# Evolving Boolean Networks for Biological Control: State Space Targeting in Scale Free Boolean Networks

# Evolving Boolean Networks for Biological Control: State Space Targeting in Scale Free Boolean Networks

Nadia S. Taou
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh, EH14 4AS, UK
Email: nt2@hw.ac.uk

David W. Corne
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh, EH14 4AS, UK
Email: d.w.corne@hw.ac.uk

Michael A. Lones
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh, EH14 4AS, UK
Email: m.lones@hw.ac.uk

***Abstract*—Gene regulatory networks are the complex dynamical structures that orchestrate the activities of biological cells. Inappropriate dynamical behaviours, caused by mutations or environmental perturbations, can lead to disease. Control interventions, for example in the form of therapeutic drugs, can lead to recovery from disease. In this paper, we consider how Boolean networks can be used to control a computational model of gene regulatory networks, focusing on the problem of state space targeting in scale-free Boolean networks, an abstract yet realistic model of biological gene regulatory networks. Our results suggest that Boolean networks can be optimised to carry out useful control, and that the approach is relatively scalable. We also take an initial look at the trade-off between the efficacy and efficiency of control, showing that many target networks can be controlled via a relatively small degree of coupling, giving hope that Boolean network controllers could one day be implemented in vivo.**

## I. Introduction

Controlling complex processes in the real world is an interesting and challenging problem [1]. Gene regulatory networks (GRNs) are complex dynamical systems found within all biological cells. They play a central role in living organisms, by orchestrating the activities of both the cell and the organism's wider developmental process. Efficient strategies for controlling GRNs could bring novel techniques to the design of therapeutic drug interventions for treating diseases [2]. However, in highly non-linear systems, like GRNs, it is hard to implement efficient control strategies.

GRNs can be modelled in many ways. In this paper, we consider Boolean networks (BNs), a simplified model of GRNs introduced by Kauffman [3]. Despite occasional criticism of their simplicity, BNs have been widely used to model and understand biological regulatory processes. This makes them a useful analogue for studying the control of biological GRNs. Previous examples of this have focussed on the use of conventional control theory. For example, Akutsu et al. [4] developed exponential time algorithms to control probabilistic BNs, having proved that the control of BNs in general is NP-complete. In [5], Chen and Qi developed an optimal control method for BNs using an algebraic approach. However, their method only worked for very restricted topologies.

In the evolutionary computing and artificial life communities, BNs have been considered from another perspective: using them as controllers [6]. A notable example is work by Roli et al. [7], where BNs were used to control robotic behaviours. More generally, computer scientists have considered the computational abilities of BNs, and used them to solve computational problems [6]. An example of this is Bull and Preen's work on evolving BNs to solve digital multiplexing problems in synchronous and asynchronous systems [8].

In earlier work [9], we explored a natural combination of these two ideas, looking at whether BNs can be evolved to control BNs. This is an appealing approach that, in effect, uses an analogue of a biological control system to control an analogue of a biological system. The relative ease of implementing Boolean logic in synthetic biology suggests that BNs may one day be refined into biological implementations [10]. Advances in drug delivery systems may even allow these biological control circuits to be deployed *in vivo*.

Our earlier work looked at the control of NK BNs (with N nodes and K connections per node) with fairly small values of N. In this work, we consider the control of scale-free BNs (SFBNs). Like many other naturally occurring networks, biological GRNs are scale free, exhibiting an exponential distribution of node connectivities. From a control perspective, this gives them significantly different characteristics to NK networks, notably their non-uniform response to perturbations. The controlled BNs are also significantly larger than those we considered previously. In addition, we take an initial look at using multiobjective evolutionary algorithms to explore trade-offs between the effectiveness of control and its ease of realisation, focusing on minimising the number of interventions required to exert control.

The paper is organised as follow: Section 2, 3 and 4 present a brief introduction to BNs, SFBNs and multi-objective optimisation. Section 5 describes our methodology. Section 6 presents results and analysis, and Section 7 concludes.

