A Near-Optimal Control for Stochastic Gene Regulatory Networks

David Murrugarra

Department of Mathematics,
University of Kentucky.

Minisymposium on Discrete Methods in Molecular Biology,
SIAM Annual Meeting,
Pittsburgh, Pennsylvania, July 10-14, 2017.

UNIVERSITY OF KENTUCKY

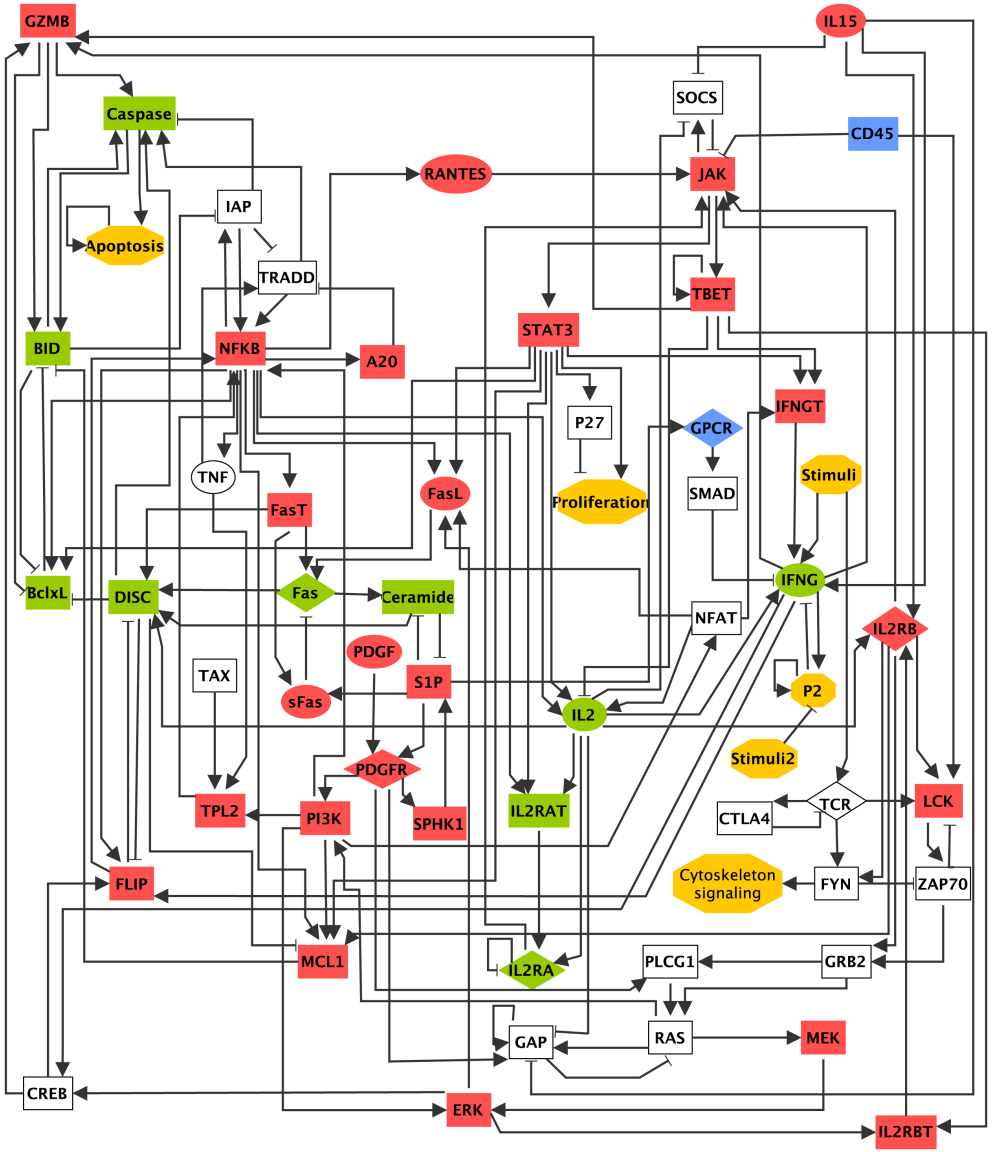


Figure: The T-LGL survival signaling network. This figure was taken from Saadatpour et al., *PLoS Comput Biol.*, 7(11), 2011.

The shape of the nodes indicates the cellular location:

- Rectangular indicates intracellular components,
- Ellipse indicates extracellular components,
- Diamond indicates receptors.

Node colors reflect the state of these nodes in leukemic cells:

- Red: Highly active components in T-LGL,
- Green: inhibited nodes,
- Blue: Nodes that have been suggested to be deregulated,
- White: the state of nodes is unknown.

The network also include

- Yellow: conceptual nodes (Stimuli, Stimuli2, P2, Cytoskeleton signaling, Proliferation, and Apoptosis).
- An arrowhead or a short perpendicular bar at the end of an edge indicates activation or inhibition, respectively.

Stochastic Discrete Dynamical Systems (SDDS)

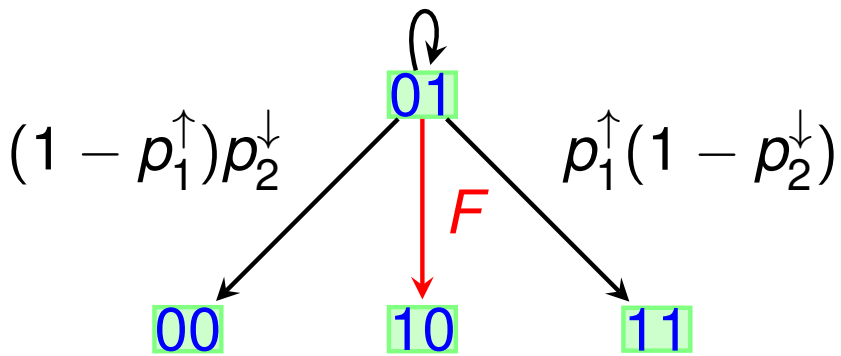
A **SDDS** on n variables $x_1, \ldots, x_n$ over a finite set X is a collection of n triplets

$$F = \{f_i, p_i^{\uparrow}, p_i^{\downarrow}\}_{i=1}^{n} \quad \text{where}$$

- $f_i : X^n \to X$ is the update function for x_i for all $i = 1, \ldots, n$.
- $p_i^{\uparrow}$ is the activation propensity.
- $p_i^{\downarrow}$ is the degradation propensity.
- $p_i^{\uparrow}, p_i^{\downarrow} \in [0, 1]$.

$$\theta^F_{k,x}(z) = \begin{cases} p_k^{\uparrow}\delta_z^{f_k} + (1 - p_k^{\uparrow})\delta_z^{x_k}, & \text{if } x_k < f_k(x), \\ p_k^{\downarrow}\delta_z^{f_k} + (1 - p_k^{\downarrow})\delta_z^{x_k}, & \text{if } x_k > f_k(x), \\ \delta_z^{x_k}, & \text{if } x_k = f_k(x). \end{cases}$$

Transition probability: $A_{x,y} = \prod_{k=1}^{n} \theta^F_{k,x}(y_k)$



Modeling Stochasticity and Variability in Gene Regulatory Networks. D. Murrugarra, A. Veliz-Cuba, B. Aguilar, S. Arat, R. Laubenbacher. *EURASIP Journal on Bioinformatics and Systems Biology*, 2012: 2012:5.

Example: WNT network in salamanders.

Genes

$x_1 = AREG$, $x_2 = PHLDA2$,
$x_3 = FGF9$, $x_4 = BMP2$,
$x_5 = NGFR$, $x_6 = HAPLN3$,
$x_7 = SP7$, $x_8 = Wnt\text{-}5a$,
$x_9 = Inhbb$, $x_{10} = DUSP$.

