



Weighted Generalized Fuzzy Petri Nets and Rough Sets for Knowledge Representation and Reasoning

Zbigniew Suraj¹ , Aboul Ella Hassanien² , and Sibasis Bandyopadhyay³

¹ Institute of Computer Science, University of Rzeszów, Rzeszów, Poland
zbigniew.suraj@ur.edu.pl

² Faculty of Computers and Artificial Intelligence, Cairo University, Giza, Egypt
aboitcairo@gmail.com

³ Asansol, India

Abstract. In this paper, we consider the decision tables provided by experts in the field. We construct an algorithm for executing a highly parallel program represented by a fuzzy Petri net from a given decision table. The constructed net allows objects to be identified in decision tables to the extent that appropriate decisions can be made. Conditional attribute values given by experts are propagated by the net at maximum speed. This is done by properly organizing the net's work. Our approach is based on rough set theory and weighted generalized fuzzy Petri nets.

Keywords: Decision system · Information system · Rough set · Decision rule · Weighted generalized fuzzy Petri net

1 Introduction

Rough set theory, proposed by Pawlak in 1982 [18], is a mathematical tool for dealing with unclear, imprecise, incoherent and uncertain knowledge. It has been observed for many years that both research and applications of rough set theory are attracting more and more attention of researchers. It can be successfully used in many areas of application alone or in combination with other approaches. Here, we use this theory to support modeling of decision-making systems using weighted generalized fuzzy Petri nets.

In this paper, we assume that a decision table S representing experimental knowledge is given [17]. It consists of a number of rows labeled by elements from a set of objects U , which contain the results of measurements, observations, reviews etc. represented by a value vector of conditional attributes (conditions) from A together with a decision d corresponding to this vector. Values of conditions are provided by experts in the field. In some applications the values of conditional attributes can be interpreted as states of local processes in a complex system and the decision value is related to the global state of that system [13, 16, 24]. Sometimes it is necessary to transform a given experimental decision

table by taking into account other relevant features (new conditional attributes) instead of the original ones. This step is necessary when the decision algorithm constructed directly from the original decision table yields an inadequate classification of unseen objects or when the complexity of decision algorithm synthesis from the original decision table is too high. In this case some additional time is necessary to compute the values of new features after the original values are given. The input for our algorithm consists of a decision table (if necessary, pre-processed as described above).

We shall construct a fuzzy Petri net allowing to make a decision as soon as a sufficient number of conditional attribute values is known and conclusions drawn from the knowledge encoded in S (cf. [22]). In the paper we formulate this problem and present its solution.

First, we assume that knowledge encoded in S is represented by rules automatically extracted from S . We consider acceptable rules in S , i.e. rules for which the accuracy factor need not necessarily be equal to 1 [23]. We assume that the knowledge encoded in S is complete in the sense that invisible objects have attribute value vectors consistent with rules extracted from S . This assumption may be too restrictive, because the rules for the classification of new objects should be generated only from appropriate features (attributes). The rule is active if the values of all attributes on its left side are given. Our algorithm should propagate information from attributes to other attributes as soon as possible. This is the reason for generating true decision rules corresponding to relative reducts with respect to the decision in S [22]. The last step of our algorithm is the implementation of the set of generated rules using fuzzy Petri nets. Each step of a computation of the constructed fuzzy Petri net consists of two phases. In the first phase, it is checked that all condition values are known, and if so, in the second phase, new information about the values is sent through the net at maximum speed. The whole computation process is carried out by proper organization of the net's work.

In the paper, we use fuzzy Petri nets [3, 5, 10, 12, 30] as a model of the target decision-making system. Net properties can be verified using tools for the analysis of Petri nets (see e.g. [28]).

Over the past few decades, there has been a series of modifications to the classic fuzzy Petri nets (FPNs) [12] to deal with complex decision-making systems. Chen [4] introduced weight factors into FPNs and proposed a weighted FPN (WFPN) model. Ha *et al.* [7] extended his work by adding input and output weight factors into WFPNs. Then the intuitionistic fuzzy sets were integrated into FPNs, and an intuitionistic FPN was presented in [11, 26]. Skowron and Suraj [23] developed a parallel algorithm for real-time decision-making based on rough set theory and classic Petri nets. Peters *et al.* [20] combined the theory of FPNs, rough sets, and colored Petri nets to develop a rough fuzzy Petri net model. Suraj and Fryc [27] introduced time factor to approximate Petri nets, which plays a vital role in developing real-time decision-making systems. Bandyopadhyay *et al.* [1] proposed to link Petri nets and soft sets and introduced a soft Petri net model. Suraj and Hassanien [29] combined the theory of FPN and sets

of fuzzy intervals to avoid the problem of determining the exact membership or truth value.