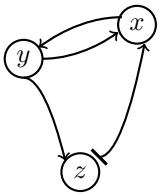



Fig. 1: Boolean network structure with three nodes.

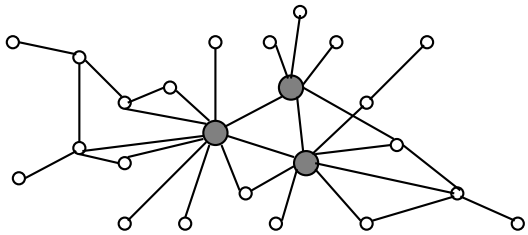


Fig. 2: Scale free Boolean network structure, showing three hubs in grey.

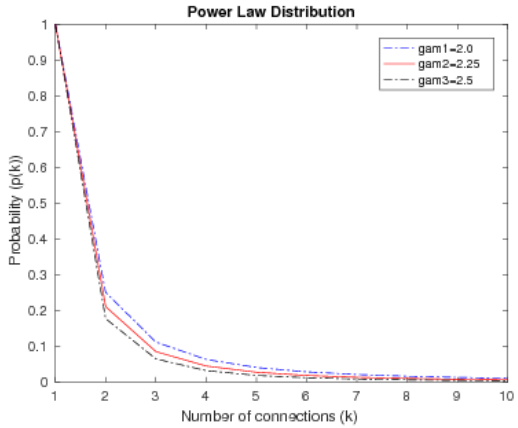



Fig. 3: Power Law Distribution ($k = [1-10]$, $\gamma_1 = 2.0$ $\gamma_2 = 2.25$, $\gamma_1 = 2.5$). The plot shows the distribution of connectivities in scale free networks for three different values of $\gamma$. A large number of nodes have only a few connections, and a small number of nodes (hubs) have a large number of connections.

## II. Boolean Networks

A Boolean network (BN) is a discrete-time non-linear dynamical system represented as a directed graph $G(V, E)$ composed of nodes, or vertices, $V$ and edges $E$ [11], [12], [13] (see Fig. 1). The time evolution of a BN is expressed by a set of Boolean functions $f_i$, $i = 1, 2, 3, ....$ Each BN node has a binary state $s$ which is updated synchronously according to its Boolean function and the states of the $k$ input nodes that are connected to it. Formally, $s(t+1) = f_i(s(t))$, where $s$ is a set of network states, $N$ is the number of nodes, $s \in \{0,1\}^N$, $t = 0, 1, 2, 3, 4, ...$ is the discrete time, and $f_i : \{0,1\}^N \rightarrow \{0,1\}$ (see Fig. 1). Since a BN is deterministic $s(t+1)$ is only determined by $s(t)$. The possible number of Boolean functions is $2^{2^k}$, and the state space is finite and equal to $2^N$. Since the state space is finite, states must eventually be repeated, leading to temporal structures called attractors. In particular, three regimes can be observed in BNs: *ordered, chaotic* and *critical*. Ordered BNs have attractors with a relatively short period, repeating the same series of states over and over again. Chaotic BNs have long periods; although deterministic, they appear random. Critical BNs also have attractors with long periods, but these appear to have a complex internal order.

