Flexible Generalized Fuzzy Petri Nets for Rule-Based Systems

Zbigniew Suraj[1(✉)], Piotr Grochowalski[1], and Sibasis Bandyopadhyay[2]

[1] Chair of Computer Science, University of Rzeszów, Rzeszów, Poland
zbigniew.suraj@ur.edu.pl

[2] Department of Mathematics, Visva Bharati, Santiniketan, India

Abstract. In 2015, the modified generalised fuzzy Petri nets (*mGFP*-nets) were proposed. This paper describes an extended class of *mGFP*-nets called flexible generalised fuzzy Petri nets (*FGFP*-nets). The main difference between the latter net model and the *mGFP*-net concerns transition operator Out_1 appearing in a triple of operators (In, Out_1, Out_2) in a *mGFP*-net. The operator Out_1 for each transition is determined automatically by the $GTVC$ algorithm, using the value of In and the value of truth degree function β in the net. This modification has significant influence on optimization of the modelled system by the *FGFP*-nets. The choice of suitable operators for the modelled system is very important, especially in systems described by incomplete, imprecise and/or vague information. The proposed approach can be used both for control design as well as knowledge representation and modelling of reasoning in decision support systems.

Keywords: Fuzzy petri net · Fuzzy logic · Knowledge representation · Approximate reasoning · Control · Decision support system

1 Introduction

Petri nets have become an important computational paradigm to represent and analyse a broad class of systems. As a computational paradigm for intelligent systems, they provide a graphical language to visualize, communicate and interpret engineering problems [5,12]. The concept of a Petri net has its origin in C.A. Petri's dissertation [13]. In the last four decades, several extensions of Petri nets have been proposed improving such aspects as hierarchical nets, high level nets or temporal nets [6]. For some time Petri nets have been gaining a growing interest among people in Artificial Intelligence and Systems Biology due to its adequacy to represent the reasoning process as a dynamic discrete event system [4,7,10,11]. In 1988, C.G. Looney proposed in [9] so called *fuzzy Petri nets* (*FP*-nets). In his model logical propositions can be associated with Petri nets allowing for logical reasoning about the modelled system. In this class of Petri net models not only crisp but also imprecise, vague and uncertain information is admissible and taken into account. Several authors proposed different classes of fuzzy Petri

C. Martín-Vide et al. (Eds.): TPNC 2016, LNCS 10071, pp. 196–207, 2016.
DOI: 10.1007/978-3-319-49001-4_16

nets [4]. These models are based on different approaches combining Petri nets and fuzzy sets introduced by L.A. Zadeh in 1965 [19]. Recently, a new class of *FP*-nets (*mGFP*-nets [16]) has been introduced. The main difference between this net model and the existing *FP*-nets [4] concerns the definition of the operator binding function δ. This function, similarly to generalised fuzzy Petri nets (*GFP*-nets) [18], connects transitions with triples of operators (In, Out_1, Out_2). The meaning of the first and third operator in the *mGFP*-nets is the same as in the case of *GFP*-nets. However, in the *mGFP*-net model, the meaning of the second operator in the triple (called transition operator) is significantly different. In the *GFP*-net model the operator Out_1 belongs to the class of t-norms, whereas in the *mGFP*-net model it is assumed that it belongs to the class of inverted fuzzy implications [15]. Due to this change the latter net model modifies existing interpretation of transition firing rule in *GFP*-nets. Since there exist infinitely numerous fuzzy implications in the field of fuzzy logic, and the nature of the marking changes variously in given *mGFP*-nets depending on used implication function, it is very difficult to select suitable implication functions for particular applications. However, taking into account the $GTVC$ algorithm for determining the optimal inverted fuzzy implication [15] for the transition operator Out_1 depended on the current marking of the net, the net model presented in the paper is more flexible than the *mGFP*-net one. The choice of suitable operators for the modelled system is very important, especially in control systems described by incomplete, imprecise and/or vague information.

The aim of this paper is to describe an extended class of *mGFP*-nets called *flexible generalised fuzzy Petri nets*. The main difference between the latter net model and the *mGFP*-net concerns the transition operator Out_1 appearing in a triple of operators (In, Out_1, Out_2) in a *mGFP*-net. The operator Out_1 for each transition is determined automatically by the $GTVC$ algorithm, using the value of In and the value of truth degree function β in the net. This modification has significant influence on optimization of the modelled system by the *FGFP*-nets. The proposed approach can be used both for control design as well as knowledge representation and modelling of approximate reasoning in decision support systems.