This paper establishes some relationships between rough set theory and fuzzy Petri nets. Parameter values such as rule certainty coefficients, input and output weights of arcs in the net model are calculated automatically from a given decision table. The empirical example provided here shows the effectiveness of the proposed model.

The rest of this paper is organized in the following way. Section 2 contains some background knowledge regarding rough set theory. In Sect. 3, the weighted generalized fuzzy Petri net formalism is given. Section 4 describes three structural forms of decision rules and a method for transformation of decision tables into weighted generalized fuzzy Petri nets. An example illustrating the approach presented in this paper is provided in Sect. 5. Finally, Sect. 6 suggests some directions for further research related to our approach.

2 Preliminaries of Rough Set Theory

In this section we recall basic notions of rough set theory. Among them are those of information systems, indiscernibility relations, dependencies of attributes, relative reducts, significance of attributes and rules [14, 15].

2.1 Information Systems and Decision Systems

An *information system* is a pair $S = (U, A)$, where U is a non-empty finite set of *objects* called the *universe* and A is a non-empty finite set of *attributes* such that $a : U \rightarrow V_a$ for every $a \in A$. The set V_a is called the *value set* of a , and $V = \bigcup_{a \in A} V_a$ is said to be the *domain* of A .

Let $S = (U, A)$ be an information system and let $B \subseteq A$ and $X \subseteq U$. Then there is associated an equivalence relation $\text{ind}(B) : \text{ind}(B) = \{(u, u') \in U \times U : \text{for every } a \in B \ a(u) = a(u')\}$. $\text{ind}(B)$ is called the *B-indiscernibility relation*. If $(u, u') \in \text{ind}(B)$, then objects u and u' are indiscernible from each other by attributes from B . The equivalence classes of the *B-indiscernibility relation* are denoted $[u]_B$.

We can approximate X using only the information contained in B , constructing the *B-lower* and *B-upper approximations* of X , denoted by $\underline{B}X$ and $\overline{B}X$ respectively, where $\underline{B}X = \{u : [u]_B \subseteq X\}$ and $\overline{B}X = \{u : [u]_B \cap X \neq \emptyset\}$. The objects in $\underline{B}X$ can be with certainty classified as members of X on the basis of knowledge in B , while the objects in $\overline{B}X$ can be only classified as possible members of X on the basis of knowledge in B . The set X is *rough* if $\overline{B}X - \underline{B}X \neq \emptyset$.

A *decision system* (a *decision table*) is any information system of the form $S = (U, A \cup \{d\})$, where $d \notin A$ is a distinguished attribute called *decision attribute* (*decision*). The elements of A are called *conditional attributes* (*conditions*).

Let $S = (U, A \cup \{d\})$ be a decision system. The cardinality of the image $d(U) = \{k : d(u) = k \text{ for some } u \in U\}$ is called the *rank* of d and is denoted by $r(d)$. We assume that the set V_d of values of the decision d is equal to $\{1, \dots, r(d)\}$.

Let us observe that the decision d determines a partition $\{X_1, \dots, X_{r(d)}\}$ of the universe U , where $X_k = \{u \in U : d(u) = k\}$ for $1 \leq k \leq r(d)$. The set X_i is called the i -th decision class of S . If $X_1, \dots, X_{r(d)}$ are the decision classes of S , then the set $\underline{B}X_1 \cup \dots \cup \underline{B}X_{r(d)}$ is called the B -positive region of S and is denoted by $POS_B(d)$.

Any decision system $S = (U, A \cup \{d\})$ can be represented by a data table with the number of rows equal to the cardinality of the universe U and the number of columns equal to the cardinality of the set $A \cup \{d\}$. On the position corresponding to the row u and column a the value $a(u)$ appears.

Example 1. A small decision system is shown in Table 1. We have a set of objects (patients) $U = \{1, 2, 3, 4, 5, 6\}$, a set of conditional attributes (symptoms) $A = \{H \text{ (Headache), M (Muscle-pain), T (Temperature)}\}$. The decision attribute is denoted by F (Flu). The possible values of attributes from $A \cup \{F\}$ are equal to no, yes, normal, high, or very high and $r(F) = 2$. The decision F defines a partition $\{X_1, X_2\}$ of U , where $X_1 = \{1, 2, 3, 6\}$, $X_2 = \{4, 5\}$. Each row of the table can be seen as information about specific patient.

Table 1. An example of a decision system

$U/A \cup \{d\}$	H	M	T	F
1	no	yes	high	yes
2	yes	no	high	yes
3	yes	yes	very high	yes
4	no	yes	normal	no
5	yes	no	high	no
6	no	yes	very high	yes

2.2 Dependency of Attributes

An important issue in data analysis is discovering of dependencies between attributes. Intuitively, a set of attributes C depends totally on a set of attributes B , denoted by $B \Rightarrow C$, if there exists a functional dependency between values of C and B .