## III. Scale Free Boolean Networks

A scale free Boolean network (SFBNs) is a dynamical system represented as a connected graph composed of a set of $N$ nodes $\{N_1, N_2, ..., N_N\}$ and connectivities ($k$), or degree, which exhibits a power law distribution $P(k) \sim k^\gamma$ [14], [15], [16], [17], [18] (see Fig. 2). $P(k)$ is the probability distribution that an arbitrary node of the network is connected to $n$ other nodes, and $\gamma$ is the scale free exponent, or scaling parameter. Scale free exponents often lie in the range $2 < \gamma < 3$; however, there are some exceptions. SFBNs can be constructed by adding nodes incrementally to an existing network (*growth mechanism*) and by creating new connections to existing nodes with a *preferential attachment mechanism* i.e. new nodes will prefer to connect to more connected nodes. The probability $p$ that a new node will be connected to a given node $i$ depends on the number of existing connections of $k_i$ that node $i$ has. The mathematical expression of this probability is: $p \sim \frac{k_i}{\sum_d k_d}$, where $k_i$ is the connectivity of node $i$ and $d$ is the index denoting the sum over network nodes. These two mechanisms explain the existence of *hubs*, which are nodes having connections with many other nodes in the network. Each node $N_i$ has a binary state, either 0 or 1, and is connected to $k_i$ other nodes of the network $\{N_{i1}, N_{i2}, ..., N_{ik_i}\}$ randomly chosen from a probability distribution $p_{inp}(k)$, $p_{inp}(k) = [(\sum_{n=1}^{\infty} k^{-\gamma})k^\gamma]^{-1}, \gamma > 1$. At each time step a Boolean function $F_i(N_{i1}, N_{i2}, ..., N_{ik_i})$ taken from a set of Boolean functions $\oint\{F_1, F_2, ..., F_N\}$ is assigned to $N_i$, such that for each state of $k_i$ other nodes, $F_i = 1$ with probability $p$ and $F_i = 0$ with probability $1-p$. Each node of the network is updated synchronously as follows: $N_i(t+1) = F_i(N_{i1}(t), N_{i2}(t), ..., N_{ik_i}(t))$ and the entire network $\chi(t)$ is updated at time $t$ with this dynamical equation: $\chi(t+1) = \oint(\chi(t))$, where $\chi(t) = \{N_1(t), N_2(t), ..., N_{(t)}\}$. SFBNs are more robust to external perturbations than Boolean networks. In [14], Aldana showed that for most real scale free networks $\gamma \in [2, 2.5]$.

## IV. Multi-Objective Optimisation

Real-world optimisation problems usually involve multiple conflicting objectives and highly complex search spaces, which prevents simultaneous optimisation of each objective. To address this, multi-objective optimisation techniques are needed. Evolutionary algorithms have various characteristics that make them useful for exploring multiple solutions at once, and consequently multi-objective evolutionary algorithms (MOEAs) are often used for such problems. Multi-objective optimisation

problems (MOOPs) have solutions which explore trade-offs in different ways. These are called Pareto optimal solutions (or non-dominated solutions), where none of the objectives in the search space can be improved without decreasing in value one or more other objectives. In general MOOP comprises a set of $n$ parameters known as decision variables, a set of $b$ objective functions, and finally a set of $m$ constraint functions. The set of feasible decision vectors is defined by the objective and constraint functions. MOOPs can be formulated in mathematical terms as follow:

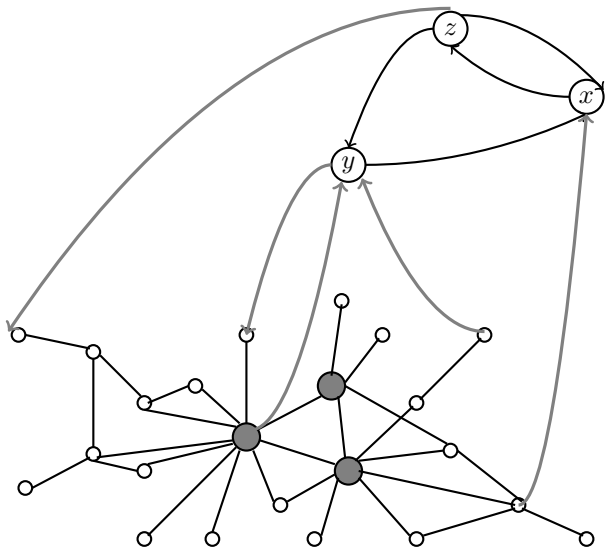
$$\begin{aligned} maximise \quad & h = f(z) = (f_1(z), f_2(z), \cdots, f_b(z)) \\ subject \quad to \quad & c(z) = (c_1(z), c_2(z), \cdots, c_m(z)) \leq 0 \\ where \quad & z = (z_1, z_2, \cdots, z_n) \in Z \\ & h = (h_1, h_2, \cdots, h_b) \in H \end{aligned}$$

$z$ is the decision vector, $h$ is the objective vector, $Z$ and $H$ are called respectively the decision space and the objective space. The main objective of a multi-objective optimisation algorithm is to find solutions in the Pareto optimal set and this requires to investigate solutions at the extreme ends of the objective function space [19].

