

Research Article

Interpolative Boolean Networks

**Vladimir Dobrić, Pavle Milošević, Aleksandar Rakićević,
Bratislav Petrović, and Ana Poledica**

Faculty of Organizational Sciences, University of Belgrade, Jove Ilića 154, 11000 Belgrade, Serbia

Correspondence should be addressed to Vladimir Dobrić; vdobric@gmail.com

Received 8 June 2017; Accepted 21 August 2017; Published 29 October 2017

Academic Editor: Valeri Mladenov

Copyright © 2017 Vladimir Dobrić et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Boolean networks are used for modeling and analysis of complex systems of interacting entities. Classical Boolean networks are binary and they are relevant for modeling systems with complex switch-like causal interactions. More descriptive power can be provided by the introduction of gradation in this model. If this is accomplished by using conventional fuzzy logics, the generalized model cannot secure the Boolean frame. Consequently, the validity of the model's dynamics is not secured. The aim of this paper is to present the Boolean consistent generalization of Boolean networks, interpolative Boolean networks. The generalization is based on interpolative Boolean algebra, the $[0, 1]$ -valued realization of Boolean algebra. The proposed model is adaptive with respect to the nature of input variables and it offers greater descriptive power as compared with traditional models. For illustrative purposes, IBN is compared to the models based on existing real-valued approaches. Due to the complexity of the most systems to be analyzed and the characteristics of interpolative Boolean algebra, the software support is developed to provide graphical and numerical tools for complex system modeling and analysis.

1. Introduction

Approaches based on logic are used for modeling systems with complex causal interactions between the components [1–3]. Obtained models usually have the form of directed graphs, with a logical state variable and a logical function assigned to each node. The state of a node evolves in continuous or discrete time, in accordance with the value realization of the logical function. In comparison to traditional models of dynamical systems, logical models are less detailed and requires less information about the process of interest. Determination of an appropriate level of detail is considered as the major challenge in modeling. If the model is very detailed, it might be extremely difficult to gain insight; thus, the essential nature of the system may be obscured. On the other hand, if too little detail is included, the model may not be a reliable representation of reality. The appropriate level of detail is determined with respect to the purpose of the model and the amount of information about the process of interest. The logical modeling is aimed at capturing the nonlinear dynamics of possibly large-scale systems. The modeling of such systems is typically hampered by the lack of information. According to the principle of incompatibility

[4], as the complexity of a system increases our ability to obtain both relevant and precise model of the system diminishes, and, eventually, these two model characteristics become incompatible. In the logical approaches, the precision suffers for the relevance. This enables modeling of systems that are too complex for traditional approaches.

Boolean networks (BNs) are the common logical model for modeling complex systems based on binary logic. They may be seen as discrete dynamical systems where each node logical function works dynamically to compute the node state at the next time step depending on the input values coming from other nodes [5, 6]. Therefore, the research in this field mainly addresses the structural and dynamical aspect of BN. The BN structure can be specified by an expert [7], identified from the data [8] or randomly constructed [5, 9]. On the other hand, the validity of the dynamics is secured by the laws of Boolean algebra (BA). According to the principle of truth functionality, the values of logical functions in such models are calculated by the values of their arguments in parallel or serial manner [10]. Classical models are widely used in biology and medicine [11–14], geosciences [15–18], and social sciences [19–22].

BNs are relevant for modeling systems which exhibit threshold behavior. If more details are available, BNs can be generalized by the introduction of gradation. In this case, logical variables and functions are treated as $[0, 1]$ -valued. In [9, 23–28], BNs are generalized using conventional, truth functional fuzzy logics. Since such fuzzy logics are not a BA [29, 30], the generalized models cannot secure the Boolean frame. Consequently, the validity of the dynamics of the model cannot be secured.

In this paper, we proposed the Boolean consistent $[0, 1]$ -valued generalization of BN. This approach, called the interpolative Boolean networks (IBN), is based on interpolative Boolean algebra (IBA), the $[0, 1]$ -valued realization of BA [31]. Unlike conventional fuzzy logics which are based on the truth functionality principle, IBA is based on the structural functionality principle to secure Boolean consistent generalization. According to this principle, logical functions have vector nature, and thus, the Boolean consistent calculations of values can be accomplished by the immanent structure vectors. This enables IBA, and consequently the IBN, to be used for modeling and analysis of complex dynamical systems in a Boolean consistent manner. Further, IBN is the general $[0, 1]$ -valued approach, while Boolecube, Hillcube, and similar approaches may be seen as the special case of IBN. The validity of the dynamics of IBN is illustrated on two examples from the literature using a software support developed for this purpose.

The paper is organized as follows. In Section 2, we describe the classical binary BNs and the existing real-valued realizations of BNs. Section 3 provides the basic concepts of interpolative Boolean algebra. In Section 4, the proposed generalization of BN is introduced. Furthermore, the software for complex system modeling and analysis, implementing IBN, is presented. In Section 5, the proposed approach is illustrated on two examples. The obtained results are compared with both conventional fuzzy approach and the Boolecube/Hillcube approach. The main conclusions and directions for future work are given in Section 6.

2. Boolean Networks

Boolean networks are the least detailed mathematical models of causally interacting entities. The model has the form of a directed graph, whose nodes represent the entities, and the edges represent the interactions between the entities:

$$\text{BN} = (N, E), \quad (1)$$

where $N = \{x_1, \dots, x_m\}$ is a set of nodes and E is a set of direct edges [5].

2.1. Binary Boolean Networks. Dynamics of BN is introduced by assigning a Boolean state variable and a Boolean function to each node. It is assumed that all nodes change state at each time step, with a unit time-delay. The dynamics of a BN reflects the topology of the directed graph. The state of node x_i takes the value of its Boolean function whose arguments

are the states of the node's predecessors in the previous time step:

$$x_i(t+1) = f_i(x_{i_1}(t), \dots, x_{i_{K_i}}(t)), \quad (2)$$

where $x_i(t+1) \in \{0, 1\}$ and $f_i : \{0, 1\}^{K_i} \rightarrow \{0, 1\}$ are the state and the Boolean function of node x_i , $x_{i_1}(t), \dots, x_{i_{K_i}}(t) \in \{0, 1\}$ are the states of node x_i predecessors in the previous time step, and K_i is the number of predecessors of the node x_i .

Dynamics of BN is defined by equations which update the state of all nodes in a system:

$$\begin{bmatrix} x_1(t+1) \\ \vdots \\ x_m(t+1) \end{bmatrix} = \begin{bmatrix} f_1(x_{1_1}(t), \dots, x_{1_{K_1}}(t)) \\ \vdots \\ f_m(x_{m_1}(t), \dots, x_{m_{K_m}}(t)) \end{bmatrix}, \quad (3)$$

where $x(t+1) = [x_i(t+1) \mid i = 1, \dots, m]^T$, $x(t+1) \in \{0, 1\}^m$, is the state of a Boolean network at time $t+1$, and m is the number of nodes.

Transition functions $f_i(\cdot)$ in a classic BN, as any binary logical function, follow the principle of truth functionality [32, 33]: the value of a complex formula is uniquely determined by the truth values of its subformulas [34]. In other words, the value of a logical function is calculated using the values of functions' arguments. Since the algebra of binary BNs is Boolean, the validity of the BN dynamics is secured. The global dynamics is characterized by the existence of one or more point or cycle attractors. This is a state or a set of states where the dynamic of the network stabilizes. For each attractor there is a basin of attraction which contains states that converge to the attractor. The dynamics of a BN depends on which basin of attraction the initial states are located. The number and size of attractors, and the corresponding basins of attraction, are the state space properties which reveal the global dynamics of BNs.

Classical Boolean networks are adequate for modeling systems with complex switch-like causal interactions. In such systems, cause and effect relationships between the components are governed by thresholds [35]. Below (or above) a certain level, a component has a little or no influence on the behavior of others, while above (or below) this level the influence saturates rapidly to a constant level. Such a switch-like causal relationships can be represented using sigmoid functions. Binary representation in the classical BNs is an idealization of such relationships.

2.2. Real-Valued Realization of Boolean Networks. When more detailed description of a system is required and/or possible, the Boolean networks can be generalized by the introduction of gradation. In this case, logical state variables and logical functions are realized as real-valued. This provides the drastic increases of the descriptive power of the model [36–38]. The behavior of a system can be quantified with much more precision, and thus, deeper insight into the system's dynamics is possible. In [9, 23–28], Gödel, product, and Lukasiewicz conventional fuzzy logics are proposed for

generalization of BNs. However, the validity of the dynamics of the proposed generalized models cannot be secured, since the Boolean laws of excluded middle and contradiction are not followed in fuzzy logic [29]. This implies that the behavior of complex systems may not be accurately predicted by such models.

The Boolecube/Hillcube model is able to predict and explain quantitative experimental data [3]. This approach is the generalization of BNs based on multilinear interpolation of Boolean functions. Boolean functions are replaced by their real-valued realizations called the Boolecubes or the Hillcubes. The Boolecube is defined by the following expression:

$$\begin{aligned} & \bar{f}(\bar{x}_1, \dots, \bar{x}_m) \\ &= \sum_{x_1=0}^1 \sum_{x_2=0}^1 \dots \sum_{x_m=0}^1 \left(f(x_1, \dots, x_m) \prod_{i=1}^m (x_i \bar{x}_i + (1-x_i)(1-\bar{x}_i)) \right), \end{aligned} \quad (4)$$

$$\bar{f}_h(\bar{x}_1, \dots, \bar{x}_m) = \sum_{x_1=0}^1 \sum_{x_2=0}^1 \dots \sum_{x_m=0}^1 \left(f(x_1, \dots, x_m) \prod_{i=1}^m (x_i h(\bar{x}_i) + (1-x_i)(1-h(\bar{x}_i))) \right), \quad (6)$$

where $x_i(t) \in \{0, 1\}$, $\bar{x}_i \in [0, 1]$, $f(x_1, \dots, x_m) \in \{0, 1\}$ is a Boolean function, and $h(\bar{x}_i) \in [0, 1]$ is the Hill function.