Let $S = (U, A)$ be an information system and let $B, C \subseteq A$.

We say that the set C depends on B in degree k ($0 \leq k \leq 1$), denoted by $B \Rightarrow_k C$, if $k = \gamma(B, C) = \frac{|POS_B(C)|}{|U|}$, where $POS_B(C) = \bigcup_{X \in U/C} \underline{B}(X)$ and $|X|$ denotes the cardinality of $X \neq \emptyset$. The set $POS_B(C)$ is called a positive region of the partition U/C with respect to B . In fact, it is the set of all elements of U that can be uniquely classified to blocks of the partition U/C by means of B .

Let $B, C \subseteq A$, and $B' \subseteq B$. A set B' is a C -reduct of B (or B' is a relative reduct of B with respect to C), if B' is a minimal subset of B and $\gamma(B, C) = \gamma(B', C)$.

Example 2. Consider once again the decision system presented in Table 1. For example, for the dependency $\{H, M, T\} \Rightarrow_k \{F\}$ we get $k = 2/3$. However, for the dependency $\{T\} \Rightarrow_k \{F\}$, we get $k = 1/2$. The attribute T offers a worse classification than the entire set of attributes H, M, T. It is worth to noting that neither H nor M can be used to recognize flu, because for both dependencies $\{H\} \Rightarrow_k \{F\}$ and $\{M\} \Rightarrow_k \{F\}$ we have $k = 0$. In Table 1 there are two relative reducts with respect to $\{F\}$, $R_1 = \{H, T\}$ and $R_2 = \{M, T\}$ of the set of conditions $\{H, M, T\}$.

2.3 Significance of Attributes

Significance of an attribute a in a decision system $S = (U, A \cup \{d\})$ can be evaluated by measuring the effect of removing of an attribute $a \in A$ from the attribute set A on the positive region defined by the table S .

Let $B \subseteq A$. Significance of an attribute $a \in A$ is defined as follows: $\sigma(B, d, a) = \gamma(B, \{d\}) - \gamma(B - \{a\}, \{d\}) = \frac{|POS_B(\{d\})| - |POS_{B-\{a\}}(\{d\})|}{|U|}$, and is simply denoted by $\sigma(a)$ when B and $\{d\}$ are understood.

This numerical factor measures the difference between $\gamma(B, \{d\})$ and $\gamma(B - \{a\}, \{d\})$, i.e. it says how the factor $\gamma(B, \{d\})$ changes when an attribute a is removed.

Note that the following relationship is also met: $0 \leq \sigma(B, d, a) \leq 1$.

Example 3. Using the above formula for the decision system from Example 1, we obtain the following results for Table 1:

1. For the set of conditional attributes A : $\sigma(H) = 0, \sigma(M) = 0, \sigma(T) = 1/2$
2. For the relative reduct R_1 : $\sigma(H) = 1/6, \sigma(T) = 2/3$
3. For the relative reduct R_2 : $\sigma(M) = 0, \sigma(T) = 3/4$

2.4 Rules in Decision Systems

Rules express some of the relationships between values of the attributes described in decision tables. In this subsection we recall the definition of rules as well as other related concepts.

Let $S = (U, A \cup \{d\})$ be a decision system, $B \subseteq A \cup \{d\}$, and $V = \bigcup_{a \in A} V_a \cup V_d$.

Atomic formulae over B and V are expressions of the form $a = v$. They are called *descriptors* over B and V , where $a \in B$ and $v \in V_a$. The set $DESC(B, V)$ of formulae over B and V is the least set containing all atomic formulae over B and V and closed with respect to the propositional connectives OR (disjunction), AND (conjunction) and NOT (negation).

Let $\tau \in DESC(B, V)$. $\|\tau_S\|$ denotes the meaning of τ in the decision system S which is the set of all objects in U with the property τ . These sets are defined as follows:

1. if τ is of the form $a = v$ then $\|\tau_S\| = \{u \in U : a(u) = v\}$
2. $\|(\tau \text{ OR } \tau')_S\| = \|\tau_S\| \cup \|\tau'_S\|$; $\|(\tau \text{ AND } \tau')_S\| = \|\tau_S\| \cap \|\tau'_S\|$; $\|\text{NOT } \tau_S\| = U - \|\tau_S\|$.

The set $\text{DESC}(A, V_a)$, $a \in A$, is called the set of *conditional formulae* of S .