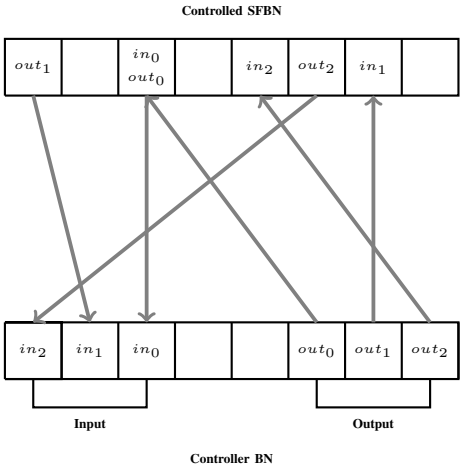
## V. Methods

We consider in this paper whether BNs can be optimised to control SFBNs. We focus on the task of state space targeting, i.e learning a control intervention that pushes a controlled SFBN to a particular point in its state space. This is analogous to the biological problem of controlling a GRN so that it moves to and remains within a particular region of its state space. For simplicity, the target state is all ones (i.e. every node has a Boolean state of one); however, this is no easier or harder to reach than any other arbitrary state for a particular sample of SFBNs, and is not similar to the max-ones problem in the genetic algorithms literature. In practice, it is likely that some SFBNs will be uncontrollable. It is also likely that the solution space will be hard to navigate for most SFBNs; for instance, a solution which guides the SFBN to a state of all-but-one nodes turned on is unlikely to be proximal to a solution which guides the SFBN to the optimal state. This means that, for many of the sampled SFBNs, it will not be feasible to reach the optimum. Rather, it is intended that the fitness distribution over a number of runs will give a general insight into the ability of the evolved Boolean networks to influence the dynamics of controlled networks, and a measure of the degree to which they are able to achieve this. At this stage, this is arguably more insightful than looking at their ability to control arbitrary Boolean networks derived from the biological literature, whose topologies and dynamics may not be representative of the wide class of networks.

To maximise the initial distance from the target, the state of all the nodes in the controlled SFBN are set to zero at the beginning of the control task. Scale free exponents in the range $\gamma \in [2, 2.5]$ are used. For each run of the evolutionary




(a) Coupled Boolean network and scale free Boolean network.




(b) Linear encoding used by the evolutionary algorithm.

Fig. 4: Boolean network coupled to a scale free Boolean network, also showing the linear encoding used by the evolutionary algorithm. Grey arrows indicate coupling between controller BN and controlled SFBN.

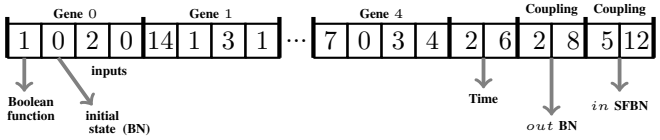



Fig. 5: Example of a Boolean network's genetic representation. Since $k = 2$, functions are numbered between 0 and 15. The timing and coupling terms indicate that this network is iterated twice each time it is executed, it is executed every 6 steps of the controlled network, its control outputs (interventions) are copied to nodes 2 and 8 of the controlled SFBN, and its feedback ($in$) inputs from the controlled SFBN are copied to nodes 5 and 12.

algorithm, the SFBN that is to be controlled is randomly sampled from the population of SFBNs of a particular size.