3. Interpolative Boolean Algebra

Interpolative Boolean algebra (IBA) is the Boolean consistent $[0, 1]$ -valued generalization of finite (atomic) Boolean algebra (BA) [30, 39]. Formally, IBA is based on the principle of structural functionality [39–41]. It implies the necessity to distinctly detach the structure of the logical function from its value realization. Therefore, IBA is two-leveled algebra consisting of symbolic and valued level. The structural transformation on a symbolic level enables us to evaluate any logical function in the Boolean consistent way. Once the transformations have been conducted, the value level is introduced. The values do not have an influence on the preservation of Boolean laws. Therefore, IBA is used as a natural frame for Boolean consistent fuzzy logic [42], logical aggregation [40], similarity measure [43], and computational cognition [44].

3.1. The Principle of Structural Functionality. Unlike the truth functional fuzzy logic and many-valued algebras, IBA is based on the principle of structural functionality [39–41]. This principle implies that the immanent structure of any logical function can be directly derived from the structures of the function's arguments. In other words, any element of finite Boolean algebra uniquely corresponds to its structure (content, relation of inclusion). Relying on the principle of structural functionality is essentially different from relying on the principle of truth functionality—the structure of the observed function is at the forefront rather than the values of its arguments [45].

The main motive of introducing the principle of structural functionality is that the principle of truth functionality is binary in its essence, that is, the truth value of an expression

where $x_i(t) \in \{0, 1\}$, $\bar{x}_i \in [0, 1]$, and $f(x_1, \dots, x_m) \in \{0, 1\}$ is a Boolean function that indicates the inclusion of the variables/functions in the Boolecube.

The Hillcube approach is the generalization of the Boolecube approach since it uses the sigmoid function for mapping variables. This function is called the Hill function and it is defined by the following expression:

$$h(\bar{x}_i) = \frac{(\bar{x}_i)^{n_i}}{(\bar{x}_i)^{n_i} + (k_i)^{n_i}}, \quad (5)$$

where n_i is the Hill coefficient which defines the steepness of the function and k_i is the threshold parameter. Therefore, the Hillcube is defined by the following expression:

may be true or false. Therefore, the truth functionality principle is valid in the sense that it preserves all Boolean axioms and theorems for a two-valued realization of Boolean algebra, while it is not valid in many-valued case and fuzzy logic [29]. On the other hand, according to the principle of structure functionality, the structure of the logical expression and the intensity of variables should both be taken into account in $[0, 1]$ -valued case. The structure of an element determines which atomic elements of IBA are included in it and/or which are not included in it. An atomic element of IBA is the simplest elements of the Boolean algebra domain of attributes BA in the sense that they do not include in themselves anything except for a trivial Boolean constant 0. Atomic element expresses the intensities of variable within the observed structure. Detailed mathematical formalization of IBA structural functionality is given further in the text.

In [40], it is shown that the truth functional principle is a value realization of the structure functional principle which is valid (in the sense that it preserves all Boolean axioms and theorems) only for a two-valued realization of Boolean algebra. On the other hand, the structure functional principle is valid in general case [40]. Therefore, the principle of truth functionality is a special case of the principle of structural functionality from the aspect of the Boolean laws.

3.2. IBA Symbolic Level. From the mathematical point of view, any logical function on IBA symbolic level is represented in a vector form, structure vector. The value of any logical function can be defined as a scalar product of two vectors:

$$f^\otimes(x_1, \dots, x_m) = \vec{\sigma}_f \vec{\alpha}, \quad (7)$$

where $\vec{\sigma}_f$ is the structure vector of $f(x_1, \dots, x_m)$ and $\vec{\alpha}$ is atomic vector. The structure vector is binary and it represents

the shape/structure of the logical function, while the atomic vector expresses the intensities of variable. In classical binary case, the intensity of variables is $\{0, 1\}$, so it is clearly a special case of IBA approach.

The structure vector is a binary vector defined by the following expression [40]:

$$\vec{\sigma}_f = [\sigma_f(\alpha_S) \mid S \in P(\Omega)]^T, \quad (8)$$

where σ_f are the structure function of logical function $f(x_1, \dots, x_m)$, α_S is an atom of $BA(\Omega) = P(P(\Omega))$, and $P(\Omega)$ is the power set of a set of free variables $\Omega = \{x_1, \dots, x_m\}$.

The structure vector $\vec{\sigma}_f$ contains information about which atoms are relevant/included in a logical function. Such information is given by the structural function σ_f , which is defined by the following expression [40]:

$$\sigma_f(\alpha_S) = \begin{cases} 1, & \alpha_S(x_1, \dots, x_m) \wedge f(x_1, \dots, x_m) = \alpha_S(x_1, \dots, x_m) \\ 0, & \alpha_S(x_1, \dots, x_m) \wedge f(x_1, \dots, x_m) = \underline{0}. \end{cases} \quad (9)$$

An atom in IBA is defined by the following expression [40]:

$$\alpha_S(x_1, \dots, x_m) = \bigwedge_{x_i \in S} x_i \bigwedge_{x_j \in \Omega \setminus S} \neg x_j, \quad (10)$$

where $S \in P(\Omega)$. For instance, atomic elements of BA generated by two attributes $\Omega = \{a, b\}$ are $a \wedge b$, $a \wedge \neg b$, $\neg a \wedge b$, and $\neg a \wedge \neg b$.

In accordance with the structural functionality principle, any logical function can be uniquely represented by the following disjunctive canonical form [39]:

$$f(x_1, \dots, x_m) = \bigcup_{\sigma_f(\alpha_S)=1 \mid S \in P(\Omega)} \alpha_S(x_1, \dots, x_m). \quad (11)$$

In order to obtain the real value realization of any logical function, IBA uses generalized Boolean polynomials (GBPs). GBP is a sum of the relevant atomic Boolean polynomials:

$$\begin{aligned} f^\otimes(x_1, \dots, x_m) &= \sum_{\sigma_f(\alpha_S)=1 \mid S \in P(\Omega)} \alpha_S^\otimes(x_1, \dots, x_m) \\ &= \sum_{\sigma_f(\alpha_S)=1 \mid S \in P(\Omega)} \left(\bigotimes_{x_i \in S} x_i \bigotimes_{x_j \in \Omega \setminus S} (1 - x_j) \right), \end{aligned} \quad (12)$$

where $x_i \in [0, 1]$ and \otimes is a generalized product.

3.3. IBA Structural Transformation and Valued Level. IBA is technically based on the mapping procedure of logical

function to GBP. Any Boolean function can be mapped into corresponding GBP using the following rules [40]:

$$\begin{aligned} (f_i(x_1, \dots, x_m) \wedge f_j(x_1, \dots, x_m))^\otimes &= f_i^\otimes(x_1, \dots, x_m) \otimes f_j^\otimes(x_1, \dots, x_m) \\ (f_i(x_1, \dots, x_m) \vee f_j(x_1, \dots, x_m))^\otimes &= f_i^\otimes(x_1, \dots, x_m) + f_j^\otimes(x_1, \dots, x_m) \\ &\quad - f_i^\otimes(x_1, \dots, x_m) \otimes f_j^\otimes(x_1, \dots, x_m) \\ (\neg f_i(x_1, \dots, x_m))^\otimes &= 1 - f_i^\otimes(x_1, \dots, x_m) \\ (x_i \wedge x_j)^\otimes &= \begin{cases} x_i \otimes x_j, & i \neq j \\ x_i, & i = j \end{cases} \\ (x_i \vee x_j)^\otimes &= x_i + x_j - x_i \otimes x_j \\ (\neg x_i)^\otimes &= 1 - x_i. \end{aligned} \quad (13)$$

The generalized product \otimes is any function that maps $\otimes : [0, 1] \times [0, 1] \rightarrow [0, 1]$ and satisfies all four axioms of t -norms (commutativity, associativity, monotonicity, and boundary condition) and the condition which ensures that the values of atomic Boolean polynomials are nonnegative [39, 40]. Therefore, the generalized product \otimes for $\Omega = \{x_1, x_2\}$ may be realized as any t -norm that produce the value from the following interval:

$$\max(x_1 + x_2 - 1, 0) \leq x_1 \otimes x_2 \leq \min(x_1, x_2). \quad (14)$$

Three cases may be distinguished as particularly important realizations of the generalized product: minimum function, ordinary product, and Lukasiewicz operator.

4. Interpolative Boolean Networks

In this section, we present the Boolean consistent $[0, 1]$ -valued generalization of Boolean networks, interpolative Boolean network. In the proposed model, states of nodes take value from the unit interval, while the Boolean functions are mapped into generalized Boolean polynomials according to (13). Such polynomials are the $[0, 1]$ -valued realizations of Boolean functions with respect to the structure functionality principle. The proposed generalization provides a drastic increase of the descriptive power of BNs. It allows the model to be used for gaining quantitative insight into the behavior of complex systems. For the purpose of this paper, we assumed that interpolative Boolean networks evolve in discrete time, although continuous time models can be generalized in the same way. Formally, in the IBN, the state of node x_i takes the value of its generalized Boolean polynomial whose arguments

are the states of node x_i predecessors in the previous time step:

$$\begin{aligned} x_i(t+1) &= f_i^{\otimes}(x_{i_1}(t), \dots, x_{i_{k_i}}(t)) \\ &= \sum_{\sigma_{f_i}(\alpha_S)=1 | S \in P(N)} \left(\bigotimes_{x_p \in S} x_p(t) \bigotimes_{x_q \in N \setminus S} (1 - x_q(t)) \right), \end{aligned} \quad (15)$$

where $x_i(t+1) \in [0, 1]$ and $f_i(x_{i_1}(t), \dots, x_{i_{k_i}}(t)) \in \text{BA}(N)$ are the state and the Boolean function of node $x_i \in N$, $x_{i_1}(t), \dots, x_{i_{k_i}}(t) \in [0, 1]$ are the states of node x_i predecessors in the previous time step, σ_{f_i} is the structure function of $f_i \in \text{BA}(N)$, $x_i \otimes x_j : [0, 1] \times [0, 1] \rightarrow [0, 1]$ is the generalized product, and K_i is the number of predecessors of the node x_i .