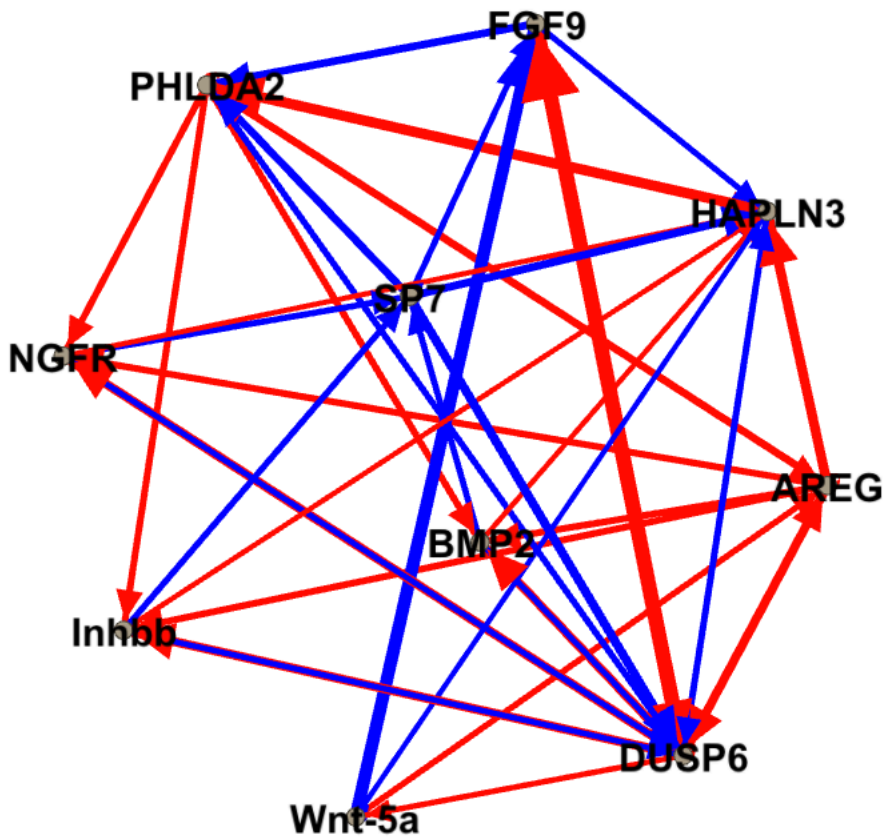
Update functions

$$f_1 = \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_8} \wedge \overline{x_{10}},$$
$$f_2 = \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_6},$$
$$f_3 = \overline{x_{10}} \vee (\overline{x_7} \wedge \overline{x_8}),$$
$$f_4 = \overline{x_{10}} \vee (\overline{x_1} \wedge \overline{x_2}),$$
$$f_5 = \overline{x_{10}} \vee (\overline{x_1} \wedge \overline{x_2}),$$
$$f_6 = (\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_4} \wedge \overline{x_5} \wedge \overline{x_9}) \vee (\overline{x_1} \wedge x_3 \wedge x_7 \wedge x_8),$$
$$f_7 = x_4 \wedge x_5 \wedge x_9,$$
$$f_8 = \overline{x_{10}} \vee x_8,$$
$$f_9 = \overline{x_{10}} \vee (\overline{x_1} \wedge \overline{x_2}),$$
$$f_{10} = (\overline{x_1} \wedge \overline{x_3} \wedge x_4 \wedge x_5 \wedge x_9 \wedge \overline{x_{10}}) \vee (\overline{x_1} \wedge x_2 \wedge x_4 \wedge x_5 \wedge$$

Propensities:

	x_1	$\cdots$	x_{10}
$p_i^{\uparrow}$	0.9	$\cdots$	0.9
$p_i^{\downarrow}$	0.9	$\cdots$	0.9

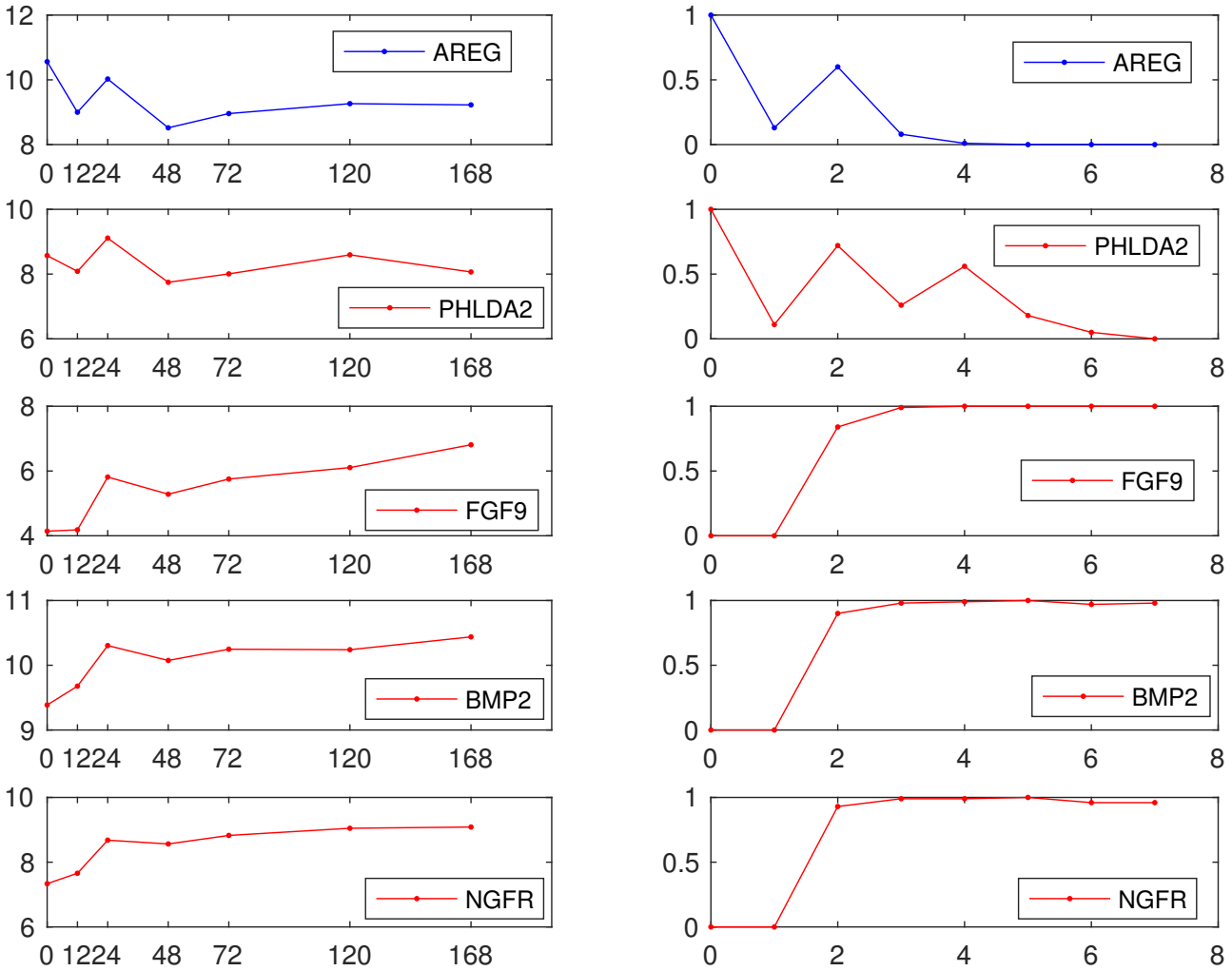
WNT Network



Wiring diagram for the genes listed on the left. Blue edges represent activation while red edges inhibition. Self loops were omitted.

Simulations with SDDS

Data vs simulations.



- The figures in the left panel are experimental data
- The figures in the righ panel are simulations.
- 100 runs initialized at 1100000001.

Markov Chain and Stationary Distribution for SDDS

$$\text{The transition from } x \text{ to } y: \quad a_{xy} = \prod_{i=1}^{n} Prob(x_i \to y_i).$$

Notice that $Prob(x_i \to y_i) = 0$ for all $y_i \notin \{x_i, f_i(x)\}$.

$$\text{Then the transition matrix is:} \quad A = (a_{xy})_{x,y \in S} \tag{1}$$

The transition probability $a_{xy} = p(X_t = x | X_{t-1} = y)$ represents the probability of being in state x at time t given that system was in state y at time $t - 1$.

$$\pi_1 = \sum_{x \in S} \pi_0(x) a_{xy}. \tag{2}$$

If we iterate Equation 2 and if we get to the point where

$$\pi = \sum_{x \in S} \pi(x) a_{xy} \tag{3}$$

We say that the Markov chain has reached a **stationary distribution**.