A controller is formed by a BN, a set of coupling terms, and two timing parameters in the range [1,50]. The BN is represented as an array of nodes, each comprising a Boolean function, an initial state, and a set of inputs, where each input is indicated by its position within the BN node array. It is thought that a BN's capacity for computation is maximal when it is in the critical regime [12]; consequently, a value of $k = 2$ is used, i.e. each node has exactly two inputs. The coupling terms indicate the nodes in the controlled SFBN whose state will be changed by the controller BN, i.e. the control interventions, and the nodes in the controlled SFBN whose state will be copied back to the controller BN, i.e. feedback connections see Fig. 4. Feedback from the controlled SFBN is always delivered by over-writing the states of nodes at the beginning of its node array. Control outputs are always read from the state of nodes at the end of the array. The first timing parameter determines how many steps the controller BN will execute for each step of the controlled SFBN, with values above 1 allowing the controller BN to execute faster than the controlled SFBN. The second parameter determines how often the controller BN is executed, in terms of the number of steps of the controlled SFBN. Larger values result in less frequent control interventions. Fig. 5 shows an example of how a controller is encoded.

In the first experiments, the controller's structure and parameters are optimised using a standard generational evolutionary algorithm, which is run for $100$ generations with a population size of $500$, uniform crossover ($p = 0.15$) and point mutation ($p = 0.06$). These values were found to be appropriate during initial experiments. A controller's fitness is a measure of the distance between the controlled SFBN's final state and the target state, after a control period of $100$ time steps of the controlled SFBN. The value is calculated by counting the number of 1s in the controlled SFBN's state at the final time step, and dividing by the size of the controlled SFBN, i.e. a value in the interval $[0, 1]$ where 1 indicates the correct all-ones state was reached. In the second experiment, NSGA-II [20] is used to explore trade-offs between control efficacy and efficiency. This is a well known and widely used MOEA. This approach uses two objectives: the first is the distance from the target, as described above; the second is the number of output couplings (or interventions) used to control the SFBN. Other parameters remain the same as for the standard evolutionary algorithm.

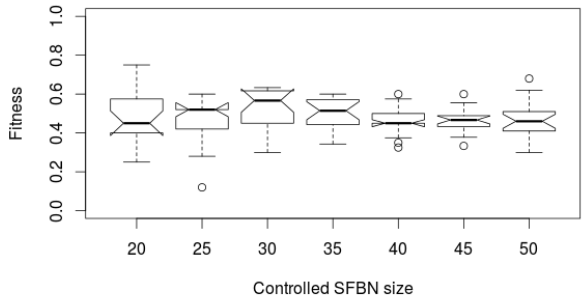
## VI. Results

Fig. 6a shows the fitness distributions for SFBNs in the absence of control, i.e. whilst following their natural dynamics. This shows that, without control, SFBNs tend towards a final state containing approximately equal numbers of 0s and 1s, indicated by a fitness around 0.5. By comparison, Figs. 6b–6d show the fitness distributions of controllers evolved to perform state space targeting in SFBNs. It is clear that fitness values are much higher when a controller is used, showing that BNs can be evolved to guide SFBNs towards particular parts of their state space. As expected, there were not many optimal solutions found; however, the fitness distributions are clearly separated from those of the uncontrolled networks. Fig. 6 also shows the effect of the controlled network size and the scale free exponent. Perhaps unsurprisingly, control becomes more difficult for larger SFBNs, since the state space grows rapidly. However, the scalability appears to be fairly good, with meaningful control still occurring for the largest SFBNs we looked at (which are larger than many of the models studied in the literature [2], [14] , [21]). To some extent, this may reflect the nature of scale free networks, with many of the nodes connected to, and therefore controllable by, a relatively small number of hubs. Nevertheless, the choice of scale free exponent had a relatively small effect on controllability, at least within the range we looked at. We found that the size of the controller BN (not shown) also had a relatively small effect beyond a certain point, reflecting our earlier results [9]. This may reflect a trade-off between increasing computational capacity in larger controllers versus a decreasing chance of finding controllers in a larger search space.