The text is organized as follows. In Sect. 2, basic notions and notation concerning triangular norms and fuzzy implications are recalled. Moreover, the algorithm for determining the optimal inverted fuzzy implication from a set of basic fuzzy implications is described. Section 3 provides a brief introduction to *mGFP*-nets. In Sect. 4, *FGFP*-nets formalism is presented. A simple example coming from the control domain is given in Sect. 5. Section 6 includes concluding remarks.

2 Preliminary Notions

In this section, we remind both basic notions and notation concerning triangular norms and fuzzy implications, as well as the $GTVC$ algorithm. For further details, see [2,8,15].

2.1 Triangular Norms

Let [0, 1] be the interval of real numbers from 0 to 1 (0 and 1 are included).

A function $t : [0,1] \times [0,1] \to [0,1]$ is said to be a t-norm if it satisfies, for all $a, b, c \in [0,1]$, the following conditions: (1) it has 1 as the unit element, (2) it is monotone, commutative, and associative. A function $s : [0,1] \times [0,1] \to [0,1]$ is said to be an s-norm if it satisfies, for all $a, b, c \in [0,1]$, the following conditions: (1) it has 0 as the unit element, (2) it is monotone, commutative, and associative.

More relevant examples of t-norms are the minimum $t(a,b) = min(a,b)$ and the algebraic product $t(a,b) = a * b$. However, the examples of s-norms are the maximum $s(a,b) = max(a,b)$ and the probabilistic sum $s(a,b) = a + b - a * b$.

The set of all triangular norms is denoted by TN.

2.2 Fuzzy Implications

A function $I : [0,1] \times [0,1] \to [0,1]$ is said to be a *fuzzy implication* if it satisfies, for all $x, x_1, x_2, y, y_1, y_2 \in [0,1]$, the following conditions: (1) $I(., y)$ is decreasing, (2) $I(x, .)$ is increasing, (3) $I(0,0) = 1$, $I(1,1) = 1$, and $I(1,0) = 0$.

The set of all fuzzy implications is denoted by FI.

Table 1 contains a sample of basic fuzzy implications. For their extended list, refer to ([2], page 4).

Table 1. A sample of basic fuzzy implications

Name	Year	Formula of basic fuzzy implication
Gödel	1932	$I_{GD}(x,y) = \begin{cases} 1 \; if \; x \leq y \\ y \; if \; x > y \end{cases}$
Goguen	1969	$I_{GG}(x,y) = \begin{cases} 1 \; if \; x \leq y \\ \frac{y}{x} \; if \; x > y \end{cases}$
Kleene-Dienes	1938	$I_{KD}(x,y) = max(1-x, y)$
Yager	1980	$I_{YG}(x,y) = \begin{cases} 1 \; if \; x = 0 \; and \; y = 0 \\ y^x \; if \; x > 0 \; or \; y > 0 \end{cases}$

In the paper [15], a method of choosing suitable fuzzy implications has been proposed. The method assumes that there is given a basic fuzzy implication $z = I(x,y)$, where x, y belong to [0,1]. x is interpreted as the truth value of the antecedent and is known, whereas z is interpreted as the truth value of the implication and is also known. In order to determine the truth value of the consequent y, the inverse function $InvI(x,z)$ is needed to be computed. Moreover, this method allows to compare two fuzzy implications. If the truth value of the antecedent and the truth value of the implication are given, we can easily optimize the truth value of the implication consequent by means of inverse fuzzy implications. In other words, one can choose the fuzzy implication which

Table 2. Inverted fuzzy implications for the fuzzy implications from Table 1

Formula of inverted fuzzy implication	Domain of inverted fuzzy implication
$InvI_{GD}(x,z) = z$	$0 \leq z < x, x \in (0,1]$
$InvI_{GG}(x,z) = x * z$	$0 \leq z < 1, x \in (0,1]$
$InvI_{KD}(x,z) = z$	$1 - x < z \leq 1, x \in (0,1]$
$InvI_{YG}(x,z) = z^{\frac{1}{x}}$	$0 \leq z \leq 1, x \in (0,1]$