The state of node x_i can be defined as the scalar product of two vectors:

$$x_i(t+1) = \vec{\sigma}_{f_i} \vec{\alpha}(t), \quad (16)$$

where $\vec{\sigma}_{f_i}$ is the structure vector of $f_i \in \text{BA}(N)$ and $\vec{\alpha}(t)$ is a vector of atomic Boolean polynomials of $\text{BA}(N)$ at time t .

Dynamics of interpolative Boolean network is defined by the following equations:

$$\begin{aligned} \begin{bmatrix} x_1(t+1) \\ \vdots \\ x_m(t+1) \end{bmatrix} &= \begin{bmatrix} f_1^{\otimes}(x_{1_1}(t), \dots, x_{1_{k_1}}(t)) \\ \vdots \\ f_m^{\otimes}(x_{m_1}(t), \dots, x_{m_{k_m}}(t)) \end{bmatrix} \\ &= \begin{bmatrix} \vec{\sigma}_{f_1} \vec{\alpha}(t) \\ \vdots \\ \vec{\sigma}_{f_m} \vec{\alpha}(t) \end{bmatrix}, \end{aligned} \quad (17)$$

where $x(t+1) = [x_i(t+1) | i = 1, \dots, m]^T$, $x(t+1) \in [0, 1]^m$, is the state of IBN at time $t+1$ and m is the number of nodes.

The interpolative Boolean network evolves according to the principle of structure functionality. This secures the Boolean frame and the logical validity of the model's dynamics. The global dynamics is characterized by the existence of point, cycle, and/or chaotic attractors, with the corresponding basins of attraction. The behavior of IBN can be fundamentally different depending on which basin of attraction the initial states are located. If the initial state of each node is two-valued, IBN reduces to the classical model; that is, the structure functionality principle reduces to the truth functionality principle. IBN can be used for modeling complex systems which may or may not exhibit the threshold behavior.

The behavior of IBN depends on the way how generalized product operators are treated. The adequate generalized product operator for the two variables is determined with respect to the nature of the variables. Two logical variables $x_i(t) \in [0, 1]$ and $x_j(t) \in [0, 1]$ are of the same nature if they represent the same quality. If this is the case, the adequate generalized product operator is minimum function;

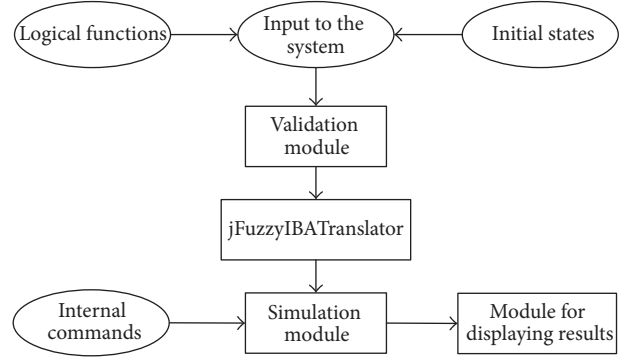


FIGURE 1: Software support tool architecture.

that is, $x_i(t) \otimes x_j(t) = \min(x_i(t), x_j(t))$. If two variables are of the same nature, but negatively correlated, the Lukasiewicz operator is used; thus, $x_i(t) \otimes x_j(t) = \max(x_i(t) + x_j(t) - 1, 0)$. On the other hand, if two variables are not correlated, that is, these variables are of the different nature, the proper operator is ordinary product, and thus, $x_i(t) \otimes x_j(t) = x_i(t) \cdot x_j(t)$.

In IBN, different operators for generalized product can be used for any pair of variables. In this way, the nature of the variables can be fully taken into consideration. Comparing to the Boolecube/Hillcube approach, in which it is assumed that all variables are of the different nature, IBN offers more descriptive power. The Boolecube/Hillcube implicitly take into account the structure of elements; its Boolean indicator functions have the role of IBN structure vector, but the only ordinary product is applied as the t -norm. In other words, if each generalized product operator in IBN is treated as ordinary product, the proposed approach reduces to the Boolecube/Hillcube approach (see (4), (6), and (15)). Further, IBN utilize comprehensible and direct IBA transformation procedure from logical function to GBP given in (13), which provides an easier interpretation of the model.

4.1. Software Support. For the task of modeling and analysis of complex dynamical systems, the software tool implementing interpolative Boolean networks is developed. The architecture of the software is shown in Figure 1. The software is based on jFuzzyIBATranslator, the very first software realization of IBA, which can also process conventional fuzzy expressions [45]. jFuzzyIBATranslator is an application written in the Java programming language using NetBeans 6.9 development environment. The task of jFuzzyIBATranslator is to transform logical functions into corresponding generalized Boolean polynomials according to the transformation rules given in (13). Logical functions are validated (syntax validation, nested parentheses, etc.) and then forwarded to jFuzzyIBATranslator. The minimal GBPs are obtained after application of the distributivity law and the rule of idempotency. The developed software tool is covered by JUnit tests, a popular testing framework for Java [46], to verify software correctness.

The procedure for complex systems modeling and analysis using the software tool is the following. The input to the software is the number of nodes of IBN, where each node

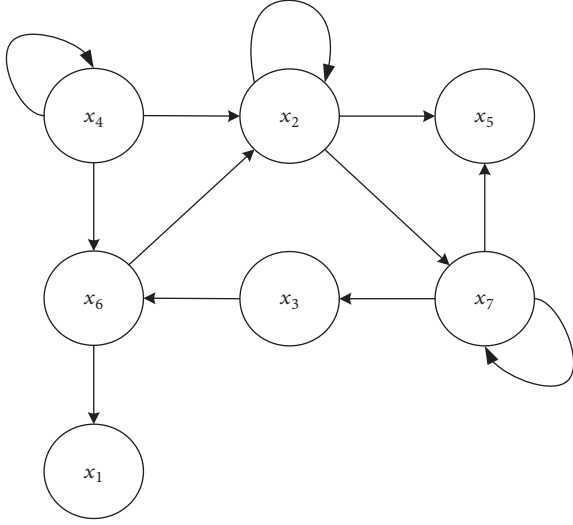


FIGURE 2: Structure of WNT5A gene regulatory network.

represents a system's component, and an initial state and a logical function for each node. The states take initial values from the unit interval. Logical functions are validated and transformed into GBPs. The nodes evolve in discrete time, governed by the GBPs. Since the initial states and the GBPs for each node are obtained, IBN can be simulated. The software tool predicts the future behavior of a complex system. After a number of iterations, an attractor is reached, while the dynamics of each node are presented in the graphical form. The software can be used for analysis of classical Boolean networks. In this case, it is sufficient to enter the binary initial states for each node.

5. Illustrative Examples

To illustrate the proposed approach, two classical BN models are taken from the literature as examples and generalized using IBA. The obtained IBN models are then simulated with the software tool presented in Section 4.1. Further, the obtained IBN results are compared to the results of conventional fuzzy BN and Boolecube approach.

5.1. Comparing the IBN Approach to the Conventional Fuzzy BN Approach. To compare the proposed IBN approach with the conventional fuzzy BN approach, the example of WNT5A gene regulatory network (Figure 2) is used. The WNT5A gene regulatory network is related to melanoma and thoroughly studied as BN in [47, 48]. First, the basic model is generalized to obtain both fuzzy BN and IBN models. Further, these real-valued networks are simulated using two t -norms: minimum and ordinary product. Finally, the obtained results are compared and discussed.

The basic WNT5A gene regulatory network dynamics is described with the following logical expressions [48]:

$$x_1(t+1) = \neg x_6(t)$$

$$x_2(t+1) = (\neg x_2(t) \wedge x_4(t) \wedge x_6(t))$$

$$\vee (x_2(t) \wedge (x_4(t) \vee x_6(t)))$$

$$x_3(t+1) = \neg x_7(t)$$

$$x_4(t+1) = x_4(t)$$

$$x_5(t+1) = x_2(t) \wedge \neg x_7(t)$$

$$x_6(t+1) = x_3(t) \vee x_4(t)$$

$$x_7(t+1) = \neg x_2(t) \vee x_7(t),$$

(18)

where $x(t+1) = [x_i(t+1) \mid i = 1, \dots, 7]^T$, $x(t+1) \in \{0, 1\}^7$.

The corresponding IBN model, presented in (A.1) of Appendix A, is obtained by mapping Boolean functions from (18) into the generalized Boolean polynomials. To translate the obtained GBPs, two operators of generalized product are used and their results are compared. In the first case scenario, a minimum function is used as an operator of generalized product; that is, $x_i \otimes x_j = \min(x_i, x_j)$, where $i, j = 1, \dots, 7$. In the second case scenario, generalized product is replaced with an ordinary product, $x_i \otimes x_j = x_i \cdot x_j$, where $i, j = 1, \dots, 7$. For both of selected operators, the proposed IBN model is reduced to the classical fuzzy network model and results are compared.

Case 1. In the first case scenario, we investigate the differences between IBN and classical fuzzy BN approach in the case of the Gödel (minimum) t -norm. The following equations (19) and (20) present, respectively, IBN and conventional fuzzy BN generalizations of the classical BN given in (18), obtained with minimum operator.