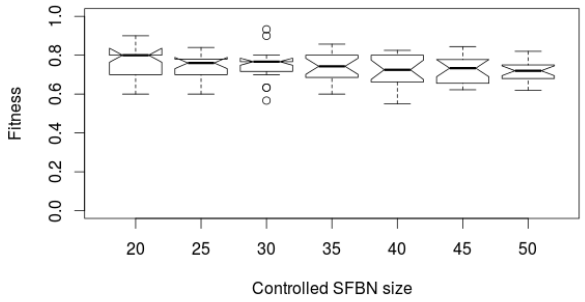
Fig. 7 shows Pareto fronts for 20 runs of NSGA-II for four different SFBN sizes, depicting the trade-off between the effectiveness of control and the number of interventions (i.e. output couplings) used to implement control. It is evident that there is a trade-off, with larger numbers of interventions generally leading to more effective control. This suggests that, if these networks were to be implemented *in vivo*, there will be a trade-off between the difficulty of implementation (more interventions are presumably harder to implement) and the effectiveness of control, though the extent of this trade-off will depend on the network being controlled. For the majority of the SFBN instances, there does not appear to be an advantage to having more than 2 or 3 interventions, and in many cases reasonable control can be enacted using only a single output coupling. Again, this may reflect the topology of scale-free networks. In future work, it would be interesting to see whether there is a correspondence between the nodes that are selected for control and their degree of connectivity, with the expectation that the EA is likely to identify the hubs of the network as control targets.

## VII. Conclusions

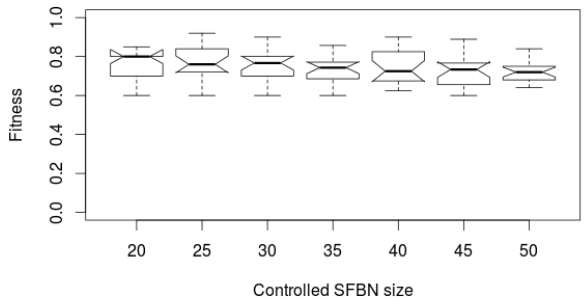
Effective control of gene regulatory networks is required in order to change the behaviour of biological cells. However, existing work in this area shows that this control problem is very difficult, and can only be solved analytically when network topologies are severely restricted. In this paper we have presented a new method for controlling the dynamics of scale-free Boolean networks (a computational model of gene regulatory networks) by coupling them to other Boolean networks optimised using evolutionary algorithms. Our results show that Boolean networks can be optimised to perform state space targeting in scale-free Boolean networks, and that the approach is relatively scalable. Using a multiobjective evolutionary algorithm, we also explored the trade-off between



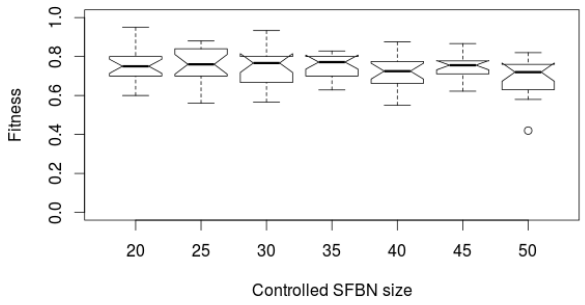

(a) No control, $\gamma_1 = 2.0$




(b) Control, $\gamma_1 = 2.0$




(c) Control, $\gamma_2 = 2.25$




(d) Control, $\gamma_3 = 2.5$

Fig. 6: Fitness distribution of SFBNs (a) following their natural dynamics, and (b–d) controlled with evolved BNs, $\gamma_1 = 2.0$, $\gamma_2 = 2.25$ and $\gamma_3 = 2.5$ and controller BN size $= 15$. High fitness values are better. Summary of 20 evolutionary runs are shown as notched box plots; overlapping notches indicate when median values (thick horizontal bars) are not significantly different at the $95\%$ confidence level.

maximising control efficacy and minimising the number of control interventions, noting that many SFBNs could be controlled with relatively few interventions. Given the relative ease with which Boolean models can be implemented using synthetic biology techniques, our results suggest this could be a useful means of generating biological controllers, potentially providing a new route to therapeutic drug discovery. In future work, we intend to look at how this approach can be used to control actual biological networks.