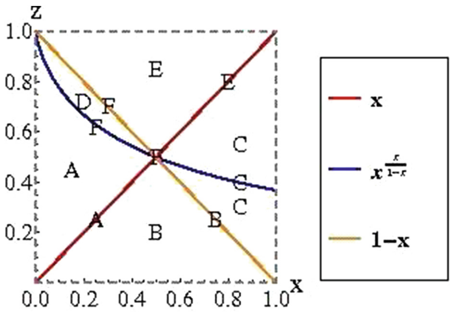

Fig. 1. The unit square $[0,1] \times [0,1]$ divided into 6 separable areas

has the greatest truth value of the implication consequent or greater truth value than other implication. Using this method we formulate a new model of fuzzy Petri nets proposed in this paper (see Sect. 4).

Table 2 lists inverse fuzzy implications and their domains for the fuzzy implications from Table 1. The resulting inverse functions can be compared with each other so that it is possible to order them. However, in general case, some of those functions are incomparable in the whole domain. Therefore, the domain is divided into separable areas within which this property is fulfilled.

The division of the unit square into 15 areas allows to compare the fuzzy implications with one another [15]. However, if we are only interested in finding the optimal implication which has the greatest truth value of the implication consequent, it is enough to divide the unit square into 6 areas A-F (see Fig. 1 and Table 3).

2.3 Algorithm

The algorithm presented below determines a basic fuzzy implication which has the greatest truth value of the consequent, and the truth value of the antecedent as well as the truth value of the implication are given. This algorithm uses the inverse fuzzy implications and their domains are presented in Table 2. The **if-then** clauses corresponding to specific cases are shown in Table 3.

Table 3. Table of optimal functions

No	Area	The optimal function
A	$z \geq x$ and $z < x^{\frac{x}{1-x}}$	I_{GG}
B	$z < x$ and $z \leq 1-x$	I_{GD}
C	$z > 1-x$ and $z < x$	$I_{GD} = I_{KD}$
D	$z > 1-x$ and $z \geq x$	I_{KD}
E	$z > x^{\frac{x}{1-x}}$ and $z \leq 1-x$	I_{YG}
F	$z = x^{\frac{x}{1-x}}$ and $x \in (0, \frac{1}{2}]$	$I_{GG} = I_{YG}$

Algorithm *GTVC*

Input: x - the truth value of the antecedent, z - the truth value of the implication
Output: I - fuzzy implication which has the greatest truth value of the consequent

if $(z \geq x$ and $z < x^{\frac{x}{1-x}})$ or $(z = x^{\frac{x}{1-x}}$ and $x \in (0, \frac{1}{2}])$ **then** $I \leftarrow I_{GG}$;
/* **case** A or F */
if $(z < x$ and $z \leq 1-x)$ or $(z > 1-x$ and $z < x)$ **then** $I \leftarrow I_{GD}$;
/* **case** B or C */
if $(z > 1-x$ and $z \geq x)$ **then** $I \leftarrow I_{KD}$;
/* **case** D */
if $(z > x^{\frac{x}{1-x}}$ and $z \leq 1-x)$ **then** $I \leftarrow I_{YG}$;
/* **case** E */
return I;

This algorithm is a compact version of the algorithm presented in [15].

3 Modified Generalised Fuzzy Petri Nets

In the paper, we assume that the reader is familiar with the basic notions of Petri nets [12].

Definition 1. *[16] A modified generalised fuzzy Petri net is said to be a tuple* $N = (P, T, S, I, O, \alpha, \beta, \gamma, Op, \delta, M_0)$, *where: (1)* $P = \{p_1, p_2, \ldots, p_n\}$ *is a finite set of places; (2)* $T = \{t_1, t_2, \ldots, t_m\}$ *is a finite set of transitions; (3)* $S = \{s_1, s_2, \ldots, s_n\}$ *is a finite set of statements; (4) the sets* P, T, S *are pairwise disjoint; (5)* $I\colon T \to 2^P$ *is the input function; (6)* $O\colon T \to 2^P$ *is the output function; (7)* $\alpha\colon P \to S$ *is the statement binding function; (8)* $\beta\colon T \to [0,1]$ *is the truth degree function; (9)* $\gamma\colon T \to [0,1]$ *is the threshold function; (10)* $Op = TN \cup FI$ *is a union of triangular norms and inverted fuzzy implications called the set of operators; (11)* $\delta\colon T \to Op \times Op \times Op$ *is the operator binding function; (12)* $M_0\colon P \to [0,1]$ *is the initial marking, and* 2^P *denotes a family of all subsets of the set* P.