IBN model with minimum operator ($\otimes := \min$) for WNT5A gene regulation is defined with the following set of equations:

$$x_1(t+1) = 1 - x_6(t)$$

$$x_2(t+1) = \min(x_4(t), x_6(t)) - 2$$

$$\cdot \min(x_2(t), x_4(t), x_6(t))$$

$$+ \min(x_2(t), x_4(t))$$

$$+ \min(x_2(t), x_6(t))$$

(19)

$$x_3(t+1) = 1 - x_7(t)$$

$$x_4(t+1) = x_4(t)$$

$$x_5(t+1) = x_2(t) - \min(x_2(t), x_7(t))$$

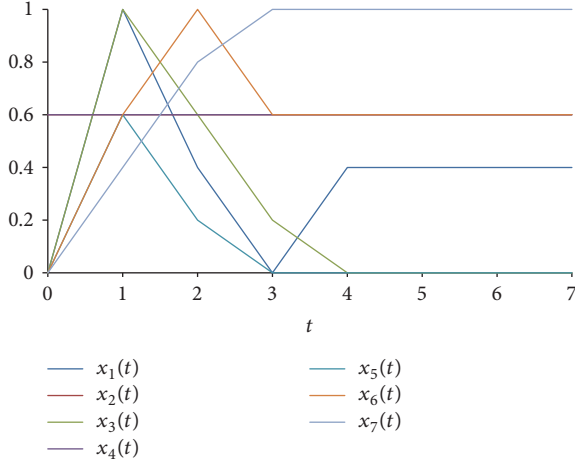
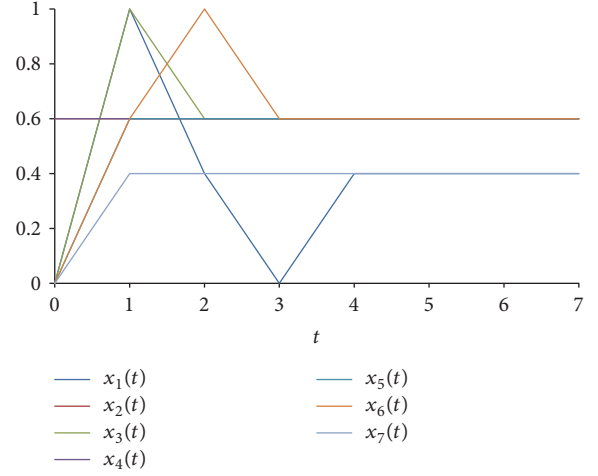
$$x_6(t+1) = x_3(t) + x_4(t) - \min(x_3(t), x_4(t))$$

$$x_7(t+1) = 1 - x_2(t) + \min(x_2(t), x_7(t)),$$

where $x(t) = [x_i(t) \mid i = 1, \dots, 7]^T$ and $x(t) \in [0, 1]^7$.

TABLE 1: Final state values for WNT5A gene regulation network nodes in IBN and fuzzy BN models obtained with minimum operator.

| | x_1 | x_2 | x_3 | x_4 | x_5 | x_6 | x_7 |
|-------------------------|-------|-------|------------|-------|------------|-------|------------|
| Initial states | 0 | 0.6 | 0 | 0.6 | 0 | 0 | 0 |
| Final states (IBN) | 0.4 | 0.6 | 0 | 0.6 | 0 | 0.6 | 1 |
| Final states (fuzzy BN) | 0.4 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.4 |

FIGURE 3: Dynamics of the IBN model for WNT5A gene regulation with initial state conditions $X(0) = [0; 0.6; 0; 0.6; 0; 0; 0]$ and minimum operator used as t -norm. The IBN model reaches the point attractor in 4th iteration $X(4) = [0.4; 0.6; 0; 0.6; 0; 0.6; 1]$.FIGURE 4: Dynamics of the fuzzy BN model for WNT5A gene regulation with initial state conditions $X(0) = [0; 0.6; 0; 0.6; 0; 0; 0]$ and minimum operator used as t -norm. The fuzzy BN model reaches the point attractor in 4th iteration $X(4) = [0.4; 0.6; 0.6; 0.6; 0.6; 0.6; 0.4]$.

The conventional fuzzy BN model with minimum operator for WNT5A gene regulation is defined with the following set of equations:

$$\begin{aligned}
x_1(t+1) &= 1 - x_6(t) \\
x_2(t+1) &= \max(\min(1 - x_2(t), x_4(t), x_6(t)), \\
&\quad \min(x_2(t), \max(x_4(t), x_6(t)))) \\
x_3(t+1) &= 1 - x_7(t) \\
x_4(t+1) &= x_4(t) \\
x_5(t+1) &= \min(x_2(t), 1 - x_7(t)) \\
x_6(t+1) &= \max(x_3(t), x_4(t)) \\
x_7(t+1) &= \max(1 - x_2(t), x_7(t)).
\end{aligned} \tag{20}$$

Both IBN and fuzzy BN models are simulated for the same initial conditions. The obtained dynamics are presented in Figures 3 and 4.

The final results of the experiment are presented in Table 1.

Although the dynamics of the state variables are similar in two approaches, state variables $x_3(t)$, $x_5(t)$, and $x_7(t)$ reach significantly different point attractors in IBN model as compared with fuzzy BN model (values are bold in Table 1). These differences are due to inconsistency (in a Boolean sense) of the conventional fuzzy logic approach.

Case 2. In the second case scenario, we investigate the differences between IBN and classical fuzzy BN approach in the case of the ordinary product t -norm. The following equations (21) and (22) present, respectively, IBN and conventional fuzzy BN generalizations of the classical BN given in (18), obtained with product operator.

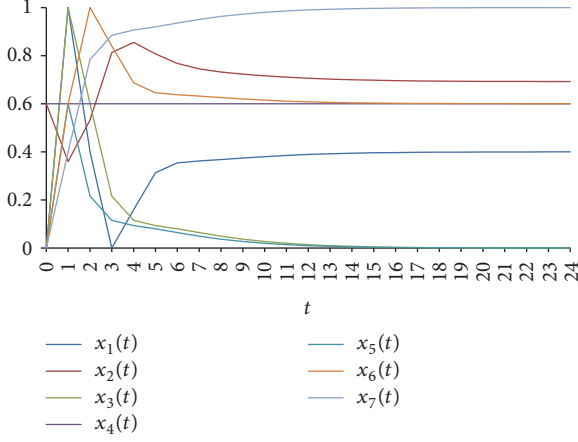
The IBN model with product operator ($\otimes := \cdot$) for WNT5A gene regulation is defined by the following set of equations:

$$\begin{aligned}
x_1(t+1) &= 1 - x_6(t) \\
x_2(t+1) &= x_4(t) \cdot x_6(t) - 2 \cdot x_2(t) \cdot x_4(t) \cdot x_6(t) \\
&\quad + x_2(t-1) \cdot x_4(t) + x_2(t) \cdot x_6(t) \\
x_3(t+1) &= 1 - x_7(t) \\
x_4(t+1) &= x_4(t) \\
x_5(t+1) &= x_2(t) - x_2(t) \cdot x_7(t) \\
x_6(t+1) &= x_3(t) + x_4(t) - x_3(t) \cdot x_4(t) \\
x_7(t+1) &= 1 - x_2(t) + x_2(t) \cdot x_7(t),
\end{aligned} \tag{21}$$

where $x(t) = [x_i(t) \mid i = 1, \dots, 7]^T$ and $x(t) \in [0, 1]^7$.

TABLE 2: Final state values for WNT5A gene regulation network nodes in IBN and fuzzy BN models obtained with product operator.

| | x_1 | x_2 | x_3 | x_4 | x_5 | x_6 | x_7 |
|-------------------------|-------|-------------|-------|-------|-------|-------|-------|
| Initial states | 0 | 0.6 | 0 | 0.6 | 0 | 0 | 0 |
| Final states (IBN) | 0.4 | 0.69 | 0 | 0.6 | 0 | 0.6 | 1 |
| Final states (fuzzy BN) | 0.4 | 0.55 | 0 | 0.6 | 0 | 0.6 | 1 |

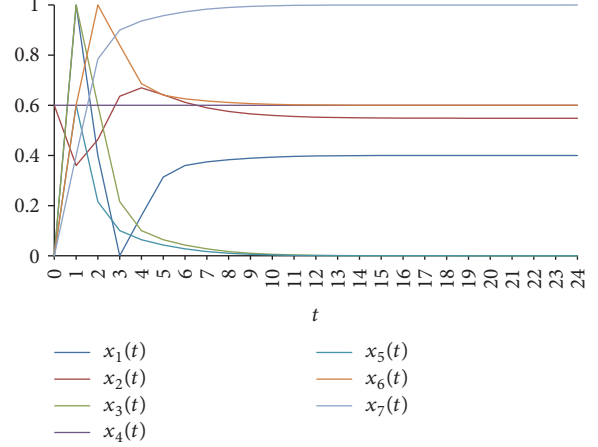
FIGURE 5: Dynamics of the IBN model for WNT5A gene regulation with initial state conditions $X(0) = [0; 0.6; 0; 0.6; 0; 0; 0]$ and product operator used as t -norm. The IBN model reaches the point attractor in 22nd iteration $X(22) = [0.4; 0.69; 0; 0.6; 0; 0.6; 1]$.

On the other hand, the conventional fuzzy BN model with the same product operator for WNT5A gene regulation is defined as

$$\begin{aligned}
 x_1(t+1) &= 1 - x_6(t) \\
 x_2(t+1) &= x_4(t) \cdot x_6(t) - 2 \cdot x_2(t) \cdot x_4(t) \cdot x_6(t) \\
 &\quad + x_2(t) \cdot x_4(t) + x_2(t) \cdot x_6(t) - ((1 - x_2(t)) \\
 &\quad \cdot x_2(t) \cdot x_4(t) \cdot x_6(t)) \\
 &\quad \cdot (x_4(t) + x_6(t) - x_4(t) \cdot x_6(t)) \\
 x_3(t+1) &= 1 - x_7(t) \\
 x_4(t+1) &= x_4(t) \\
 x_5(t+1) &= x_2(t) - x_2(t) \cdot x_7(t) \\
 x_6(t+1) &= x_3(t) + x_4(t) - x_3(t) \cdot x_4(t) \\
 x_7(t+1) &= 1 - x_2(t) + x_2(t) \cdot x_7(t).
 \end{aligned} \tag{22}$$

Notice that in this case the only difference between the two models is in the second equation. The Boolean function for state variable $x_2(t)$ from (18) is too complex to be generalized in the Boolean consistent way with the conventional fuzzy approach.