A *decision rule* r for S is any expression of the form IF τ THEN $d = v$, where $\tau \in \text{DESC}(A, V_a)$, $v \in V_d$ and $\|\tau_S\| \neq \emptyset$. Formulae τ and $d = v$ are called the *predecessor* and the *successor* of the decision rule r . $\|\tau_S\|$ is the non-empty set of objects *matching* the decision rule and $\|\tau_S\| \cap \|(d = v)_S\|$ is the set of objects *supporting* the rule. With every decision rule r we can associate several numerical factors. The *accuracy factor* of the decision rule r is the number $acc(r) = \frac{\|\|\tau_S\| \cap \|(d=v)_S\|\|}{\|\|\tau_S\|\|}$, while the *strength factor* of the decision rule r is understood as $str(r) = \frac{\|\|\tau_S\| \cap \|(d=v)_S\|\|}{|U|}$. The decision rule r is *true* in S , if $acc(r) = 1$, otherwise it is *acceptable* in S .

It is also easy to see that $0 \leq str(r) \leq acc(r) \leq 1$ for every the decision rule r in S .

Example 4. Let us consider the decision system table S from Example 1 presented in Table 1. Using the method for generating decision rules in S [22], we get the following rules, corresponding to the relative reduct $R_1 = \{H, T\}$ along with the numerical factors defined above:

- r_1 : IF H=no AND T=very high THEN F=yes; $str(r_1) = 1/6$, $acc(r_1) = 1$
- r_2 : IF H=yes AND T=very high THEN F=yes; $str(r_2) = 1/6$, $acc(r_2) = 1$
- r_3 : IF H=no AND T=high THEN F=yes; $str(r_3) = 1/6$, $acc(r_3) = 1$
- r_4 : IF H=yes AND T=high THEN F=yes; $str(r_4) = 1/6$, $acc(r_4) = 1/2$
- r_5 : IF H=yes AND T=high THEN F=no; $str(r_5) = 1/6$, $acc(r_5) = 1/2$
- r_6 : IF H=no AND T=normal THEN F=no; $str(r_6) = 1/6$, $acc(r_6) = 1$

Note that the rules r_1, r_2, r_3, r_6 are true in Table 1, while the other rules are acceptable in this table.

For a systematic overview of rule synthesis, see e.g. [9, 15, 21].

3 Weighted Generalized Fuzzy Petri Nets

Fuzzy Petri nets are a modification of classic Petri nets to deal with imprecise, unclear or incomplete information in knowledge-based systems that are widely used to model fuzzy production rules and rule-based reasoning.

In this section, we define weighted generalized fuzzy Petri nets (WGFP-net). The new model is a modification of generalized fuzzy Petri nets, proposed in [25]. The main difference between the current net model and the previous one concerns the weights of arcs. Weights are now added to the input and output arcs. They are any numbers from 0 to 1, automatically calculated from the data table and interpreted in the concepts of rough set theory (see Sect. 4) (cf. [2, 10]).

In this paper WGFP-nets are used as a tool for computing a parallel program from a given decision table. After modeling a decision table by a WGFP-net the states are identified in the net to an extent allowing to take the appropriate decisions.

We also assume that the reader knows the basic concepts of classic Petri nets [6] and triangular norms [8].

Let $[0, 1]$ denotes the set of real numbers between 0 and 1.

A *weighted generalized fuzzy Petri net* is a tuple $N = (P, T, I, O, M_0, S, \alpha, \beta, \gamma, Op, \delta)$, where: (1) $P = \{p_1, p_2, \dots, p_n\}$ is a finite set of places; (2) $T = \{t_1, t_2, \dots, t_m\}$ is a finite set of transitions; (3) $I: P \times T \rightarrow [0, 1]$ is the input function that maps directed arcs from places to output transitions of those places. If a directed arc (p, t) exists between a place p and a transition t , then $I(p, t) > 0$, otherwise 0. The values of $I(p, t)$ for $(p, t) \in P \times T$ are called input weights of transitions t and are denoted by iw ; (4) $O: T \times P \rightarrow [0, 1]$ is the output function that maps directed arcs from transitions to output places of those transitions. If a directed arc (t, p) exists between a transition t and a place p , then $O(t, p) > 0$, otherwise 0. The values of $O(t, p)$ for $(t, p) \in T \times P$ are called output weights of transitions t and are denoted by ow ; (5) $M_0: P \rightarrow [0, 1]$ is the initial marking; (6) $S = \{s_1, s_2, \dots, s_n\}$ is a finite set of statements; (7) $\alpha: P \rightarrow S$ is the statement binding function; (8) $\beta: T \rightarrow [0, 1]$ is the truth degree function; (9) $\gamma: T \rightarrow [0, 1]$ is the threshold function; (10) Op is a union of t-norms and s-norms called the set of operators, and the sets P, T, S, Op are pairwise disjoint; (11) $\delta: T \rightarrow Op \times Op \times Op$ is the operator binding function.