As for the graphical interpretation, places are denoted by circles and transitions by rectangles. The function I describes the oriented arcs connecting places with transitions, and the function O describes the oriented arcs connecting transitions with places. If $I(t) = \{p\}$ then a place p is called an *input place* of a transition t, and if $O(t) = \{p'\}$, then a place p' is called an *output place* of t. The initial marking M_0 is an initial distribution of numbers in the places. It can be represented by a vector of dimension n of real numbers from $[0, 1]$. For $p \in P$, $M_0(p)$ can be interpreted as a truth value of the statement s bound with a given place p by means of the statement binding function α. Pictorially, the tokens are represented by means of grey "dots" together with suitable real numbers placed inside the circles corresponding to appropriate places.

We assume that if $M_0(p) = 0$ then the token does not exist in the place p. The numbers $\beta(t)$ and $\gamma(t)$ are placed in a net picture under the transition t. The first number is usually interpreted as the truth degree of an implication corresponding to a given transition t. The role of the second one is to limit the possibility of transition firings, i.e., if the input operator In value for all values corresponding to input places of the transition t is less than a threshold value $\gamma(t)$ then this transition cannot be fired (activated). The operator binding function δ connects similarly to *GFP*-nets [18] transitions with triples of operators (In, Out_1, Out_2). The meaning of the first and third operator is the same as in the case of *GFP*-nets. The first operator in the triple is called the input operator, and the third one is the output operator. The input operator In concerns the way in which all input places are connected with a given transition t. In the case of the input operator we assume that it can belong to one of two classes, i.e., t- or s-norms. However, the second operator in the triple (i.e., Out_1) is now called the transition operator and its meaning is significantly different. In the *GFP*-net model the operator Out_1 belongs to the class of t-norms, whereas in the *mGFP*-net we assume that it belongs to the class of inverted fuzzy implications. The transition operator Out_1 and the output operator Out_2 concern the way in which the next marking is computed after firing the transition t. In the case of the output operator we assume similarly to *GFP*-nets that it can belong to the class of s-norms.

Let N be a *mGFP*-net. A marking of N is a function $M: P \to [0, 1]$.

The *mGFP*-net dynamics defines how new markings are computed from the current marking when transitions are fired.

Let $N = (P, T, S, I, O, \alpha, \beta, \gamma, Op, \delta, M_0)$ be a *mGFP*-net, M be a marking of N, $t \in T$, $I(t) = \{p_{i1}, p_{i2}, \ldots, p_{ik}\}$ be a set of input places for a transition t and $\beta(t) \in (0, 1]$. Moreover, let $\delta(t) = (In, Out_1, Out_2)$ and D be the domain of a transition operator Out_1, i.e., the domain of an inverted fuzzy implication corresponding to the transition t.

A transition $t \in T$ is *enabled* for marking M, if the value of input operator In for all input places of the transition t by M is positive and greater than, or equal to, the value of threshold function γ corresponding to the transition t, and the value belongs to the domain of a transition operator Out_1 of t, i.e., the following two conditions must be satisfied:

1. $In(M(p_{i1}), M(p_{i2}), \ldots, M(p_{ik})) \geq \gamma(t) > 0$,
2. $In(M(p_{i1}), M(p_{i2}), \ldots, M(p_{ik})) \in D$.

Only enabled transitions can be fired. If M is a marking of N enabling transition t and M' is the marking derived from M by firing transition t, then for each $p \in P$ a procedure for computing the next marking $M^{'}$ is as follows:

1. Numbers in all output places of t are modified in the following way: at first, the value of input operator In for all input places of t is computed, next, the value of output operator Out_1 for the value of In and for the value of truth degree function $\beta(t)$ is determined, and finally, a value corresponding to $M'(p)$ for each $p \in O(p)$ is obtained as a result of output operator Out_2 for the value of Out_1 and the current marking $M(p)$.
2. Numbers in the remaining places of net N are not changed.