Example: Lac operon network

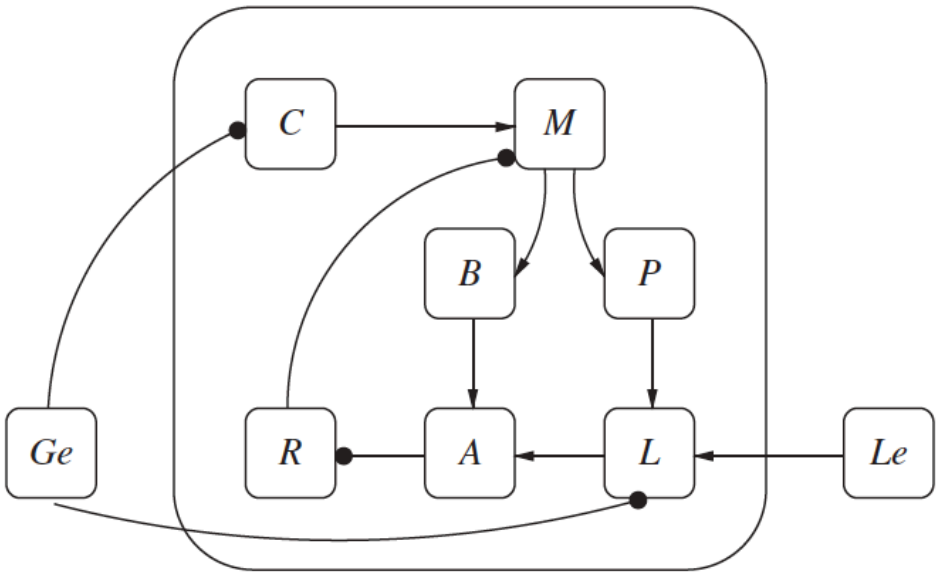
Genes

$x_1 = M$: *lac* mRNA,
$x_2 = P$: *lac permease*,
$x_3 = B$: *lac*β*-galactosidase*,
$x_4 = C$: CAP,
$x_5 = R$: repressor,
$x_6 = Rm$: repressor at medium
$x_7 = A$: allolactose,
$x_8 = Am$: allolactose at mediur
$x_9 = L$: lactose,
$x_{10} = Lm$: lactose at medium c

Update functions

$f_1 = x_4 \wedge \overline{x_5} \wedge \overline{x_6}$,
$f_2 = x_1$,
$f_3 = x_1$,
$f_4 = \overline{G_e}$,
$f_5 = \overline{x_7} \wedge \overline{x_8}$,
$f_6 = (\overline{x_7} \wedge \overline{x_8}) \vee x_5$,
$f_7 = x_9 \wedge x_3$,
$f_8 = x_9 \vee x_{10}$,
$f_9 = x_2 \wedge L_e \wedge \overline{G_e}$,
$f_{10} = ((L_{em} \wedge x_2) \vee L_e) \wedge \overline{G_e}$.

Lac operon Network

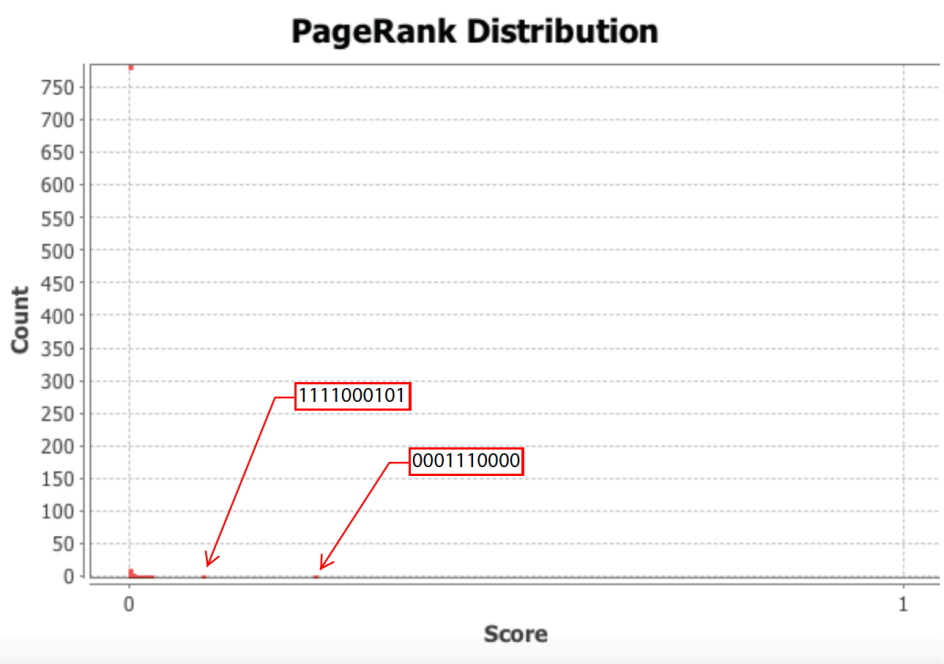


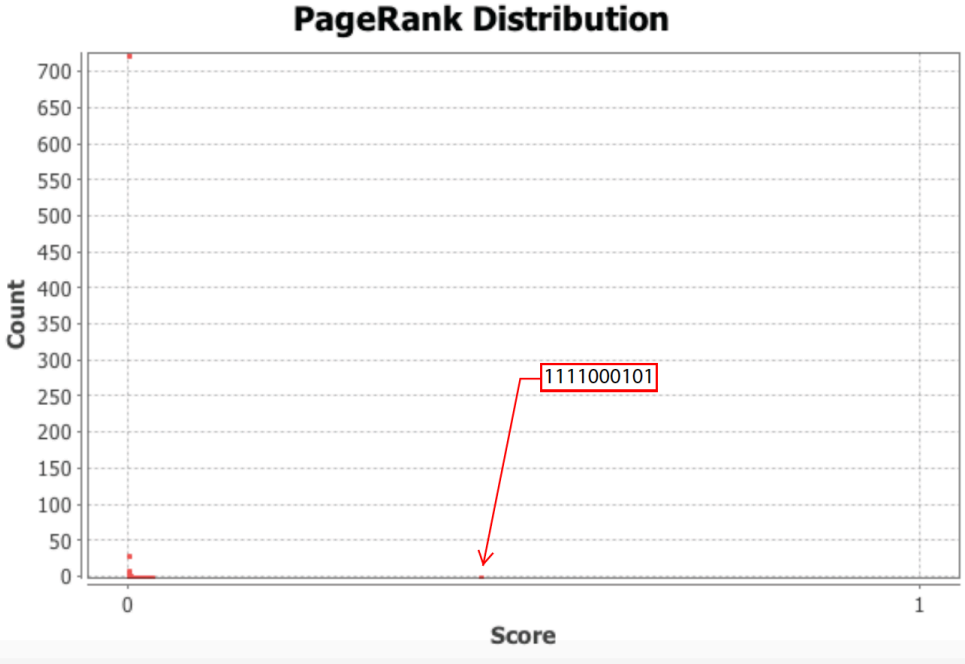
Wiring diagram for the genes listed on the left. Arrows represent activation while blunt arrows inhibition.

Propensities:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
$p_i^{\uparrow}$	0.81	1.00	0.97	0.62	0.11	0.63	0.22	0.82	0.48	0.60
$p_i^{\downarrow}$	0.17	0.59	0.03	0.98	0.39	1.00	0.33	0.07	0.52	0.06

Example: Lac operon network



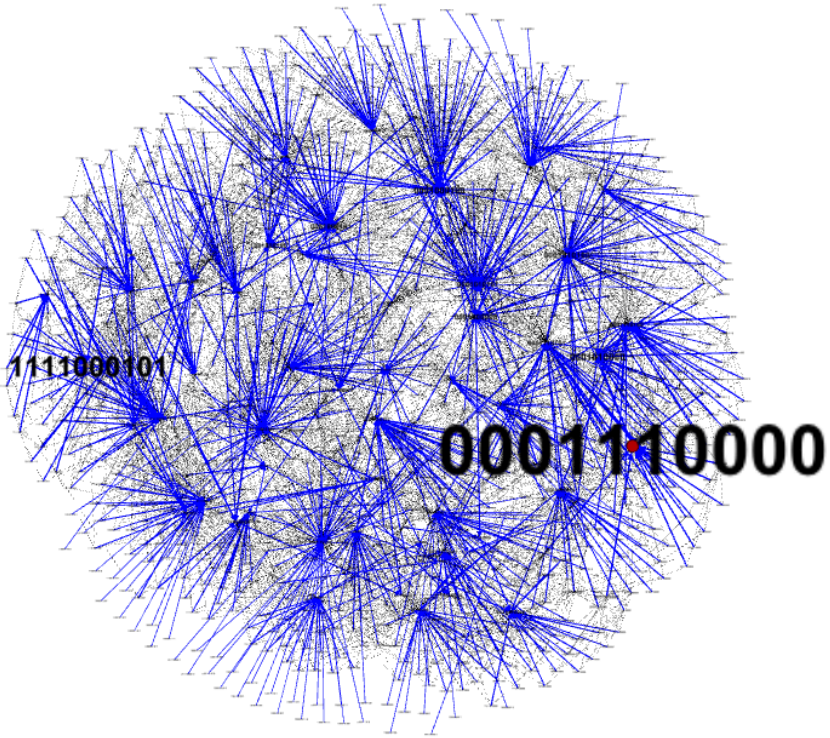


Scores with all 0.9 propensities.