## REFERENCES


[1] M. Lones, A. Turner, L. Caves, S. Stepney, S. Smith and A. Tyrrell, *Artificial biochemical networks: Evolving dynamical systems to control dynamical systems*, Evolutionary Computation, IEEE Transactions on, 18 (2), 145–166, 2014.

[2] J. Kim, S. Park and K. Cho, *Discovery of a kernel for controlling biomolecular regulatory networks*, Scientific reports, 3, 2013.

[3] S. Kauffman, *Metabolic stability and epigenesis in randomly constructed genetic nets*, Journal of theoretical biology, 3, 22, 437–467, 1969.

[4] T. Akutsu, M. Hayashida, W. Ching and M. Ng, *Control of Boolean networks: hardness results and algorithms for tree structured networks*, Journal of Theoretical Biology, 244 (4), 670–679, 2007.

[5] D. Cheng and H. Qi, *Controllability and observability of Boolean control networks*, Automatica, 45 (7), 1659–1667, 2009.

[6] M. Lones, *Computing with artificial gene regulatory networks*, Evolutionary Computation in Gene Regulatory Network Research, 398–424, 2016.

[7] A. Roli, M. Manfroni, C. Pinciroli and M. Birattari, *On the design of Boolean network robots*, Applications of Evolutionary Computation, Proceedings of the 2011 International Conference on Applications of Evolutionary Computation - Volume Part I, EvoApplications'11, Springer, Berlin Heidelberg, 43–52, 2011.

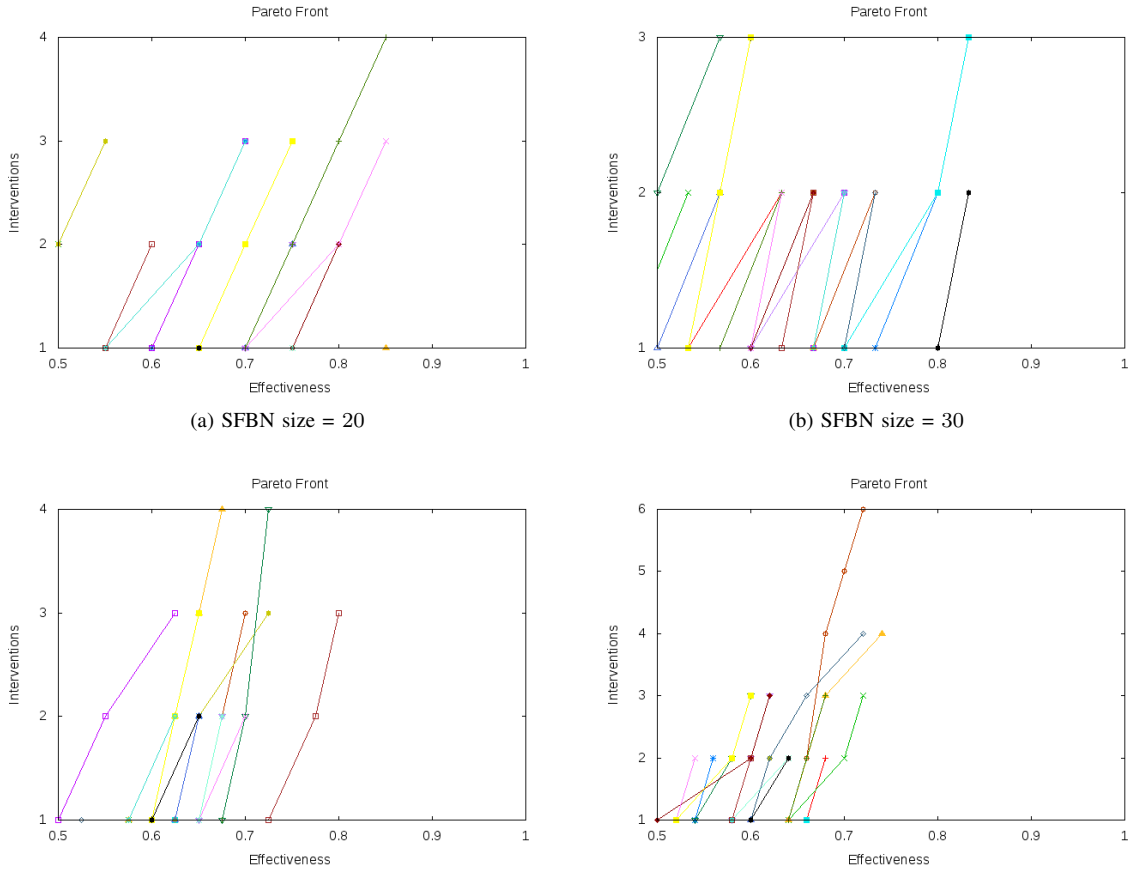
[8] L. Bull and R. Preen, *On dynamical genetic programming: Random boolean networks in learning classifier systems*, Genetic Programming, Proceedings of the 12th EuropeIn Bedau, M., P. Husbands, T. Hutton, S. Kumar, and H. Suzuki (eds.) Workshop and Tutorial Proceedings, Ninth International Conference on the Simulation and Synthesis of Living Systemsan Conference on Genetic Programming, EuroGP 2009, Leonardo Vanneschi, Steven Gustafson, Alberto Moraglio, Ivanoe De Falco and Marc Ebner, LNCS vol. 5481, Springer, 37–48, 2009.