We also accept that if $I(p, t) = 0$ ($O(p, t) = 0$) then the directed arc from input (output) place p to transition t does not exist in the net drawing. Similarly, if $M_0(p) = 0$ then the token does not exist in the place p . In addition, if $I(p, t) = 1$ ($O(t, p) = 1$), then the weight of the arc equal to 1 is also disregarded in the net drawing. The numbers $\beta(t)$ and $\gamma(t)$ are placed in a net picture under the transition t . The first number is 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 transitions with triples of operators (In, Out_1, Out_2) . The first operator in the triple is called the input operator, and two remaining ones are the output operators. The input operator In concerns the way in which all input places are connected with a given transition t (more precisely, statements corresponding to those places). However, the output operators Out_1 and Out_2 concern the way in which the next marking is computed after firing the 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-norm, whereas the second one belongs to the class of t-norms and the third to the class of s-norms.

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

The dynamic behavior of the system is represented by the *firing* of the corresponding transition, and the evolution of the system is represented by a *firing*

sequence of transitions. We assume that the networks built in the form presented in this paper operate according to the *firing rule* consisting of the following three steps:

1. A transition $t \in T$ is *enabled* (or *ready for firing*) for marking M if the number produced by input operator In for all input places of the transition t by M multiplied by the relevant weights of arcs is positive and greater than, or equal to the number being a value of threshold function γ corresponding to the transition t . Formally, the following condition for $\gamma(t)$ should be satisfied: $In(iw_{i1} \cdot M(p_{i1}), iw_{i2} \cdot M(p_{i2}), \dots, iw_{ik} \cdot M(p_{ik})) \geq \gamma(t) > 0$, where In is an input operator of the transition t , iw_{ij} is an input weight of t and $M(p_{ij})$ is a marking of a place p_{ij} for $j = 1, 2, \dots, k$.
2. A transition can fire only if it is enabled.
3. 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) Tokens 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 multiplied by the weight ow and the current marking $M(p)$. (2) Tokens in the remaining places of net N are not changed.

Formally, for each $p \in P$

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

We also assume that if several transitions are simultaneously enabled in the same marking (i.e. transitions are *concurrent*) then they can be fired by an application of the firing rule described above in one and the same step and the resulting marking is computed according to this rule.

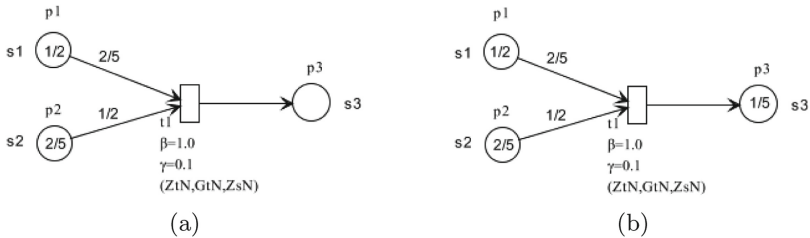


Fig. 1. A WGFP-net with the initial marking: (a) before firing t_1 , (b) after firing t_1

Example 5. Consider a WGFP-net in Fig. 1. 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: $I(p_1, t_1) = iw_1 = 2/5$, $I(p_2, t_1) = iw_2 = 1/2$, $I(p_3, t_1) = iw_3 = 0$, $O(t_1, p_1) = ow_1 = 0$, $O(t_1, p_2) = ow_2 = 0$, $O(t_1, p_3) = ow_3 = 1$, and the initial marking $M_0 = (1/2, 2/5, 0)$, 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 : $\beta(t_1) = 1.0$, the threshold function γ : $\gamma(t_1) = 0.1$, the set of operators $Op = \{ZtN, GtN, ZsN\}$, the operator binding function δ : $\delta(t_1) = (ZtN, GtN, ZsN)$, where $ZtN(a, b) = \min(a, b)$ (minimum, Zadeh t-Norm), $GtN(a, b) = a \cdot b$ (algebraic product, Goguen t-Norm), and $ZsN(a, b) = \max(a, b)$ (maximum, Zadeh s-Norm). The transition t_1 is enabled by the initial marking M_0 , since $ZtN(I(p_1, t_1) \cdot M_0(p_1), I(p_2, t_1) \cdot M_0(p_2)) = \min(1/5, 1/5) = 1/5 \geq 0.1 = \gamma(t_1)$. Firing transition t_1 by the marking M_0 transforms M_0 to the resulting marking $M' = (1/2, 2/5, 1/5)$, because $ow_3 \cdot GtN(1/5, \beta(t_1)) = 1 \cdot GtN(1/5, 1.0) = 1/5$ and $ZsN(1/5, M_0(p_3)) = \max(1/5, 0) = 1/5$. Note that in this case the transition t_1 is still enabled by M' , but when it is fired at this marking, the result marking is the same as M' . We omit the detailed description of the relevant calculations illustrating the transformation from the marking M' to M' after firing t_1 . They run similarly to these above.