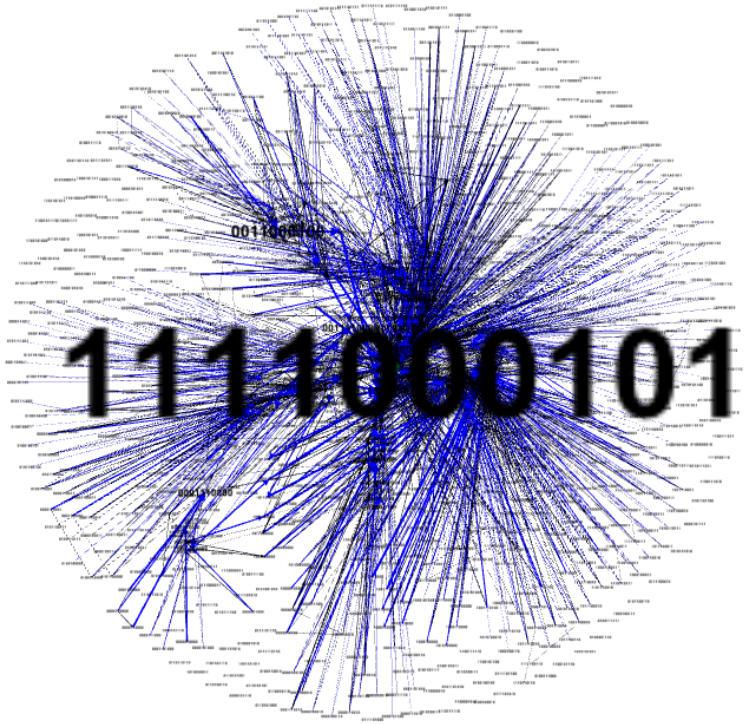
Scores with computed propensities.

PageRank scores before and after the genetic algorithm. The x-axes show the PageRank scores while the y-axes show the frequencies of states with the given scores in the x-axis. Left panel shows the state space where all the propensities are equal to 0.9 while the right panel shows the state space where the propensity parameters where estimated using the genetic algorithm.

Example: Lac operon network



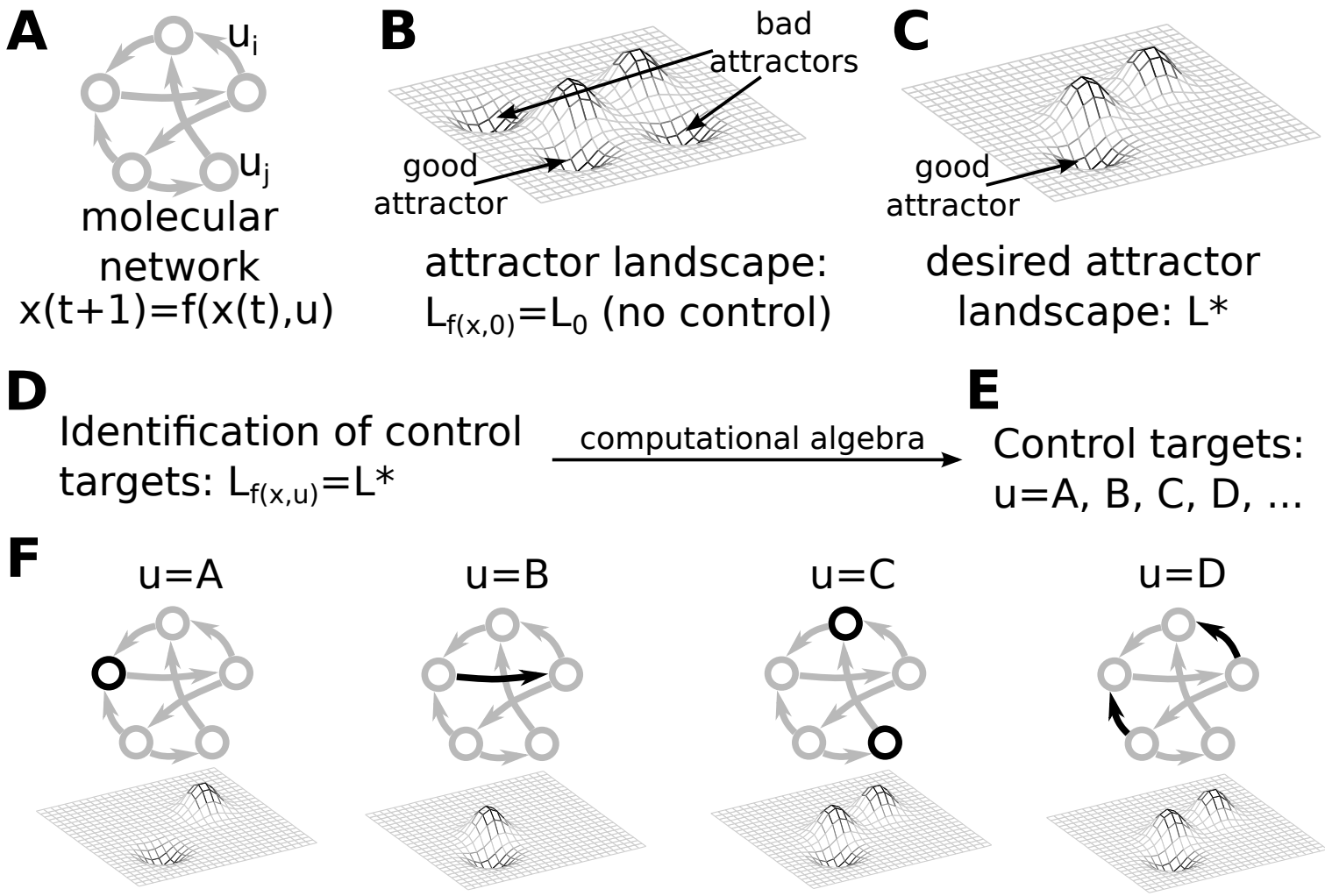
State space with all 0.9 propensities.



State space with computed propensities.

State space comparison before and after the genetic algorithm. Left panel shows the state space where all the propensities are equal to 0.9 while the right panel shows the state space with the estimated propensity parameters using the genetic algorithm. The edges in blue represent the most likely trajectory. The size of the labels of the nodes were scaled according to their PageRank score.

Network Control: for Boolean networks.



Identification of control targets in Boolean molecular network models via computational algebra.
D. Murrugarra, A. Veliz-Cuba, B. Aguilar, and R. Laubenbacher. BMC Systems Biology, 10:94, 2016.

Network Control: for Boolean networks.

Theorem (Effect of an edge deletion on the state space)

Let $\mathbf{F} = (f_1, \ldots, f_n) : \{0,1\}^n \to \{0,1\}^n$ *be a Boolean network where*

$$f_t(x_1, \ldots, x_n) = M_1(M_2(\ldots(M_{m-1}(M_m P_c + 1) + 1)\ldots) + 1) + b,$$

where $M_i = \prod_{i=1}^{\ell_i}(x_{i_j} + a_{i_j})$, P_c *is a polynomial with no canalizing variables, and* $d = \ell_1 + \ell_2 + \cdots + \ell_m$ *is the canalizing depth.*
The probability that any transition will be removed from the state space upon deletion of $x_k \to x_t$ *is at most*

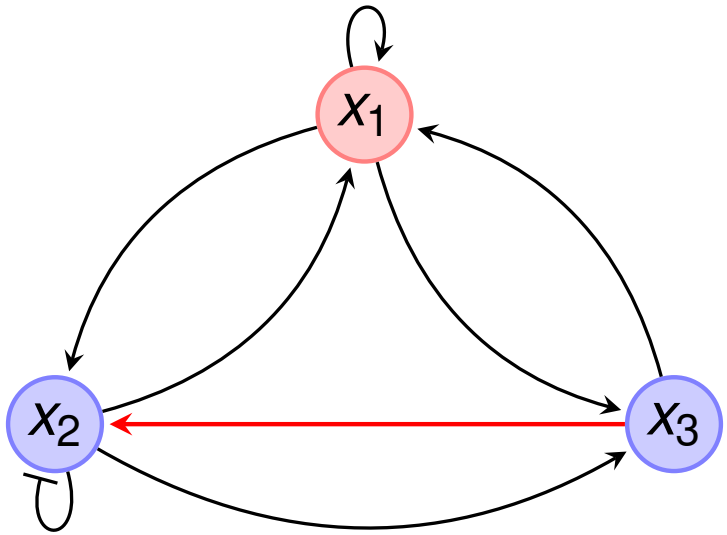
$$2^{n-\ell_1-\ell_2-\ldots-\ell_r}/2^n = (\frac{1}{2})^{\ell_1+\ell_2+\cdots+\ell_r}.$$

Molecular Network Control Through Boolean Canalization.
David Murrugarra and Elena Dimitrova.
EURASIP Journal on Bioinformatics and Systems Biology, 2015:9, 2015.