Formally, for each $p \in P$

$$M'(p) = \begin{cases} Out_2(Out_1(In(M(p_{i1}), M(p_{i2}), \ldots, M(p_{ik})), \beta(t)), M(p)) \\ \text{ if } p \in O(t), \\ M(p) \text{ otherwise.} \end{cases}$$

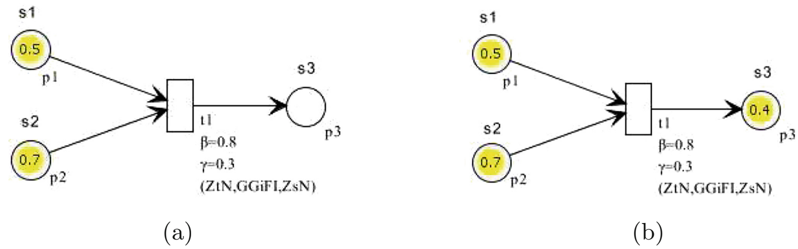

Fig. 2. A *mGFP*-net with: (a) the initial marking, (b) the marking after firing t_1

Example 2. Consider a $mGFP$-net in Fig. 2(a). For the net we have: the set of places $P = \{p_1, p_2, p_3\}$, the set of transitions $T = \{t_1\}$, the input function I and the output function O in the form: $I(t_1) = \{p_1, p_2\}$, $O(t_1) = \{p_3\}$, the set of statements $S = \{s_1, s_2, s_3\}$, the statement binding function α: $\alpha(p_1) = s_1$, $\alpha(p_2) = s_2$, $\alpha(p_3) = s_3$, the truth degree function β: $\beta(t_1) = 0.8$, the threshold function γ: $\gamma(t_1) = 0.3$, and the initial marking $M_0 = (0.5, 0.7, 0)$. Moreover, there are: the set of operators $Op = \{ZtN, ZsN\} \cup \{GGiFI\}$, where $ZtN(a, b) = min(a, b)$ (Zadeh t-Norm), $ZsN(a, b) = max(a, b)$ (Zadeh s-Norm), $GGiFI(x, z) = x * z$ (Goguen inverted Fuzzy Implication) (see Table 2) and the operator binding function δ defined as follows: $\delta(t_1) = (ZtN, GGiFI, ZsN)$. The transition t_1 is enabled by the initial marking M_0. Firing transition t_1 by the marking M_0 transforms M_0 to the marking $M^{'} = (0.5, 0.7, 0.4)$ (Fig. 2(b)).

For further details, see [16].

4 Flexible Generalised Fuzzy Petri Nets

This section presents the main contribution to the paper. Using the $GTVC$ algorithm from Subsect. 2.3 we reformulate the definition of *mGFP*-net as follows:

Definition 3. *A flexible generalised fuzzy Petri net is said to be a tuple* $N' = (P, T, S, I, O, \alpha, \beta, \gamma, Op, \delta, M_0)$, *where: (1)* $P, T, S, I, O, \alpha, \beta, \gamma, M_0$ *have the same meaning as in Definition 1; (2)* $Op = TN \cup OPTInvFI$ *is a union of triangular norms and optimal inverted fuzzy implications determined by the* $GTVC$ *algorithm (Subsect. 2.3) called the set of operators; (3)* $\delta\colon T \to Op \times Op \times Op$ *is the operator binding function.*

The operator binding function δ connects similarly to *mGFP*-nets transitions with triples of operators (In, Out_1, Out_2). The meaning of the first and third operator is the same as in the case of *mGFP*-nets. However, in this net model, the value of the second operator in the triple (i.e., Out_1) for each transition t in the *FGFP*-net is determined automatically by the $GTVC$ algorithm using the value of In and the value of truth degree function β.

It is easy to see that the role of a given transition t in the *FGFP*-net changes dependently on the current marking of the net. In the *mGFP*-net model we also assume that the operator belongs to the class of inverted fuzzy implications, but this operator is defined in advance by the users depending on their knowledge and experience.

In many cases, taking into account various combinations of the values of In and the value of truth degree function β for a given transition t, the choice of the suitable inverted fuzzy implication by the user is very difficult or sometimes impossible. However, considering the $GTVC$ algorithm (see Subsect. 2.3) the user can indicate the operator Out_1 as being chosen by the algorithm during the execution of the *FGFP*-net.

The dynamics of *FGFP*-nets is defined in an analogous way to the case of *mGFP*-nets.