4 Transformation of Decision Systems into WGFP-nets

Now we present a method for transforming decision rules representing a given decision system into a WGFP-net.

We assume that a decision system is represented by decision rules of the form IF τ THEN $d = v$.

Let $S = (U, A \cup \{d\})$ be a decision system, and $DESC(A, V_a)$ be the set of the set of conditional formulae of S .

In the paper, we consider three structural forms of decision rules with a list of numerical factors enclosed in square brackets '[' and ']' characterizing these rules (cf. [4, 7, 10]).

Type 1: A simple decision rule

$$r_1 : \text{ IF } a = v \text{ THEN } d = v' \\ [b; \sigma(a), str(r_1); acc(r_1)]$$

where $a = v$ and $d = v'$ denote descriptors such that $a = v \in DESC(A, V_a)$ and $v' \in V_d$, b is the truth degree value of $a = v$, $\sigma(a)$ is significance of the attribute a , while $str(r_1)$ and $acc(r_1)$ are the strength factor and the accuracy factor of the rule r_1 , respectively.

A WGFP-net structure of the decision rule r_1 is shown in Fig. 2, where iw is the input weight of the transition r_1 and interpreted as $\sigma(a)$, while ow is the output weight of r_1 and interpreted as $str(r_1)$ (see Subsect. 2.3 and 2.4). A larger value of i_w or ow means a stronger corresponding connection. However, the value $\beta(r_1) = c$ is interpreted as $acc(r_1)$. Similarly as before, the larger value of β the

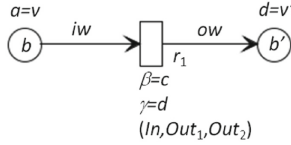


Fig. 2. A WGFP-net representation of the rule of type 1

more credible the rule is. The value of γ represents the threshold value. Larger value b requires greater truth degree of the rule precedence, i.e., $a = v$. The operator In and the operators Out_1, Out_2 represent the input operator and the output operators, respectively. According to Fig. 3 the token value in an output place p' of a transition t corresponding to the decision rule r_1 is calculated as $b' = ow \cdot Out_1(b \cdot iw, c)$, if $b \cdot iw \geq d$, where $d = \gamma(r_1)$ and $\gamma(r_1)$ is the threshold value associated to the transition r_1 and it is given by an expert in the field during the simulation process of the network.

If the predecessor or the successor of a decision rule contains AND or OR (propositional connectives), it is called a *composite decision rule*. Below, two types of composite decision rules are presented together with their WGFP-net representation (see Fig. 3 and Fig. 4).

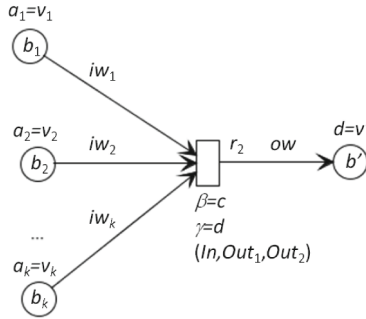


Fig. 3. A WGFP-net representation of the rule of type 2

Type 2: A composite conjunctive decision rule in the predecessor of the rule

$$r_2 : \text{IF } a_1 = v_1 \text{ AND } a_2 = v_2 \cdots \text{ AND } a_k = v_k \text{ THEN } d = v'$$

$$[b_1, b_2, \dots, b_k; \sigma^1(a), \sigma^2(a), \dots, \sigma^k(a), str(r_2); acc(r_2)]$$

where $a_1 = v_1, a_2 = v_2, \dots, a_k = v_k, d = v'$ denotes descriptors, and b_1, b_2, \dots, b_k, b' their truth degree values, respectively. The meaning of all numerical factors characterizing this rule is similar to the meanings of the relevant factors described for the rule of type 1. The token value b' is calculated in the output

place as follows (Fig. 3): $b' = Out_1(In(b_1 \cdot iw_1, b_2 \cdot iw_2, \dots, b_k \cdot iw_k), c) \cdot ow$, if $In(b_1 \cdot iw_1, b_2 \cdot iw_2, \dots, b_k \cdot iw_k) \geq d$, where $d = \gamma(r_2)$.