Definition of Control Targets:

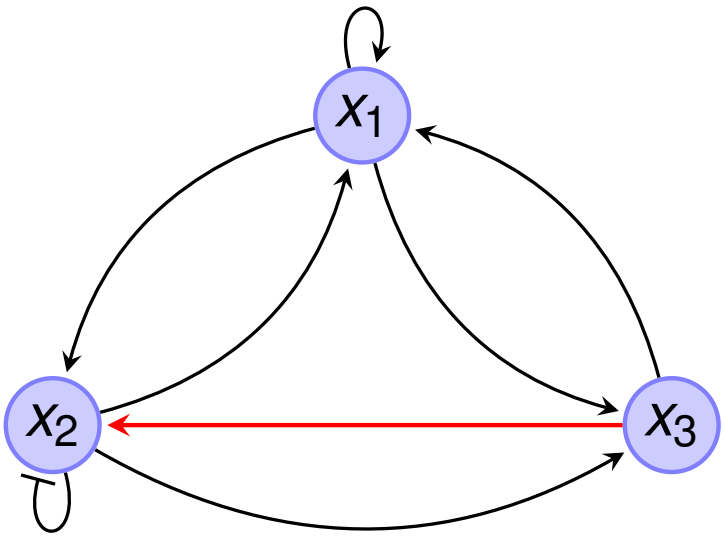
We consider two types of control actions:

1. Deletion or constant expression of edges
2. Deletion or constant expression of nodes.



Edge Deletion

Network



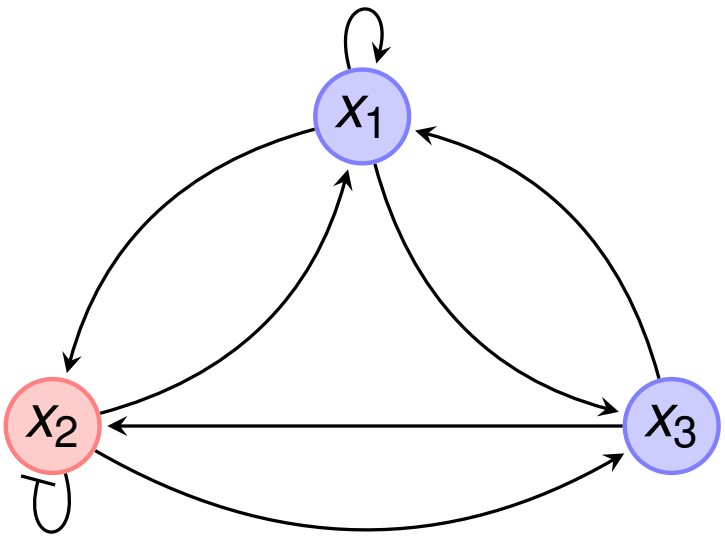
Controlled System

$$\mathcal{F}_2(\mathbf{x}, u_{3,2}) = f_2(x_1, x_2, (u_{3,2}+1)x_3)$$

- For $u_{3,2} = 0$, $\mathcal{F}_2(\mathbf{x}, 0) = f_2(x_1, x_2, x_3)$.
  The control is not active.
- For $u_{3,2} = 1$, $\mathcal{F}_2(\mathbf{x}, 1) = f_2(x_1, x_2, 0)$.
  The control is active and the action represents the deletion of the edge $x_3 \to x_2$.

Node Deletion

Network



Regulatory rule

$$\mathcal{F}_j(\mathbf{x}, u_i^-, u_i^+) := (u_i^- + u_i^+ + 1) f_j(\mathbf{x}) + u_i^+$$

- For $u_i^- = 0, u_i^+ = 0$, $\mathcal{F}_j(x, 0, 0) = f_j(x)$. The control is not active.
- For $u_i^- = 1, u_i^+ = 0$, $\mathcal{F}_j(x, 1, 0) = 0$. This action represents the knock out of the node x_j.
- For $u_i^- = 0, u_i^+ = 1$, $\mathcal{F}_j(x, 0, 1) = 1$. This action represents the constant expression of the node x_j.
- For $u_i^- = 1, u_i^+ = 1$, $\mathcal{F}_j(x, 1, 1) = f_j(x_{t_1}, \ldots, x_{t_m}) + 1$.

Control in the stochastic setting

Control actions. Consider

- E control edges and
- V control nodes.

We define a control action a as an array of binary elements of size $|U| = E + V$.

The set of all possible actions

$$A = \{(0, \ldots, 0), (0, \ldots, 1), \ldots, (1, \ldots, 1)\}$$

A has $|A| = 2^{|U|}$ elements.

Markov Decision Process for SDDS. A Markov decision process (MDP) for the set of states S and the set of actions A, consists of transition probabilities ($P^a_{x,y}$) and associated costs ($C(x, a, y)$), for each transition from state x to state y due to an applied action a.

Control in the stochastic setting

Transition Probabilities. The application of an action a results in a new SDDS,

$$F' = (\mathcal{F}_k(x, a), p_k^{\uparrow}, p_k^{\downarrow})_{k=1}^{n}.$$

Then, for each state action pair (x, a), $x \in S$, $a \in A$, the probability of transition to each state y upon execution of action a from state x, $P_{x,y}^{a}$, is computed with the f_k replaced by $\mathcal{F}_k$,

$$P_{x,y}^{a} = \prod_{k=1}^{n} \theta_{k,x^a}^{F'}(y_k).$$

Cost distribution. The cost of going from state x to state y under action a, $C(x, a, y)$, is a combination of two additive costs, one for actions C_a and one for states C_y.

$$C(x, a, y) = C_a + C_y \tag{4}$$

Cost distribution.

$$C(x, a, y) = C_a + C_y$$

The application of control edges or nodes have a penalty, c_e and c_v respectively, that represent expenses associated to the use of technologies and drugs required to silence nodes and edges.

$$\text{Thus, the cost of actions is: } C_a = c_v N_v + c_e N_e$$

where N_v and N_e are the number of applied control nodes and edges in a given action a.

The cost of ending up in a state y is the weighted distance between state y and a user specified desirable state s^*.

$$C_y = \sum_{k=1}^{N} w_k |y_k - s_k^*|$$

where w_k are user specified weights. If all the weights are 1, then C_y is simply the Hamming distance between y and s^*.

Control in the stochastic setting

A deterministic **control policy** π is defined as a set $\pi = \{\pi_0, \pi_1, \pi_2, \dots\}$, where

$$\pi_t : S \to A$$

is a mapping that associate a state $x(t)$ to an action a at time step t.

The Optimal Control Problem

The optimal control problem in this setting is to derive a control policy that dictates how to move from one state to another so that the probability of reaching a desirable attractor is maximized.

Control in the stochastic setting

Optimal Control Policies. We formulate the optimal control problem for infinite horizon MDPs with a discounting factor. Given a state $x \in S$, a control policy π, and a discounting factor $\gamma \in (0,1)$, the cost function V^π for π, is defined as:

$$V^\pi(x) = \mathrm{E}\left[\sum_{t=0}^{\infty} \gamma^t C(x(t), a) | s_0, \pi\right] \tag{5}$$

where $C(x(t), a)$ represents the expected cost at step t for executing the policy π from state x, $C(x(t), a) = \mathrm{E}_y[C(x, a, y)]$. The goal is to find the optimal policy $\pi^* = \{\pi_0^*, \pi_1^*, \dots\}$, where $\pi_t^* : S \to A$, $t = 1, 2, \dots,$ that minimizes the function cost for all states. The function cost associated with π^* is

$$V^*(x) = \min_\pi V^\pi(x) \text{ for all } x \in S.$$

Control in the stochastic setting

We also define the Q-function for π by

$$Q^{\pi}(x,a) = C(x(t),a) + \gamma \mathrm{E}_y[V^{\pi}(y)] \tag{6}$$

Similarly, for the optimal policy, $Q^*(x,a) = \min_{\pi} Q^{\pi}(x,a)$. It has been shown that the optimal cost function V^* satisfies the Bellman's principle:

$$V^*(x) = \min_{a \in A} [C(x,a) + \gamma \mathrm{E}_y[V^*(y)]] = \min_{a \in A} Q^*(x,a), \text{ for all } x \in S$$

The optimal policy for the MDP defined for SDDS is a stationary policy in which every state is associated with an action.

We can determine π^* with the help an iterative algorithm called *value iteration*.

Approximation Algorithm: near-optimality

We developed a randomized algorithm A for SDDS that takes as input any state s_0 and outputs an action.

The value function of the stochastic policy derived from A satisfies

$$|V^A(s_0) - V^*(s_0)| \leq \epsilon$$

where the value $\epsilon = \epsilon(c, h) > 0$ and

- c is the sample size and
- h is the number os iterations.

Michael J. Kearns and Yishay Mansour and Andrew Y. Ng.
A Sparse Sampling Algorithm for Near-Optimal Planning in Large Markov Decision Processes
Machine Learning, 49, 193-208, 2002.