Example 4. Consider an $FGFP$-net in Fig. 3(a). We assume that in the net: the sets P, T, S, and the functions I, O, α, β, γ, M_0 are described analogously to Example 1. However, the set of operators $Op = \{ZtN, ZsN\} \cup \{KDiFI\}$,

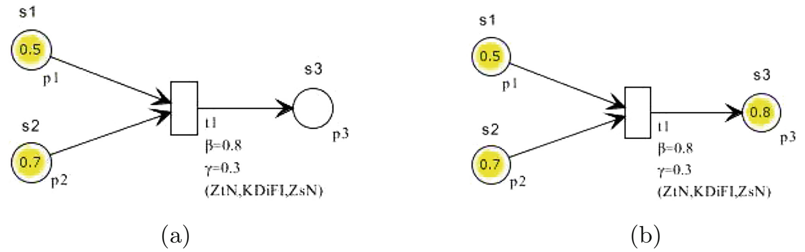

Fig. 3. A *FGFP*-net with: (a) the initial marking, (b) the marking after firing t_1

where $KDiFI(x, z) = z$ (Kleene-Dienes inverted Fuzzy Implication, see Table 2) denotes the optimal inverted fuzzy implication determined automatically by the $GTVC$ algorithm. The operator binding function δ is defined as follows: $\delta(t_1) = (ZtN, KDiFI, ZsN)$. The transition t_1 is enabled by the initial marking M_0. Firing transition t_1 by the marking M_0 transforms M_0 to the marking $M^{'} = (0.5, 0.7, 0.8)$ (Fig. 3(b)).

5 Example

In order to illustrate our methodology, let us describe a simple example coming from the domain of control. For this goal, we propose to consider the following production rules describing the rule controller for a technical plant: (1) IF s_2 THEN s_4, (2) IF s_1 AND s_4 THEN s_5, (3) IF s_3 AND s_4 THEN s_6, (4) IF s_5 AND s_6 THEN s_7, where: s_1 =‘Plant work is non-stable’, s_2 =‘Temperature sensor of plant indicates the temperature over 150 C degrees’, s_3 =‘Plant cooling does not work’, s_4 =‘Plant temperature is high’, s_5 =‘Plant is in failure state’, s_6 =‘Plant makes a huge hazard for environment’, s_7 =‘Turn off plant supply’.

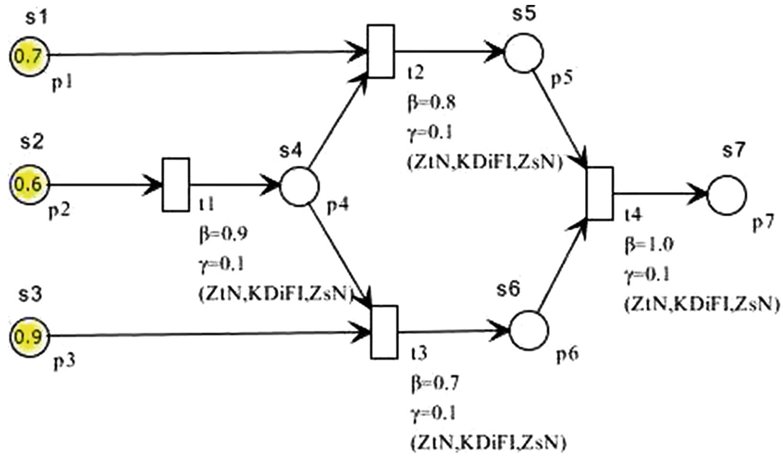

Fig. 4. An example of *FGFP*-net model of technical plant rule controller

In the further considerations we accept the assumptions as in Fig. 4, i.e., (1) the logical operator AND we interpret as *min* (Zadeh t-Norm); (2) to the statements s_1, s_2,..., s_7 we assign the fuzzy values 0.7, 0.6, 0.9, 0, 0, 0, 0, respectively; (3) the truth-values of transitions t_1, t_2, t_3, t_4 are equal to 0.9, 0.8, 0.7, 1.0, respectively; (4) all the threshold values for these four transitions are equal to 0.1; (5) the $GTVC$ algorithm (Subsect. 2.3) determines the transition operator Out_1 for all net transitions. Firstly, assessing the statements s_1, s_2, s_3, we see that the transition t_1 can be fired by the initial marking $M_0 = (0.7, 0.6, 0.9, 0, 0, 0, 0)$.