Type 3: A composite disjunctive decision rule in the successor of the rule

$$r_3 : \text{IF } a' = v' \text{ THEN } d = v_1 \text{ OR } d = v_2 \cdots \text{ OR } d = v_n$$

$$[b'; \sigma^1(a'), \sigma^2(a'), \dots, \sigma^n(a'), str^1(r_3), str^2(r_3), \dots, str^n(r_3); acc^1(r_3), acc^2(r_3), \dots, acc^n(r_3)]$$

where $a' = v'$, $d = v_1, d = v_2, \dots, d = v_n$ denotes descriptors, and b' is the truth degree value of $a' = v'$. The token value for the type 3 is calculated in each output place as follows (Fig. 4): $b_j = ow_j \cdot Out_1(b' \cdot iw, c_j)$, if $b' \cdot iw \geq d_j$, where $d_j = \gamma^j(r_3), j = 1, \dots, n$.

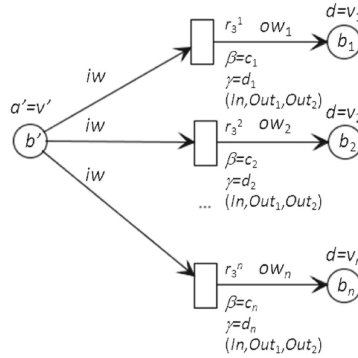


Fig. 4. A WGFP-net representation of the rule of type 3

Remarks:

1. It is easy to see that the rule of type 1 is a particular case of the rule of type 2, as in the case of the rule of type 1, there is only one descriptor in the predecessor. Type 3 can also be easily converted to type 1. Therefore, without losing generality, we can only consider the rules of type 1 and 2.
2. As the rules of type 1 and 3 have only one descriptor in their predecessors, we may omit the input operator In in Fig. 2 and 4. Nevertheless, for better readability of these figures we leave the operator where it is. What's more, the rule of type 3 can be generalized in the case when in the predecessor of the rule instead of one descriptor we have a conjunction of descriptors (as in the rule of type 2). Then the net modeling of such a rule in relation to its predecessor is similar to the one done for the rule of type 2.
3. We assume that the initial markings of output places are equal to 0 in all net models corresponding to the considered rule types. Therefore, in the descriptions of the token values in output places we do not regard the output operator Out_2 . In the opposite case, i.e., for non-zero markings of output places, we should take into account this output operator. Thus, in each formula presented above the final token value a' should be computed as follows:

$b' = Out_2(b'', M(p'))$, where b'' denotes the token values computed for suitable rule types by means of formulas presented above, and $M(p')$ is a marking of output place p' . Intuitively, a final token value corresponding to $M'(p')$ for each output place p' of a transition representing a decision rule r is obtained as a result of Out_2 operation for the computed Out_1 operation value and the current marking $M(p')$.

Using the method described above, we can formulate a simple algorithm that constructs a WGFP-net based on a given set of rules extracted from a decision system S . This algorithm transforms the rule into a WGFP-net depending on the form of the transformed rule.

Let $S = (U, A \cup \{d\})$ be a decision system.

Algorithm 1: Construction of WGFP-net using a set of decision rules in S

Input : A finite set R of decision rules in with a list of parameters
Output: A WGFP-net N_S
 $F \leftarrow \emptyset$; (* The empty set. *)
for each $r \in R$
if r is a rule of type 1 **then**
 \perp construct a subnet N_r as shown in Fig. 2;
if r is a rule of type 2 **then**
 \perp construct a subnet N_r as shown in Fig. 3;
if r is a rule of type 3 **then**
 \perp construct a subnet N_r as shown in Fig. 4;
 $F \leftarrow F \cup \{N_r\}$;
integrate all subnets from a family F on joint places and create a result net N_S ;
return N_S ;

5 An Example

To illustrate our methodology, let's reconsider the decision rules corresponding to the relative reduct R_1 from Example 4 along with a full list of parameters needed to build a structure of WGFP net model:

- r_1 : IF H=no AND T=very high THEN F=yes [$\sigma(H) = 1/6, \sigma(T) = 2/3, str(r_1) = 1/6; acc(r_1) = 1$]
- r_2 : IF H=yes AND T=very high THEN F=yes [$\sigma(H) = 1/6, \sigma(T) = 2/3, str(r_2) = 1/6; acc(r_2) = 1$]
- r_3 : IF H=no AND T=high THEN F=yes [$\sigma(H) = 1/6, \sigma(T) = 2/3, str(r_3) = 1/6; acc(r_3) = 1$]
- r_4 : IF H=yes AND T=high THEN F=yes [$\sigma(H) = 1/6, \sigma(T) = 2/3, str(r_4) = 1/6; acc(r_4) = 1/2$]

- r_5 : IF H=yes AND T=high THEN F=no [$\sigma(H) = 1/6, \sigma(T) = 2/3, str(r_5) = 1/6; acc(r_5) = 1/2$]
- r_6 : IF H=no AND T=normal THEN F=no [$\sigma(H) = 1/6, \sigma(T) = 2/3, str(r_6) = 1/6; acc(r_6) = 1$]