[9] N. Taou, D. Corne and M. Lones, *Towards Intelligent Biological Control: Controlling Boolean Networks with Boolean Networks*, Applications of Evolutionary Computation, LNCS vol. 9597, Springer, 351–362, 2016.

[10] O. Purcell and T. Lu, *Synthetic analog and digital circuits for cellular computation and memory*, Current opinion in biotechnology, 29, 146–155, 2014.

[11] S. Kauffman, *The origins of order: Self organization and selection in evolution*, Oxford university press, 1993.

[12] C. Gershenson, *Introduction to random Boolean networks*, Artificial Life IX Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALife IX), 160-173, 2004.




(a) SFBN size = 20

(b) SFBN size = 30

(c) SFBN size = 40

(d) SFBN size = 50

Fig. 7: Pareto fronts, showing the trade-off between control efficacy and the number of interventions, $\gamma = 2.25$, $k = 2$. The different coloured lines indicate the non-dominated solutions from 20 different runs, each with a different controlled SFBN.


[13] B. Drossel, *Random boolean networks*, Reviews of nonlinear dynamics and complexity, 1, 69–110, 2008.

[14] M. Aldana, *Boolean dynamics of networks with scale-free topology*, Physica D: Nonlinear Phenomena, 185 (1), 45–66, 2003.

[15] A. Barabási and E. Bonabeau, *Scale-free networks*, Scientific American, 288 (5), 50–59, 2003.

[16] A. Clauset, C. Shalizi, M. Newman, *Power-law distributions in empirical data*, SIAM review, 51 (4), 661–703, 2009.

[17] R. Cohen, S. Havlin and D. Ben-Avraham, *Structural properties of scale free networks*, Handbook of graphs and networks, 2003.

[18] R. Serra, M. Villani and L. Agostini, *On the dynamics of scale-free Boolean networks*, Neural Nets, 43–49, 2003.

[19] A. Konak, D. Coit and A. Smith, *Multi-objective optimization using genetic algorithms: A tutorial*, Reliability Engineering & System Safety, 91 (9), 992–1007, 2006.

[20] K. Deb, *Multi-objective optimization using evolutionary algorithms*, 16, 2001.

[21] Y. Zhao, Park. Sang-Min and C. Kwang-Hyun *Control of large-scale boolean networks via network aggregation*, 2015.