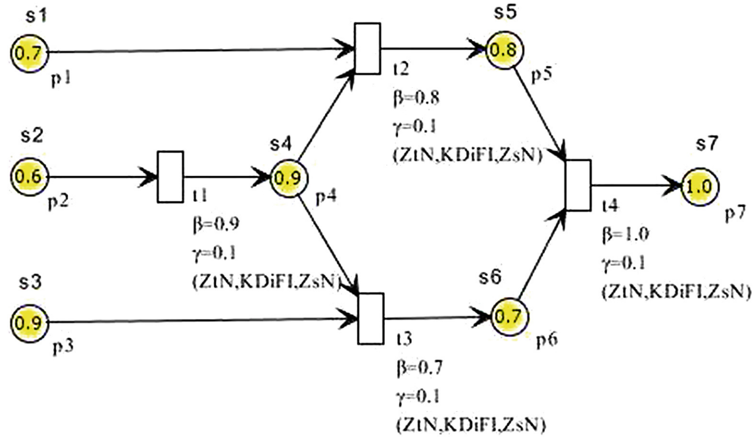

Fig. 5. *FGFP*-net model from Fig. 4 after simulation with *KDiFI* interpretation for the transition operator Out_1

The $GTVC$ algorithm determines the transition operator $Out_1 = KDiFI$ (the Kleene-Dienes inverted fuzzy implication, see Table 2) for the transition t_1 by M_0. If t_1 is fired then we obtain the new marking $M' = (0.7, 0.6, 0.9, 0.9, 0, 0, 0)$. Two new transitions t_2 and t_3 are enabled by M'. If one chooses a sequence of transitions t_2t_3 then they obtain the marking $M'' = (0.7, 0.6, 0.9, 0.9, 0.8, 0.7, 0)$. After firing the enabled transition t_4 the final value, corresponding to the statement s_7, equal to 1.0 is obtained (see Fig. 5). It is worth to observe that the same inverted fuzzy implication as for the transition t_1 is assigned by the $GTVC$ algorithm to the remaining transitions for all markings reachable from the initial marking M_0. Secondly, if we interpret these four transitions as the Goguen fuzzy implications, and if we choose the same sequences of transitions as above, we obtain the final value for the statement s_7 equal to 0.38. We omit the detailed computations performed in this case. Thirdly, if we execute the similar simulation of approximate reasoning for four transitions considered above and, if we interpret the transitions as the Yager fuzzy implications, we obtain the final value for s_7 equal to 1.0. It is easy to observe that the final value 1 for the statement s_7 is the greatest (optimal).

This example shows clearly that different interpretations of the transitions may lead to quite different decision results. It is also possible to see that using the $GTVC$ algorithm we automatically obtain the best interpretation for fuzzy implications. Certainly, it is limited to a set of considered fuzzy implications. In the paper this set consists of only four fuzzy implications presented in Table 1.

6 Concluding Remarks

In this paper, a flexible generalized fuzzy Petri net model has been proposed in fuzzy environment with inverted fuzzy implications having some benefits compared to those proposed in the literature which can be stated as follow:

1. This paper uses inverted fuzzy implications together with t-norms and as such opens an approach towards the optimization of the truth degree at the output places.
2. The generalized Petri net model with the inverted fuzzy implications has significant influence on optimization of the modelled system by the *FGFP*-nets.
3. The fuzzy Petri net model as proposed in this paper is more flexible and hence distinct to usual ones in the sense that it gives the option to define In and Out_2 operators manually, and the third operator Out_1 is determined automatically by the algorithm presented in Subsect. 2.3. The choice of suitable operators In, Out_1, and Out_2 for the modelled system is very important, especially in real systems described by incomplete, imprecise and/or vague information.

The flexibility of choosing the operators inspires for an extension with the weighted intuitionistic fuzzy sets [1] as proposed in [14]. When an weight is associated with an input or output values then it is concerned with the reliability of the information provided, leading to more generalization in approximate reasoning process in decision support system but also it often leads to computational complexity. The flexibility of choosing operators can minimize such computational complexity. It makes the model simple and thus speeds up the approximate reasoning process. Moreover, inverted fuzzy implication helps in optimization but the method discussed here also opens the choice of using other operators, e.g. t-norm operators, if the input values does not belong to the domain of inverted fuzzy implication functions. This is the novelty of this research work.