Models are simulated for the same initial conditions, and obtained dynamics are presented in Figures 5 and 6.

FIGURE 6: Dynamics of the fuzzy BN model for WNT5A gene regulation with initial state conditions $X(0) = [0; 0.6; 0; 0.6; 0; 0; 0]$ and product operator used as t -norm. The fuzzy BN model reaches the point attractor in 22nd iteration $X(22) = [0.4; 0.55; 0; 0.6; 0; 0.6; 1]$.

The final results of the experiment are presented in Table 2.

In this case scenario, the dynamics of the state variables of the two models are much more similar than in the previous case. There is only one state that significantly differs in the value of the point attractor reached, and it is $x_2(t)$. It is a direct consequence of the difference between IBN and fuzzy BN model for x_2 state space equations. However, this difference is important since it indicates inconsistency in the classical fuzzy BNs. It is also important to mention that the presented WNT5A gene regulation models based on product operator need more time to reach its point attractors comparing to the previous models based on minimum operator. This is due to the characteristics of the operator itself.

5.2. Comparing the IBN Approach to the Boolecube Approach.

To compare the proposed IBN approach with the Boolecube approach, the example of modeling gene interaction is used. The starting point is the classical BN model used in [7] to describe interactions between the *Drosophila* segment polarity genes. The following equations specify its dynamic:

$$\text{SLP}_i(t+1) = \begin{cases} 0, & i \in \{1, 2\} \\ 1, & i \in \{3, 4\} \end{cases}$$

$$\text{wg}_i(t+1) = (\text{CIA}_i(t) \wedge \text{SLP}_i(t) \wedge \neg \text{CIR}_i(t))$$

TABLE 3: Final state values for *Drosophila* segment polarity genes model variables in IBN approach and Boolecube approach.

| | SLP_i | wq_i | WG_i | en_i | EN_i | hh_i | HH_i | ptc_i | PTC_i | ci_i | CI_i | CIA_i | CIR_i |
|--------------------------|---------|--------|--------|-------------|-------------|-------------|-------------|---------|---------|-------------|-------------|---------|-------------|
| Initial states | 0.5 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 |
| Final states (IBN) | 0.5 | 0 | 0 | 0.35 | 0.35 | 0.12 | 0.12 | 0 | 0 | 0.65 | 0.65 | 0 | 0.65 |
| Final states (Boolecube) | 0.5 | 0 | 0 | 0.45 | 0.45 | 0.2 | 0.2 | 0 | 0 | 0.55 | 0.55 | 0 | 0.55 |

$$\begin{aligned}
& \vee (wg_i(t) \wedge (CIA_i(t) \vee SLP_i(t)) \wedge \neg CIR_i(t)) \\
WG_i(t+1) &= wg_i(t) \\
en_i(t+1) &= (WG_{i-1}(t) \vee WG_{i+1}(t)) \wedge \neg SLP_i(t) \\
EN_i(t+1) &= en_i(t) \\
hh_i(t+1) &= EN_i(t) \wedge \neg CIR_i(t) \\
HH_i(t+1) &= hh_i(t) \\
ptc_i(t+1) &= CIA_i(t) \wedge \neg EN_i(t) \wedge \neg CIR_i(t) \\
PTC_i(t+1) &= ptc_i(t) \vee (PTC_i(t) \wedge \neg HH_{i-1}(t) \\
& \wedge \neg HH_{i+1}(t)) \\
ci_i(t+1) &= \neg EN_i(t) \\
CI_i(t+1) &= ci_i(t) \\
CIA_i(t+1) &= CI_i(t) \wedge (\neg PTC_i(t) \vee HH_{i-1}(t) \\
& \vee HH_{i+1}(t) \vee hh_{i-1}(t) \vee hh_{i+1}(t)) \\
CIR_i(t+1) &= CI_i(t) \wedge PTC_i(t) \wedge \neg HH_{i-1}(t) \\
& \wedge \neg HH_{i+1}(t) \wedge \neg hh_{i-1}(t) \wedge \neg hh_{i+1}(t),
\end{aligned} \tag{23}$$

where $(t) = [SLP_i(t), wg_i(t), WG_i(t), en_i(t), EN_i(t), hh_i(t), HH_i(t), ptc_i(t), PTC_i(t), ci_i(t), CI_i(t), CIA_i(t), CIR_i(t)]^T$ is the network state at time t , and $x(t) \in \{0, 1\}^{13}$.

The IBN model, presented in (B.1) of Appendix B, is obtained by mapping the Boolean functions into generalized Boolean polynomials. Operators of generalized product in this model are treated in the following way. Since the variables $WG_{i-1}(t)$ and $WG_{i+1}(t)$, $HH_{i-1}(t)$ and $HH_{i+1}(t)$, $hh_{i-1}(t)$ and $hh_{i+1}(t)$ are of the same nature, the adequate generalized product operator for these pairs of variables is minimum function. Thus,

$$\begin{aligned}
& WG_{i-1}(t) \otimes WG_{i+1}(t) \\
&= \min(WG_{i-1}(t), WG_{i+1}(t)), \\
& HH_{i-1}(t) \otimes HH_{i+1}(t) = \min(HH_{i-1}(t), HH_{i+1}(t)), \\
& hh_{i-1}(t) \otimes hh_{i+1}(t) = \min(hh_{i-1}(t), hh_{i+1}(t)).
\end{aligned} \tag{24}$$

The remaining variables are of different nature and, therefore, the adequate operator of generalized product is the ordinary product. The final IBN model is presented in (B.2) of Appendix B.

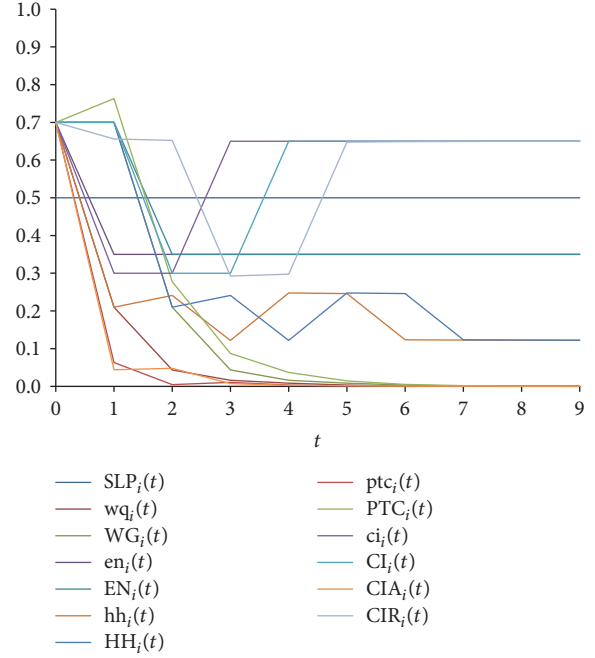


FIGURE 7: Dynamics of the IBN model of *Drosophila* segment polarity genes interactions with initial state conditions $X(0) = [0.5; 0.7; 0.7; 0.7; 0.7; 0.7; 0.7; 0.7; 0.7; 0.7; 0.7; 0.7; 0.7]$. The IBN model reaches the point attractor in 7th iteration $X(7) = [0.5; 0; 0; 0.35; 0.35; 0.12; 0.12; 0; 0; 0.65; 0.65; 0; 0.65]$.

On the other hand, in the Boolecube/Hillcube approach, each generalized product in (B.1) is replaced with an ordinary product. The final Boolecube/Hillcube model is presented in (B.3).

Both IBN and Boolecube/Hillcube models are simulated using the following initial conditions $(0) = [0.5, 0.7, 0.7, 0.7, 0.7, 0.7, 0.7, 0.7, 0.7, 0.7, 0.7, 0.7, 0.7]$. The obtained models' dynamics are presented in Figures 7 and 8.

The final results of the experiment are presented in Table 3.

It is easy to see differences in IBN and Boolecube models dynamics and their point attractors reached. This difference is a direct consequence of the nature of the model variables that is taken into account when selecting an appropriate operator in the IBN approach, unlike the Boolecube/Hillcube approach. For this reason, the IBN offers superior descriptive power and it is adaptive with respect to nature of inputs. In fact, the IBN approach can be considered as the generalization of the Boolecube/Hillcube approach since it can be reduced to it when only the product t -norm is used as an operator of the generalized product.

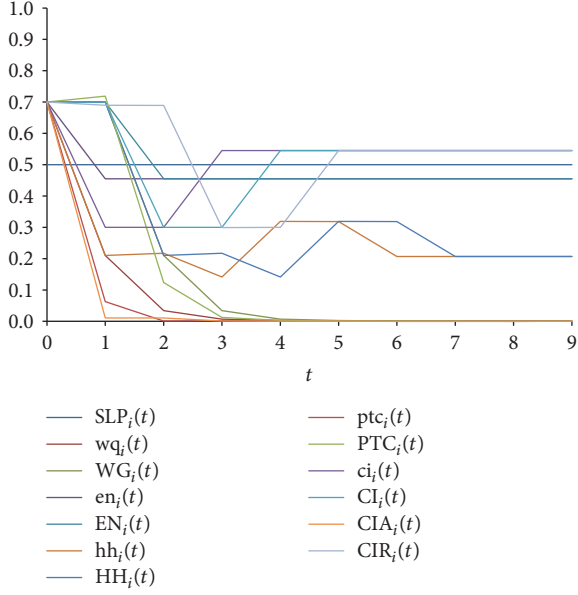


FIGURE 8: Dynamics of the Boolecube model of *Drosophila* segment polarity genes interactions with initial state conditions $X(0) = [0.5; 0.7; 0.7; 0.7; 0.7; 0.7; 0.7; 0.7; 0.7; 0.7; 0.7; 0.7; 0.7; 0.7]$. The Boolecube model reaches the point attractor in 7th iteration $X(7) = [0.5; 0; 0; 0.45; 0.45; 0.2; 0.2; 0; 0; 0.55; 0.55; 0; 0.55]$.