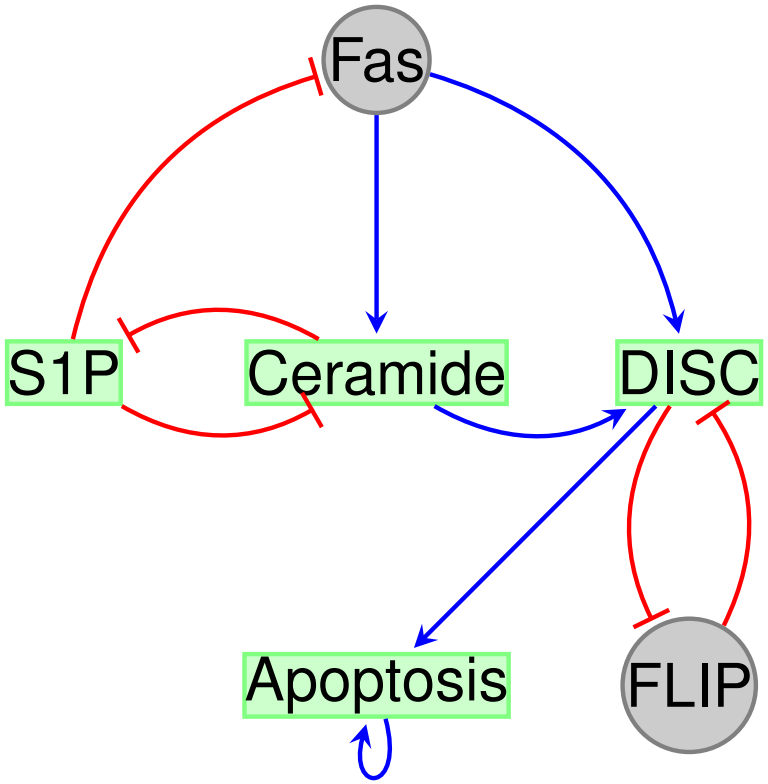
- Instead of computing a infinite horizon cost value function $V^\pi(s)$ under a policy π.
- The approximation creates a sub-MDP of finite horizon h by sampling the neighborhood of initial state s_0.
- The total expected cost function of the sub-MDP under a policy π is:

$$V_h^\pi(s_0) = \mathrm{E}\left[\sum_{t=0}^{h-1} \gamma^t C(x(t), a) | s_0, \pi\right] \tag{7}$$

The optimal cost over the sub-MDP is $V_h^*(s) = \min_\pi V_h^\pi(S)$.

The approximation algorithm computes an estimate $\hat{V}_h^*(s_0)$ of the optimal $V_h^*(s_0)$ by performing a sampling of the sub-MDP in the neighborhood of s_0.

Results



Reduced *T-LGL* network adapted from Saadatpour et al., *PLoS Comput Biol.*, 7(11), 2011.

Controllers: $\begin{cases} FLIP = OFF \\ Fas = ON. \end{cases}$

Control nodes (in gray) represent:

- the deletion of *FLIP* (*FLIP* = OFF or $x_2 = 0$) and
- the constant expression of *Fas* (*Fas* = ON or $x_3 = 1$).

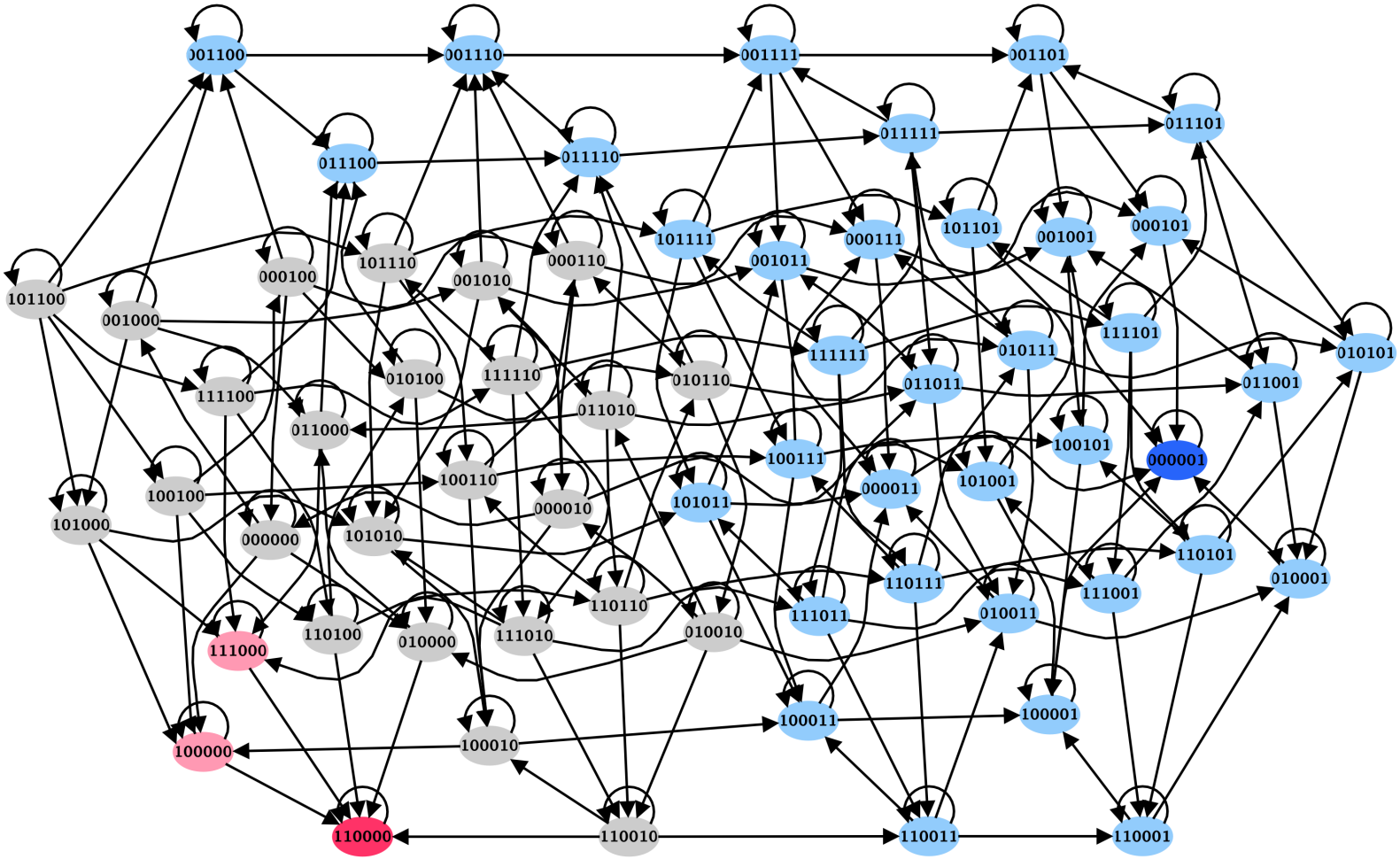
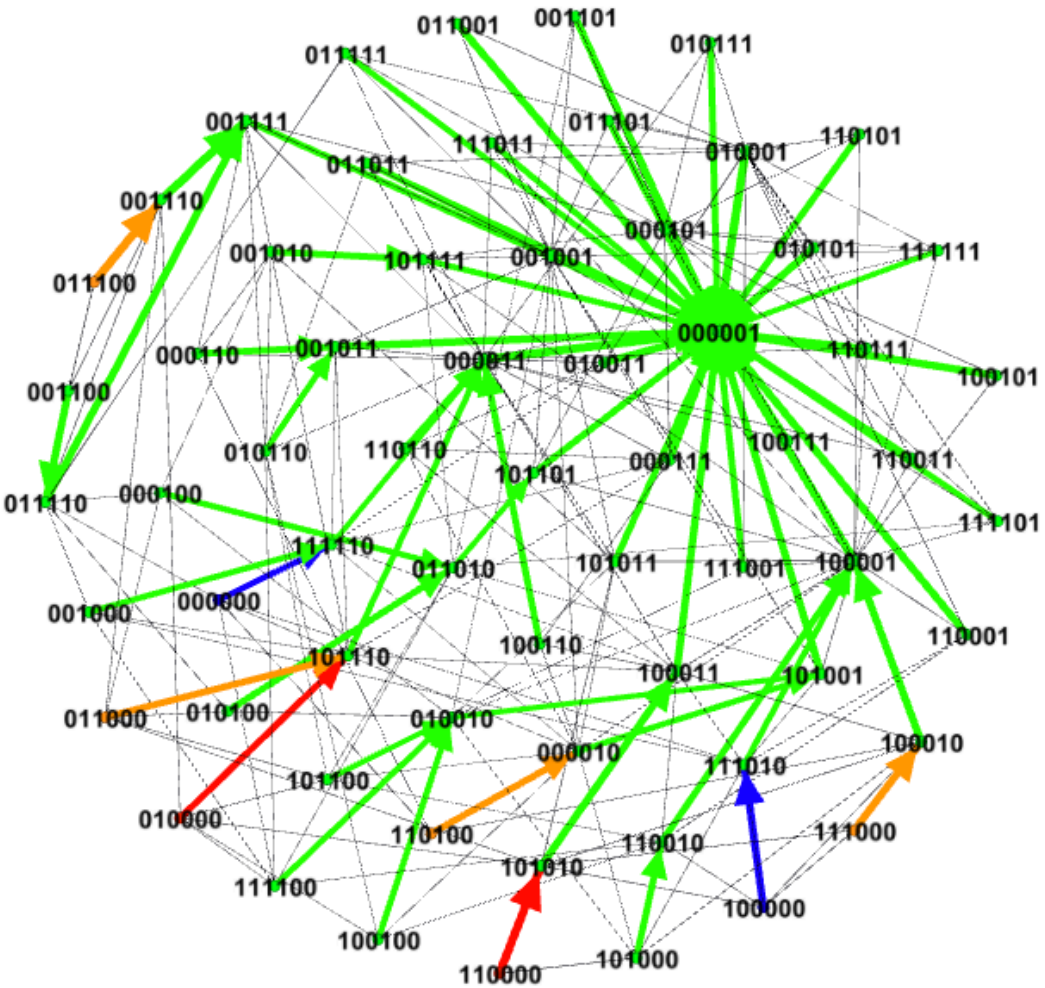