Using a simple real-life example suitability and usefulness of the proposed approach have been proved for the control design. The elaborated approach looks promising with regard to alike application problems that could be solved in a similar manner. It is worth to mention that an experimental application has been implemented in *Java*, consisting of an editor and a simulator. The editor allows inputting and editing the *FGFP*-nets, while the simulator starts with a given initial marking and executes enabled transitions visualising reached markings and simulation parameters. All figures and simulation results presented in the paper were produced by this application.

The following research problems of our concern are both the adaptation of the *FGFP*-nets in modelling the reasoning in decision support systems using intuitionistic fuzzy sets [1] as well as the use of this approach in Systems Biology [7].

Acknowledgments. This work was partially supported by the Center for Innovation and Transfer of Natural Sciences and Engineering Knowledge at the University of Rzeszów. The authors are grateful to the anonymous referees for their helpful comments.

References

1. Atanassov, K.: Intuitionistic fuzzy sets. Fuzzy Sets Syst. **20**, 87–96 (1986)
2. Baczyński, M., Jayaram, B.: Fuzzy Implications. Springer, Heidelberg (2008)
3. Bandyopadhyay, S., Suraj, Z., Grochowalski, P.: Modified generalized weighted fuzzy petri net in intuitionistic fuzzy environment. In: Flores, V. (ed.) IJCRS 2016. LNCS (LNAI), vol. 9920, pp. 342–351. Springer, Heidelberg (2016). doi:10.1007/978-3-319-47160-0_31
4. Cardoso, J., Camargo, H. (eds.): Fuzziness in Petri Nets. Springer, Heidelberg (1999)
5. Desel, J., Reisig, W., Rozenberg, G. (eds.): ACPN 2003. LNCS, vol. 3098. Springer, Heidelberg (2004)
6. Jensen, K., Rozenberg, G.: High-level Petri Nets. Springer, Heidelberg (1991)
7. Koch, I., Reisig, W., Schreiber, F. (eds.): Modeling in Systems Biology. The Petri Net Approach. Springer, Heidelberg (2011)
8. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Kluwer, Boston (2000)
9. Looney, C.G.: Fuzzy petri nets for rule-based decision-making. IEEE Trans. Syst., Man, Cybern. **18**(1), 178–183 (1988)
10. Pedrycz, W., Gomide, F.: A generalized fuzzy Petri net model. IEEE Trans. Fuzzy Syst. **2–4**, 295–301 (1994)
11. Pedrycz, W.: Generalized fuzzy Petri nets as pattern classifiers. Pattern Recog. Lett. **20–14**, 1489–1498 (1999)
12. Peterson, J.L.: Petri net theory and the modeling of systems. Prentice-Hall Inc., Englewood Cliffs (1981)
13. Petri, C.A.: Kommunikation mit Automaten. Schriften des IIM Nr. 2, Institut für Instrumentelle Mathematik, Bonn (1962)
14. Suraj, Z, Bandyopadhyay, S.: Generalized weighted fuzzy petri net in intuitionistic fuzzy environment. In: Proceedings the IEEE World Congress on Computational Intelligence, IEEE WCCI 2016, 25-29 July, 2016, Vancouver, Canada, pp. 2385–2392, IEEE (2016)
15. Suraj, Z., Lasek, A., Lasek, P.: Inverted fuzzy implications in approximate reasoning. Fundam. Informat. **143**, 151–171 (2015)
16. Suraj, Z.: Modified generalised fuzzy petri nets for rule-based systems. In: Yao, Y., Hu, Q., Yu, H., Grzymala-Busse, J.W. (eds.) RSFDGrC 2015. LNCS (LNAI), vol. 9437, pp. 196–206. Springer, Heidelberg (2015). doi:10.1007/978-3-319-25783-9_18
17. Suraj, Z., Lasek, A.: Toward optimization of approximate reasoning based on rule knowledge. In: Proceedings 2nd International Conference on Systems and Informatics, ICSAI 2014, November 15-17, 2014, Shanghai, China, IEEE Systems, Man and Cybernetics Society, IEEE Catalog Numbers: CFP1473R-CDR ISBN: 978-1-4799-5457-5, pp. 281–285 (2014)
18. Suraj, Z.: A new class of fuzzy Petri nets for knowledge representation and reasoning. Fundam. Inform. **128**(1–2), 193–207 (2013)
19. Zadeh, L.A.: Fuzzy sets. Inform. Control **8**, 338–353 (1965)