6. Conclusion

Logical approaches are widely used for modeling and analysis of complex dynamical systems. Classical models are within the Boolean frame, which secures the validity of their dynamics. Such models are adequate for modeling system which exhibit threshold behavior. If more detail is required, the Boolean networks should be generalized to include real-valued case.

In this paper, interpolative Boolean networks are proposed. They are based on interpolative Boolean algebra and represent a Boolean consistent real-valued generalization of classical Boolean networks. The validity of their dynamics is secured thanks to the principle of structure functionality. Furthermore, the proposed approach is a generalization of the current state-of-the-art approaches (e.g., Boolecube/Hillcube) and offers a guideline for the appropriate realization of generalized product. Also, IBN utilizes comprehensible and direct IBA transformation procedure from logical function to GBP.

Interpolative Boolean networks can be used to predict behavior of complex systems. For the purpose of their practical application, a simple software support tool is developed. The proposed approach is applied on two examples from the literature and compared against other real-valued logical approaches. Interpolative Boolean networks offer superior descriptive power compared to other approaches. This approach facilitates modeling and analysis of systems too complex for quantitative mathematical approaches.

Appendix

A. WNT5A Gene Regulatory Network Realized Using IBN

The WNT5A gene regulatory network realized as IBN is defined by the following equations:

$$\begin{aligned}
 x_1(t+1) &= 1 - x_6(t) \\
 x_2(t+1) &= x_4(t) \otimes x_6(t) - 2x_2(t) \otimes x_4(t) \otimes x_6(t) \\
 &\quad + x_2(t) \otimes x_4(t) + x_2(t) \otimes x_6(t) \\
 x_3(t+1) &= 1 - x_7(t) \\
 x_4(t+1) &= x_4(t) \\
 x_5(t+1) &= x_2(t) - x_2(t) \otimes x_7(t) \\
 x_6(t+1) &= x_3(t) + x_4(t) - x_3(t) \otimes x_4(t) \\
 x_7(t+1) &= 1 - x_2(t) + x_2(t) \otimes x_7(t),
 \end{aligned} \tag{A.1}$$

where $x(t) = [x_i(t) \mid i = 1, \dots, 7]^T$ and $x(t) \in [0, 1]^7$.

B. *Drosophila* Segment Polarity Genes Regulatory Network Realized Using IBN

The *Drosophila* segment polarity genes network realized as IBN is defined by the following equations:

$$\begin{aligned}
 \text{SLP}_i(t+1) &= \text{SLP}_i(t) \\
 \text{wg}_i(t+1) &= (\text{CIA}_i(t) \wedge \text{SLP}_i(t) \wedge \neg \text{CIR}_i(t)) \\
 &\quad \vee (\text{wg}_i(t) \wedge (\text{CIA}_i(t) \vee \text{SLP}_i(t)) \wedge \neg \text{CIR}_i(t)) \\
 &= \text{CIA}_i(t) \otimes \text{SLP}_i(t) - \text{CIA}_i(t) \otimes \text{SLP}_i(t) \\
 &\quad \otimes \text{CIR}_i(t) + \text{CIA}_i(t) \otimes \text{wg}_i(t) + \text{SLP}_i(t) \otimes \text{wg}_i(t) \\
 &\quad - 2 * \text{CIA}_i(t) \otimes \text{SLP}_i(t) \otimes \text{wg}_i(t) - \text{CIA}_i(t) \\
 &\quad \otimes \text{CIR}_i(t) \otimes \text{wg}_i(t) - \text{CIA}_i(t) \otimes \text{CIR}_i(t) \otimes \text{wg}_i(t) \\
 &\quad + 2 * \text{CIA}_i(t) \otimes \text{SLP}_i(t) \otimes \text{CIR}_i(t) \otimes \text{wg}_i(t) \\
 \text{WG}_i(t+1) &= \text{wg}_i(t) \\
 \text{en}_i(t+1) &= (\text{WG}_{i-1}(t) \vee \text{WG}_{i+1}(t)) \wedge \neg \text{SLP}_i(t) \\
 &= \text{WG}_{i-1}(t) + \text{WG}_{i+1}(t) - \text{WG}_{i-1}(t) \otimes \text{WG}_{i+1}(t) \\
 &\quad - \text{WG}_{i-1}(t) \otimes \text{SLP}_i(t) - \text{WG}_{i+1}(t) \otimes \text{SLP}_i(t) \\
 &\quad + \text{WG}_{i-1}(t) \otimes \text{WG}_{i+1}(t) \otimes \text{SLP}_i(t) \\
 \text{EN}_i(t+1) &= \text{en}_i(t) \\
 \text{hh}_i(t+1) &= \text{EN}_i(t) \wedge \neg \text{CIR}_i(t) = \text{EN}_i(t) - \text{EN}_i(t) \\
 &\quad \otimes \text{CIR}_i(t) \\
 \text{HH}_i(t+1) &= \text{hh}_i(t)
 \end{aligned}$$

$$\begin{aligned}
& * CIR_i(t) \\
HH_i(t+1) &= hh_i(t) \\
ptc_i(t+1) &= CIA_i(t) \wedge \neg EN_i(t) \wedge \neg CIR_i(t) \\
&= CIA_i(t) - CIA_i(t) * EN_i(t) - CIA_i(t) \\
& * CIA_i(t) + CIA_i(t) * EN_i(t) * CIR_i(t) \\
PTC_i(t+1) &= ptc_i(t) \vee (PTC_i(t) \wedge \neg HH_{i-1}(t) \\
& \wedge \neg HH_{i+1}(t)) = ptc_i(t) + PTC_i(t) - PTC_i(t) \\
& * HH_{i-1}(t) - PTC_i(t) * HH_{i+1}(t) + PTC_i(t) \\
& * \min(HH_{i-1}(t), HH_{i+1}(t)) - ptc_i(t) * PTC_i(t) \\
& + ptc_i(t) * PTC_i(t) * HH_{i-1}(t) + ptc_i(t) \\
& * PTC_i(t) * HH_{i+1}(t) - ptc_i(t) * PTC_i(t) \\
& * \min(HH_{i-1}(t), HH_{i+1}(t)) \\
ci_i(t+1) &= \neg EN_i(t) = 1 - EN_i(t) \\
CI_i(t+1) &= ci_i(t) \\
CIA_i(t+1) &= CI_i(t) \wedge (\neg PTC_i(t) \vee HH_{i-1}(t) \\
& \vee HH_{i+1}(t) \vee hh_{i-1}(t) \vee hh_{i+1}(t)) = CI_i(t) \\
& - CI_i(t) * PTC_i(t) + CI_i(t) * PTC_i(t) \\
& * HH_{i-1}(t) + CI_i(t) * PTC_i(t) * HH_{i+1}(t) \\
& + CI_i(t) * PTC_i(t) * hh_{i-1}(t) + CI_i(t) \\
& * PTC_i(t) * hh_{i+1}(t) - CI_i(t) * PTC_i(t) \\
& * \min(HH_{i-1}(t), HH_{i+1}(t)) - CI_i(t) * PTC_i(t) \\
& * HH_{i-1}(t) * hh_{i-1}(t) - CI_i(t) * PTC_i(t) \\
& * HH_{i-1}(t) * hh_{i+1}(t) - CI_i(t) * PTC_i(t) \\
& * HH_{i+1}(t) * hh_{i-1}(t) - CI_i(t) * PTC_i(t) \\
& * HH_{i+1}(t) * hh_{i+1}(t) - CI_i(t) * PTC_i(t) \\
& * \min(hh_{i-1}(t), hh_{i+1}(t)) + CI_i(t) * PTC_i(t) \\
& * \min(HH_{i-1}(t), HH_{i+1}(t)) * hh_{i-1}(t) + CI_i(t) \\
& * PTC_i(t) * \min(HH_{i-1}(t), HH_{i+1}(t)) \\
& * hh_{i+1}(t) + CI_i(t) * PTC_i(t) * HH_{i-1}(t) \\
& * \min(hh_{i-1}(t), hh_{i+1}(t)) + CI_i(t) * PTC_i(t) \\
& * HH_{i+1}(t) * \min(hh_{i-1}(t), hh_{i+1}(t)) - CI_i(t) \\
& * PTC_i(t) * \min(HH_{i-1}(t), HH_{i+1}(t)) \\
& * \min(hh_{i-1}(t), hh_{i+1}(t)) \\
CIR_i(t+1) &= CI_i(t) \wedge PTC_i(t) \wedge \neg HH_{i-1}(t) \\
& \wedge \neg HH_{i+1}(t) \wedge \neg hh_{i-1}(t) \wedge \neg hh_{i+1}(t) = CI_i(t)
\end{aligned}$$

