H. B., Elena, A., Gillois, P., Noual, M., & Sené, S. (2009). Robustness in regulatory interaction networks. A generic approach with applications at different levels: physiologic, metabolic and genetic. *Int. J. Mol. Sci.*, *10*, 4437–4473.

Elena, A. (2009). *Robustesse des réseaux d'automates booléens à seuil aux modes ditération. Application à la modélisation des réseaux de régulation génétique*. PhD thesis Université Joseph Fourier, Grenoble, France.

Fauré, A., Naldi, A., Chaouiya, C., & Thieffry, D. (2006). Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics*, *22*, e124–e131.

Gershenson, C. (2004). Updating schemes in random boolean networks: do they really matter? In J. Pollack, M. Bedau, P. Husbands, T. Ikegami, & R. A. Watson (Eds.), *Proceedings of the ninth international conference on the simulation and synthesis of living systems (ALife IX)*, Boston, USA (pp. 238–243). Cambridge: MIT Press.

Goles, E., & Salinas, L. (2008). Comparison between parallel and serial dynamics of Boolean networks. *Theor. Comput. Sci.*, *396*, 247–253.

Goles, E., & Noual, M. (2012). Disjunctive networks and update schedules. *Adv. Appl. Math.*, *48*, 646–662.

Greil, F., Drossel, B., & Sattler, J. (2007). Critical Kauffman networks under deterministic asynchronous update. *New J. Phys.*, *9*, 373.

Irons, D. J. (2009). Logical analysis of the budding yeast cell cycle. *J. Theor. Biol.*, *257*, 543–559.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, *22*, 437–467.

Li, F., Long, T., Lu, Y., Ouyang, Q., & Tang, C. (2004). The yeast cell-cycle network is robustly designed. *Proc. Natl. Acad. Sci. USA*, *101*, 4781–4786.

Mangla, K., Dill, D. L., & Horowitz, M. A. (2010). Timing robustness in the budding and fission yeast cell cycles. *PLoS ONE*, *5*(2), e8906.

Mendoza, L., & Alvarez-Buylla, E. (1998). Dynamics of the genetic regulatory network for Arabidopsis thaliana flower morphogenesis. *J. Theor. Biol.*, *193*, 307–319.

Montalva, M. (2011). *Feedback set problems and dynamical behavior in regulatory networks*. PhD thesis Universidad de Concepción, Concepción, Chile.

Mortveit, H. S., & Reidys, C. (2004). Reduction of discrete dynamical systems over graphs. *Adv. Complex Syst.*, *7*, 1–20.

Novak, B., & Tayson, J. J. (2004). A model for restriction point control of the mammalian cell cycle. *J. Theor. Biol.*, *230*, 563–579.

Richard, A., Rossignol, G., Comet, J.-P., Bernot, G., Guespin-Michel, J., & Merieau, A. (2012). Boolean models of biosurfactants production in pseudomonas fluorescens. *PLoS ONE*, *7*(1), e24651. doi:10.1371/journal.pone.0024651.

Robert, F. (1986). *Discrete iterations*. Berlin: Springer.

Ruz, G. A., & Goles, E. (in press). Learning gene regulatory networks using the bees algorithm. *Neural Computing and Applications*.

Ruz, G. A., & Goles, E. (2012). Reconstruction and update robustness of the mammalian cell cycle network. In *Proceedings of IEEE symposium on computational intelligence in bioinformatics and computational biology (CIBCB 2012)*, San Diego, CA, USA, 9–12 May 2012 (pp. 397–403).

Serra, R., Villani, M., Barbieri, A., Kauffman, S. A., & Colacci, A. (2010). On the dynamics of random Boolean networks subject to noise: attractors, ergodic sets and cell types. *J. Theor. Biol.*, *265*, 185–193.