Figure: State space for reduced *T-LGL* network. Picture from Saadatpour et al., *PLoS Comput Biol.*, 7(11), 2011.

Results



Optimal control policy for the reduced *T-LGL* network.

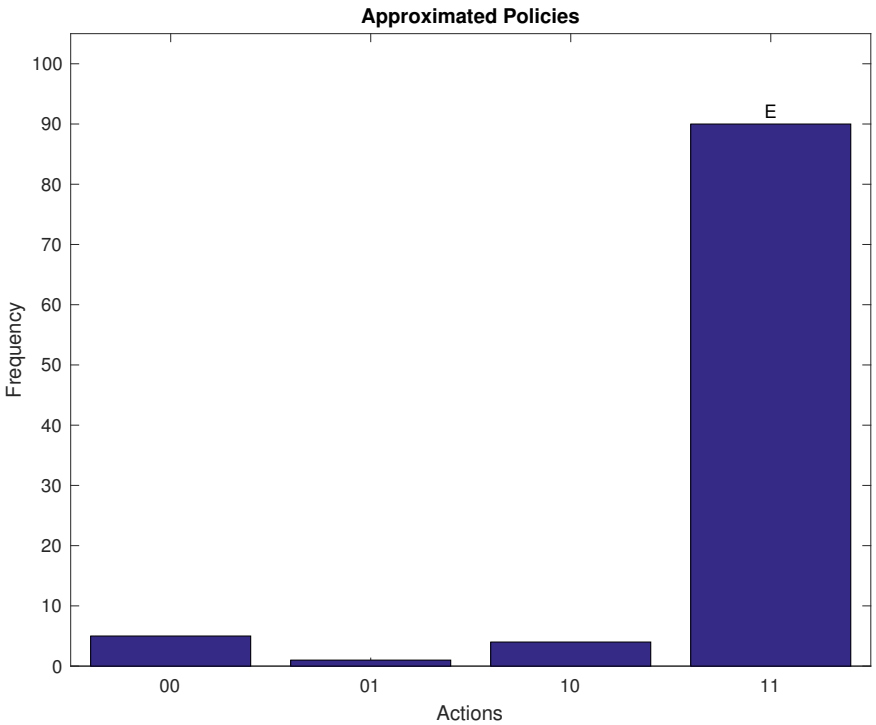
Controllers: $\begin{cases} FLIP = OFF \\ Fas = ON. \end{cases}$

- Arrows in green represent no control,
- Arrows in blue represent the control of the node *FLIP* ($x_2 = 0$),
- Arrows in orange represent the control of the node *Fas* ($x_3 = 1$), and
- Arrows in red represent the control of both nodes.

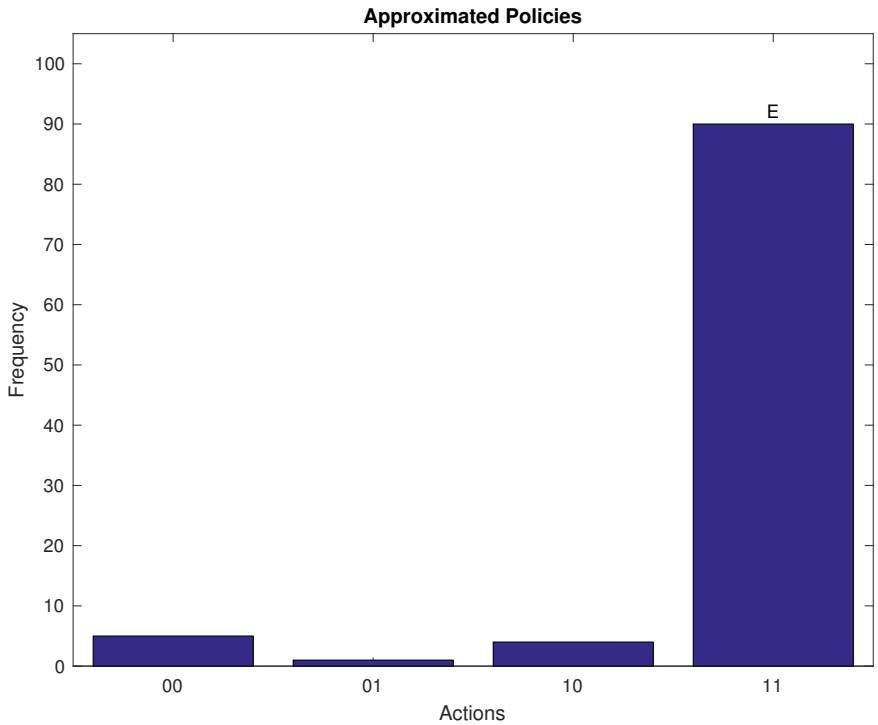
The colored thick arrows show the most likely transition while arrows in gray represent other possible transitions.

Results

Statistics for the 6 nodes *T-LGL* network for the state 110000.



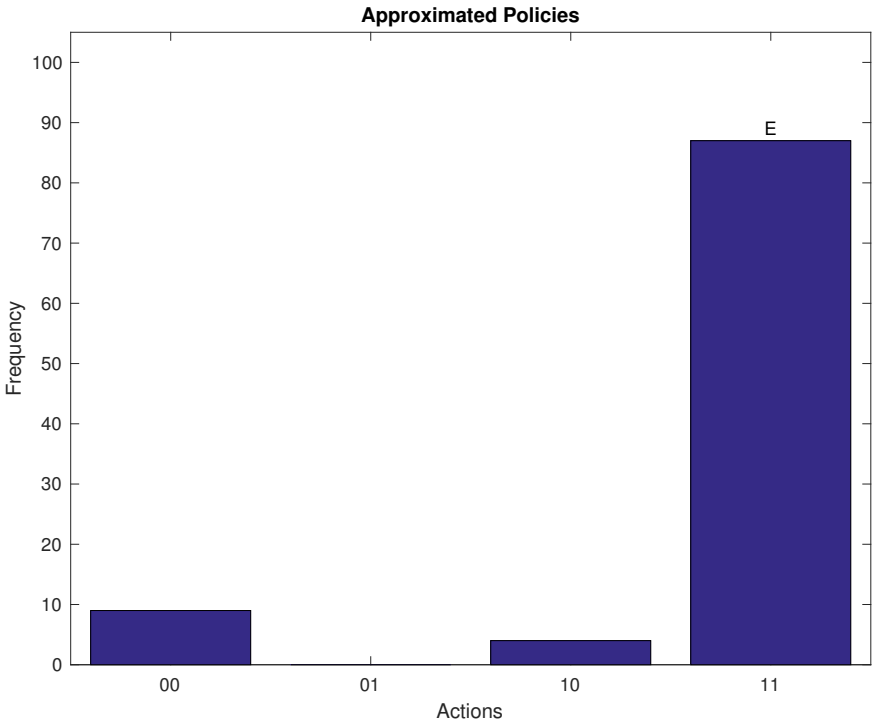
Approximations with $h = 2$ and $c = 6$.