$$\begin{aligned}
& * PTC_i(t) - CI_i(t) * PTC_i(t) * HH_{i-1}(t) \\
& - CI_i(t) * PTC_i(t) * HH_{i+1}(t) - CI_i(t) \\
& * PTC_i(t) * hh_{i-1}(t) - CI_i(t) * PTC_i(t) \\
& * hh_{i+1}(t) + CI_i(t) * PTC_i(t) \\
& * \min(HH_{i-1}(t), HH_{i+1}(t)) + CI_i(t) * PTC_i(t) \\
& * HH_{i-1}(t) * hh_{i-1}(t) + CI_i(t) * PTC_i(t) \\
& * HH_{i-1}(t) * hh_{i+1}(t) + CI_i(t) * PTC_i(t) \\
& * HH_{i+1}(t) * hh_{i-1}(t) + CI_i(t) * PTC_i(t) \\
& * HH_{i+1}(t) * hh_{i+1}(t) + CI_i(t) * PTC_i(t) \\
& * \min(hh_{i-1}(t), hh_{i+1}(t)) - CI_i(t) * PTC_i(t) \\
& * \min(HH_{i-1}(t), HH_{i+1}(t)) * hh_{i-1}(t) - CI_i(t) \\
& * PTC_i(t) * \min(HH_{i-1}(t), HH_{i+1}(t)) \\
& * hh_{i+1}(t) - CI_i(t) * PTC_i(t) * HH_{i-1}(t) \\
& * \min(hh_{i-1}(t), hh_{i+1}(t)) - CI_i(t) * PTC_i(t) \\
& * HH_{i+1}(t) * \min(hh_{i-1}(t), hh_{i+1}(t)) + CI_i(t) \\
& * PTC_i(t) * \min(HH_{i-1}(t), HH_{i+1}(t)) \\
& * \min(hh_{i-1}(t), hh_{i+1}(t)),
\end{aligned} \tag{B.2}$$

where $x(t) = [SLP_i(t), wg_i(t), WG_i(t), en_i(t), EN_i(t), hh_i(t), HH_i(t), ptc_i(t), PTC_i(t), ci_i(t), CI_i(t), CIA_i(t), CIR_i(t)]^T$ is the network state at time t , and $x(t) \in [0, 1]^{15}$.

The final Boolecube model of the *Drosophila* segment polarity genes network is defined by the following equations:

$$\begin{aligned}
SLP_i(t+1) &= SLP_i(t) \\
wg_i(t+1) &= (CIA_i(t) \wedge SLP_i(t) \wedge \neg CIR_i(t)) \\
& \vee (wg_i(t) \wedge (CIA_i(t) \vee SLP_i(t)) \wedge \neg CIR_i(t)) \\
& = CIA_i(t) * SLP_i(t) - CIA_i(t) * SLP_i(t) \\
& * CIR_i(t) + CIA_i(t) * wg_i(t) + SLP_i(t) * wg_i(t) \\
& - 2 * CIA_i(t) * SLP_i(t) * wg_i(t) - CIA_i(t) \\
& * CIR_i(t) * wg_i(t) - CIA_i(t) * CIR_i(t) * wg_i(t) \\
& + 2 * CIA_i(t) * SLP_i(t) * CIR_i(t) * wg_i(t) \\
WG_i(t+1) &= wg_i(t) \\
en_i(t+1) &= (WG_{i-1}(t) \vee WG_{i+1}(t)) \wedge \neg SLP_i(t) \\
& = WG_{i-1}(t) + WG_{i+1}(t) - WG_{i-1}(t) \\
& * WG_{i+1}(t) - WG_{i-1}(t) * SLP_i(t) - WG_{i+1}(t)
\end{aligned}$$

$$\begin{aligned}
& * SLP_i(t) + WG_{i-1}(t) * WG_{i+1}(t) * SLP_i(t) \\
EN_i(t+1) &= en_i(t) \\
hh_i(t+1) &= EN_i(t) \wedge \neg CIR_i(t) = EN_i(t) - EN_i(t) \\
& * CIR_i(t) \\
HH_i(t+1) &= hh_i(t) \\
ptc_i(t+1) &= CIA_i(t) \wedge \neg EN_i(t) \wedge \neg CIR_i(t) \\
& = CIA_i(t) - CIA_i(t) * EN_i(t) - CIA_i(t) \\
& * CIR_i(t) + CIA_i(t) * EN_i(t) * CIR_i(t) \\
PTC_i(t+1) &= ptc_i(t) \vee (PTC_i(t) \wedge \neg HH_{i-1}(t) \\
& \wedge \neg HH_{i+1}(t)) = ptc_i(t) + PTC_i(t) - PTC_i(t) \\
& * HH_{i-1}(t) - PTC_i(t) * HH_{i+1}(t) + PTC_i(t) \\
& * HH_{i-1}(t) * HH_{i+1}(t) - ptc_i(t) * PTC_i(t) \\
& + ptc_i(t) * PTC_i(t) * HH_{i-1}(t) + ptc_i(t) \\
& * PTC_i(t) * HH_{i+1}(t) - ptc_i(t) * PTC_i(t) \\
& * HH_{i-1}(t) * HH_{i+1}(t) \\
ci_i(t+1) &= \neg EN_i(t) = 1 - EN_i(t) \\
CI_i(t+1) &= ci_i(t) \\
CIA_i(t+1) &= CI_i(t) \wedge (\neg PTC_i(t) \vee HH_{i-1}(t) \\
& \vee HH_{i+1}(t) \vee hh_{i-1}(t) \vee hh_{i+1}(t)) = CI_i(t) \\
& - CI_i(t) * PTC_i(t) + CI_i(t) * PTC_i(t) \\
& * HH_{i-1}(t) + CI_i(t) * PTC_i(t) * HH_{i+1}(t) \\
& + CI_i(t) * PTC_i(t) * hh_{i-1}(t) + CI_i(t) \\
& * PTC_i(t) * hh_{i+1}(t) - CI_i(t) * PTC_i(t) \\
& * HH_{i-1}(t) * HH_{i+1}(t) - CI_i(t) * PTC_i(t) \\
& * HH_{i-1}(t) * hh_{i-1}(t) - CI_i(t) * PTC_i(t) \\
& * HH_{i-1}(t) * hh_{i+1}(t) - CI_i(t) * PTC_i(t) \\
& * HH_{i+1}(t) * hh_{i-1}(t) - CI_i(t) * PTC_i(t) \\
& * HH_{i+1}(t) * hh_{i+1}(t) - CI_i(t) * PTC_i(t) \\
& * hh_{i-1}(t) * hh_{i+1}(t) + CI_i(t) * PTC_i(t) \\
& * HH_{i-1}(t) * HH_{i+1}(t) * hh_{i-1}(t) + CI_i(t) \\
& * PTC_i(t) * HH_{i-1}(t) * HH_{i+1}(t) * hh_{i+1}(t) \\
& + CI_i(t) * PTC_i(t) * HH_{i-1}(t) * hh_{i-1}(t) \\
& * hh_{i+1}(t) + CI_i(t) * PTC_i(t) * HH_{i+1}(t) \\
& * hh_{i-1}(t) * hh_{i+1}(t) - CI_i(t) * PTC_i(t) \\
& * HH_{i-1}(t) * HH_{i+1}(t) * hh_{i-1}(t) * hh_{i+1}(t)
\end{aligned}$$

$$\begin{aligned}
CIR_i(t+1) &= CI_i(t) \wedge PTC_i(t) \wedge \neg HH_{i-1}(t) \\
& \wedge \neg HH_{i+1}(t) \wedge \neg hh_{i-1}(t) \wedge \neg hh_{i+1}(t) = CI_i(t) \\
& * PTC_i(t) - CI_i(t) * PTC_i(t) * HH_{i-1}(t) \\
& - CI_i(t) * PTC_i(t) * HH_{i+1}(t) - CI_i(t) \\
& * PTC_i(t) * hh_{i-1}(t) - CI_i(t) * PTC_i(t) \\
& * hh_{i+1}(t) + CI_i(t) * PTC_i(t) * HH_{i-1}(t) \\
& * HH_{i+1}(t) + CI_i(t) * PTC_i(t) * HH_{i-1}(t) \\
& * hh_{i-1}(t) + CI_i(t) * PTC_i(t) * HH_{i-1}(t) \\
& * hh_{i-1}(t) + CI_i(t) * PTC_i(t) * HH_{i+1}(t) \\
& * hh_{i+1}(t) + CI_i(t) * PTC_i(t) * hh_{i-1}(t) \\
& * hh_{i+1}(t) - CI_i(t) * PTC_i(t) * HH_{i-1}(t) \\
& * HH_{i+1}(t) * hh_{i-1}(t) - CI_i(t) * PTC_i(t) \\
& * HH_{i-1}(t) * HH_{i+1}(t) * hh_{i+1}(t) - CI_i(t) \\
& * PTC_i(t) * HH_{i-1}(t) * hh_{i-1}(t) * hh_{i+1}(t) \\
& - CI_i(t) * PTC_i(t) * HH_{i+1}(t) * hh_{i-1}(t) \\
& * hh_{i+1}(t) + CI_i(t) * PTC_i(t) * HH_{i-1}(t) \\
& * HH_{i+1}(t) * hh_{i-1}(t) * hh_{i+1}(t), \tag{B.3}
\end{aligned}$$

where $x(t) = [SLP_i(t), wg_i(t), WG_i(t), en_i(t), EN_i(t), hh_i(t), HH_i(t), ptc_i(t), PTC_i(t), ci_i(t), CI_i(t), CIA_i(t), CIR_i(t)]^T$ is the network state at time t , and $x(t) \in [0, 1]^{13}$.

Disclosure

Short version of this paper was presented at "1st International Conference on Decision Support System Technology: Big Data Analytics for Decision Making," Belgrade, Serbia, 2015.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] R. Thomas, "Regulatory networks seen as asynchronous automata: a logical description," *Journal of Theoretical Biology*, vol. 153, no. 1, pp. 1–23, 1991.
- [2] M. Ghil, I. Zaliapin, and B. Coluzzi, "Boolean delay equations: a simple way of looking at complex systems," *Physica D: Nonlinear Phenomena*, vol. 237, no. 23, pp. 2967–2986, 2008.
- [3] D. M. Wittmann, J. Krumsiek, J. Saez-Rodriguez, D. A. Lauffenburger, S. Klamt, and F. J. Theis, "Transforming Boolean models to continuous models: methodology and application to T-cell receptor signaling," *BMC Systems Biology*, vol. 3, article 98, 2009.