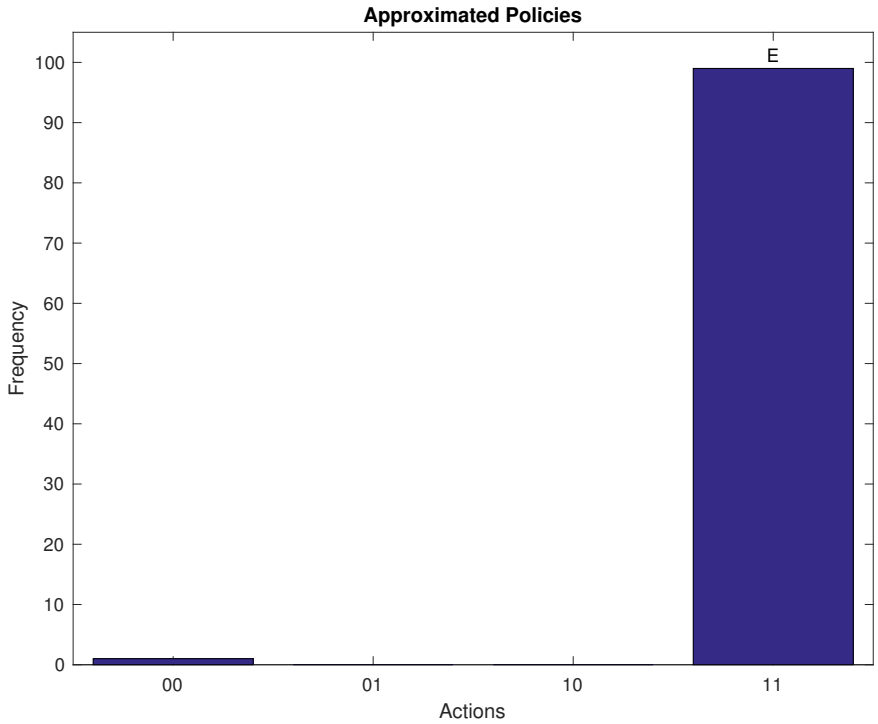
Approximations with $h = 3$ and $c = 6$.

Statistics for the 60 nodes *T-LGL* network for the state:

00111101110111111011001011010100111011010001101010111111 0100



Approximations with cyclic policies, $L = W = 2$, and $h = 3$ and $c = 6$.

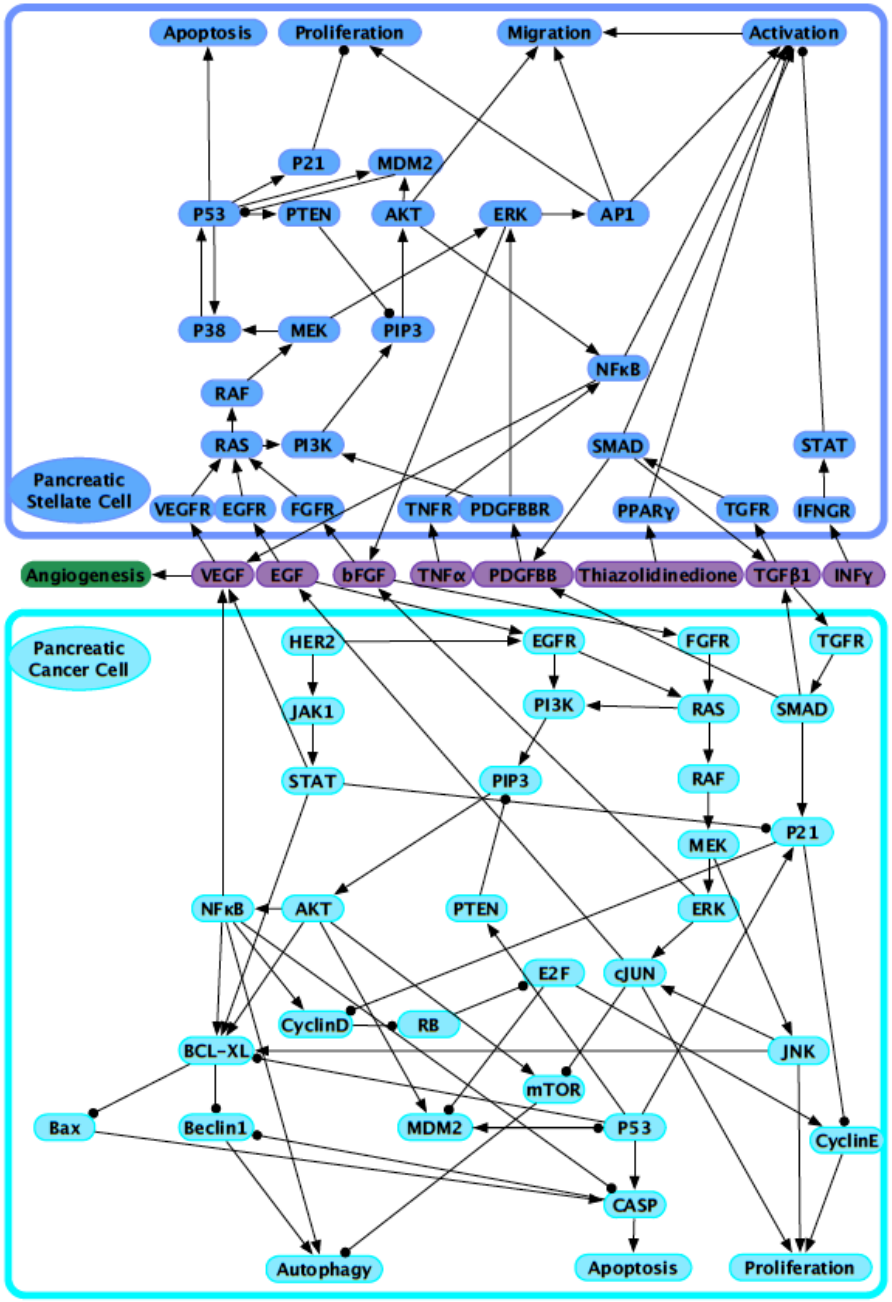


Approximations with cyclic policies, $L = W = 2$, and $h = 4$ and $c = 6$.

Results

States	Binary Expression
Cycle state	11111101110111111011001011010100111011010001101010111111010 0
Next state	10111101110111111011001011010100111011110100101010111111010 0
Next state	00111101110111111011001011010100111011110110101000111111010 0
Next state	01111101110111111011001011010100111011110110111001111111010 0
Fixed point	00000000000000000000000000000000000010000000000100001101000
Cycle state	10111101110111111011001011010100111011010001101010111111010 0
Next state	00111101110111111011001011010100111011110100101010111111010 0
Next state	01111101110111111011001011010100111011110110101000111111010 0
Next state	11111101110111111011001011010100111011110110111001111111010 0
Fixed point	00000000000000000000000000000000000010000000000100001101000
Cycle state	00111101110111111011001011010100111011010001101010111111010 0
Next state	01111101110111111011001011010100111011110100101010111111010 0
Next state	11111101110111111011001011010100111011110110101000111111010 0
Next state	10111101110111111011001011010100111011110110111001111111010 0
Fixed point	00000000000000000000000000000000000010000000000100001101000
Cycle state	00111101110111111011001011010100111011010001101010111111010 0
Next state	01111101110111111011001011010100111011110100101010111111010 0
Next state	11111101110111111011001011010100111011110110101000111111010 0
Next state	10111101110111111011001011010100111011110110111001111111010 0
Fixed point	00000000000000000000000000000000000010000000000100001101000

Table: Simulations from the states of the limit cycle for the 60 nodes *T-LGL* network.



Future Directions

- Built a SDDS for this network.
- Consider distributions of propensities.
- Apply the control methods for SDDS for this network.

Summary and conclusions:

- SDDS is a useful stochastic extension for discrete models.
- Google's PageRank algorithm combined with genetic algorithms can help to estimate the propensity parameters.
- SDDS can be used as a framework to study optimal control problems.
- Approximation techniques can be useful for the control of large networks.

Acknowledgments

Thank You!

Collaborators

- Boris Aguilar, Institute for Systems Biology.
- Randal Voss, University of Kentucky.
- Reinhard Laubenbacher, University of Connecticut Health Center.

Students:

- Pan Fang, University of Kentucky.
- Jacob Miller, University of Kentucky.
- Alex Mueller, University of Kentucky.

References:

- Identification of control targets in Boolean molecular network models via computational algebra.
  David Murrugarra, Alan Veliz-Cuba, Boris Aguilar, and Reinhard Laubenbacher.
  BMC Systems Biology, 10:94, 2016.
- Estimating Propensity Parameters using Google PageRank and Genetic Algorithms.
  David Murrugarra, Jacob Miller, and Alex Mueller.
  Frontiers in Neuroscience, 10:513, 2016.
- Molecular Network Control Through Boolean Canalization.
  David Murrugarra and Elena Dimitrova.
  EURASIP Journal on Bioinformatics and Systems Biology, 2015:9, 2015.
- Modeling Stochasticity and Variability in Gene Regulatory Networks.
  David Murrugarra, Alan Veliz-Cuba, Boris Aguilar, Seda Arat, Reinhard Laubenbacher.
  EURASIP Journal on Bioinformatics and Systems Biology, 2012:5, 2012.

email: murrugarra@uky.edu