- [4] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 1, pp. 28–44, 1973.
- [5] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, 1969.
- [6] J. A. Tomé and J. P. Carvalho, "Fuzzy Boolean nets—a nature inspired model for learning and reasoning," *Fuzzy Sets and Systems*, vol. 253, pp. 1–27, 2014.
- [7] A. Saadatpour and R. Albert, "Boolean modeling of biological regulatory networks: a methodology tutorial," *Methods*, vol. 62, no. 1, pp. 3–12, 2013.
- [8] S. Videla, C. Guziolowski, F. Eduati et al., "Learning Boolean logic models of signaling networks with ASP," *Theoretical Computer Science*, vol. 599, pp. 79–101, 2015.
- [9] D. M. Wittmann and F. J. Theis, "Dynamic regimes of random fuzzy logic networks," *New Journal of Physics*, vol. 13, Article ID 013041, 2011.
- [10] E. Goles and L. Salinas, "Comparison between parallel and serial dynamics of Boolean networks," *Theoretical Computer Science*, vol. 396, no. 1–3, pp. 247–253, 2008.
- [11] R. Thomas, "Boolean formalization of genetic control circuits," *Journal of Theoretical Biology*, vol. 42, no. 3, pp. 563–585, 1973.
- [12] R.-S. Wang, A. Saadatpour, and R. Albert, "Boolean modeling in systems biology: an overview of methodology and applications," *Physical Biology*, vol. 9, no. 5, Article ID 055001, 2012.
- [13] M. Chaves, R. Albert, and E. D. Sontag, "Robustness and fragility of Boolean models for genetic regulatory networks," *Journal of Theoretical Biology*, vol. 235, no. 3, pp. 431–449, 2005.
- [14] M. Chaves and A. Carta, "Attractor computation using interconnected Boolean networks: testing growth rate models in *E. Coli*," *Theoretical Computer Science*, vol. 599, pp. 47–63, 2015.
- [15] D. G. Wright, T. F. Stocker, and L. A. Mysak, "A note on quaternary climate modelling using Boolean delay equations," *Climate Dynamics*, vol. 4, no. 4, pp. 263–267, 1990.
- [16] A. Saunders and M. Ghil, "A Boolean delay equation model of ENSO variability," *Physica D: Nonlinear Phenomena*, vol. 160, no. 1–2, pp. 54–78, 2001.
- [17] I. Zaliapin, V. Keilis-Borok, and M. Ghil, "A boolean delay equation model of colliding cascades. Part I: multiple seismic regimes," *Journal of Statistical Physics*, vol. 111, no. 3–4, pp. 815–837, 2003.
- [18] I. Zaliapin, V. Keilis-Borok, and M. Ghil, "A boolean delay equation model of colliding cascades. Part II: prediction of critical transitions," *Journal of Statistical Physics*, vol. 111, no. 3–4, pp. 839–861, 2003.
- [19] G. Easton, R. J. Brooks, K. Georgieva, and I. Wilkinson, "Understanding the dynamics of industrial networks using Kauffman Boolean networks," *Advances in Complex Systems. A Multidisciplinary Journal*, vol. 11, no. 1, pp. 139–164, 2008.
- [20] B. Coluzzi, M. Ghil, S. Hallegatte, and G. Weisbuch, "Boolean delay equations on networks in economics and the geosciences," *International Journal of Bifurcation and Chaos*, vol. 21, no. 12, pp. 3511–3548, 2011.
- [21] J.-W. Gu, T.-K. Siu, and H. Zheng, "On modeling credit defaults: a probabilistic Boolean network approach," *Risk and Decision Analysis*, vol. 4, no. 2, pp. 119–129, 2013.
- [22] K. Kobayashi and K. Hiraishi, "A probabilistic approach to control of complex systems and its application to real-time pricing," *Mathematical Problems in Engineering*, vol. 2014, Article ID 906717, 8 pages, 2014.
- [23] T. Kok and P. Wang, "A study of 3-gene regulation networks using NK-boolean network model and fuzzy logic networking," in *Fuzzy Applications in Industrial Engineering*, vol. 201 of *Studies in Fuzziness and Soft Computing*, pp. 119–151, Springer, Berlin, Germany, 2006.
- [24] Y. Cao, P. P. Wang, and A. Tokuta, "Gene regulatory network modeling: a data driven approach," in *Fuzzy Logic*, vol. 215 of *Studies in Fuzziness and Soft Computing*, pp. 247–281, Springer, Berlin, Germany, 2007.
- [25] M. K. Morris, J. Saez-Rodriguez, D. C. Clarke, P. K. Sorger, and D. A. Lauffenburger, "Training signaling pathway maps to biochemical data with constrained fuzzy logic: quantitative analysis of liver cell responses to inflammatory stimuli," *PLoS Computational Biology*, vol. 7, no. 3, Article ID e1001099, 2011.
- [26] J. L. Salmeron and E. I. Papageorgiou, "A Fuzzy Grey Cognitive Maps-based Decision Support System for radiotherapy treatment planning," *Knowledge-Based Systems*, vol. 30, pp. 151–160, 2012.
- [27] A. Poret, C. M. Sousa, and J. P. Boissel, "A logic-based modeling derived from boolean networks: adding fuzzy logic and edge tuning," *Simulation*, vol. 1, article 10, 2014.
- [28] M. Grieb, A. Burkovski, J. E. Sträng et al., "Predicting variabilities in cardiac gene expression with a boolean network incorporating uncertainty," *PLoS ONE*, vol. 10, no. 7, Article ID e0131832, 2015.
- [29] P. Hájek, *Metamathematics of Fuzzy Logic*, Kluwer Academic, Dordrecht, The Netherlands, 1998.
- [30] S. Jenei, "A more efficient method for defining fuzzy connectives," *Fuzzy Sets and Systems*, vol. 90, no. 1, pp. 25–35, 1997.
- [31] D. Radojevic, "New [0, 1]-valued logic: a natural generalization of Boolean logic," *Yugoslav Journal of Operations Research*, vol. 10, no. 2, pp. 185–216, 2000.
- [32] H. Lähdesmäki, I. Shmulevich, and O. Yli-Harja, "On learning gene regulatory networks under the Boolean network model," *Machine Learning*, vol. 52, no. 1–2, pp. 147–167, 2003.
- [33] R. Sharan and R. M. Karp, "Reconstructing Boolean models of signaling," *Journal of Computational Biology*, vol. 20, no. 3, pp. 249–257, 2013.
- [34] D. M. Gabbay and F. Guenther, *Handbook of Philosophical Logic*, Springer-Verlag, Berlin, Germany, 2nd edition, 2011.
- [35] M. I. Davidich and S. Bornholdt, "Boolean Network Model Predicts Knockout Mutant Phenotypes of Fission Yeast," *PLoS ONE*, vol. 8, no. 9, Article ID e71786, 2013.
- [36] V. Dobric, "Fuzzification of boolean networks," in *Decision Making and Soft Computing—Proceedings of the 11th International FLINS Conference*, R. M. De Moraes, E. E. Kerre, L. Dos Santos Machado, and J. Lu, Eds., pp. 276–281, World Scientific, Singapore.
- [37] V. D. Dobrić, *Concepts modelling using consistent fuzzy logic [PhD, thesis]*, Faculty of Organisational Sciences, University of Belgrade, Belgrade, Serbia, 2016.
- [38] V. Dobrić, P. Milošević, and B. Petrović, "Modelling Complex Systems using Interpolative Boolean algebra," in *Proceedings of the 1st EWG-DSS International Conference on Decision Support System Technology: Big Data Analytics for Decision Making (ICDSSST '15)*, B. Delibašić and et al., Eds., Belgrade, Serbia, May 2015.
- [39] D. Radojevic, "Fuzzy set theory in the Boolean frame," *International Journal of Computers Communications & Control*, vol. 3, no. 3, pp. 121–131, 2008.

- [40] D. Radojević, “Logical aggregation based on interpolative Boolean algebra,” *Mathware & Soft Computing*, vol. 15, no. 1, pp. 125–141, 2008.
- [41] D. Radojević, “Real sets as consistent Boolean generalization of classical sets,” in *From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence*, pp. 150–171, Editing House of Romanian Academy, Bucharest, Romania, 2008.
- [42] D. Radojević, “Real-valued realizations of boolean algebras are a natural frame for consistent fuzzy logic,” in *On Fuzziness*, vol. 299 of *Studies in Fuzziness and Soft Computing*, pp. 559–565, 2013.
- [43] A. Poledica, P. Milošević, I. Dragović, B. Petrović, and D. Radojević, “Modeling consensus using logic-based similarity measures,” *Soft Computing*, vol. 19, no. 11, pp. 3209–3219, 2015.
- [44] V. Dobrić, D. Kovačević, B. Petrović, D. Radojević, and P. Milošević, “Formalization of human categorization process using interpolative Boolean algebra,” *Mathematical Problems in Engineering*, vol. 2015, Article ID 620797, 8 pages, 2015.
- [45] P. Milošević, B. Petrović, D. Radojević, and D. Kovačević, “A software tool for uncertainty modeling using Interpolative Boolean algebra,” *Knowledge-Based Systems*, vol. 62, pp. 1–10, 2014.
- [46] J. Baumeister, “Advanced empirical testing,” *Knowledge-Based Systems*, vol. 24, no. 1, pp. 83–94, 2011.
- [47] A. T. Weeraratna, Y. Jiang, G. Hostetter et al., “Wnt5a signaling directly affects cell motility and invasion of metastatic melanoma,” *Cancer Cell*, vol. 1, no. 3, pp. 279–288, 2002.
- [48] K. Kobayashi and K. Hiraishi, “Verification and optimal control of context-sensitive probabilistic Boolean networks using model checking and polynomial optimization,” *The Scientific World Journal*, vol. 2014, Article ID 968341, 8 pages, 2014.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