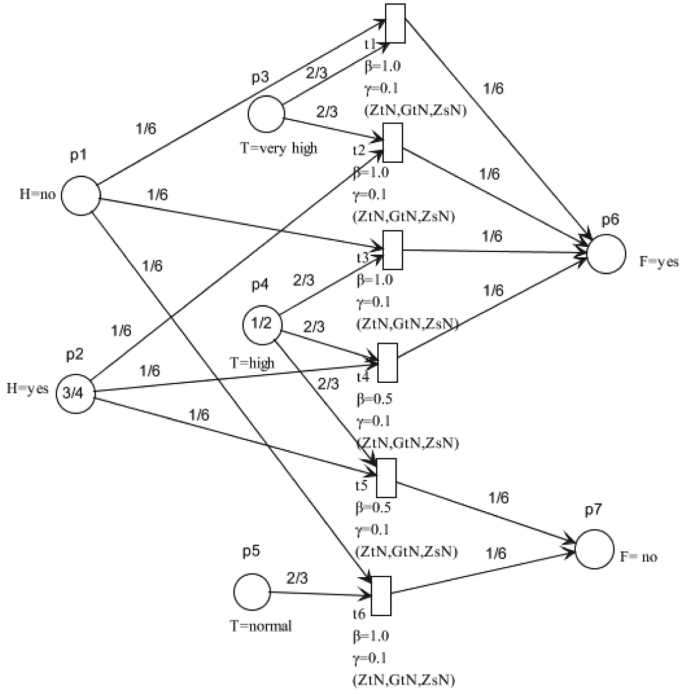


Fig. 5. An example of the WGFP-net model for the diagnosis of flu diseases with the initial marking

Using Algorithm 1 (Sect. 4) for constructing a WGFP-net on the base of a given set of rules, we present the WGFP-net model corresponding to these rules. This net model is shown in Fig. 5. Note that the places p_2 and p_4 include the truth degree values $3/4$ and $1/2$ corresponding to the descriptors H=yes and T=high, respectively. The remaining places on the net model are empty. In this example, input weights iw attached to arcs belong to the interval $[0,1]$ and are shown in Fig. 5. Moreover, there are: the truth degree function β : $\beta(t_1) = \beta(t_2) = \beta(t_3) = \beta(t_6) = 1.0$ and $\beta(t_4) = \beta(t_5) = 0.5$, the threshold function γ : $\gamma(t_i) = 0.1$ for $i = 1, 2, \dots, 6$, the set of operators $Op = \{ZtN, GtN, ZsN\}$ and the operator binding function δ defined as follows: $\delta(t_i) = (ZtN, GtN, LsN)$ for all transitions in the net.

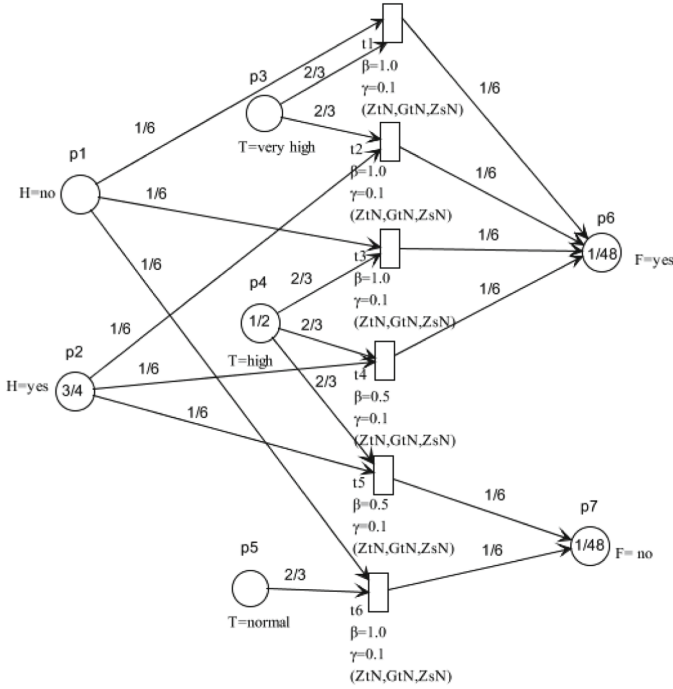


Fig. 6. An example of the WGFP-net model for the diagnosis of flu diseases with the final marking after firing the transitions t_4, t_5

Assessing the statements (descriptors) attached to places p_2 and p_4 , we observe that transitions t_4 and t_5 are enabled in the initial marking (see Fig. 5). After firing these transitions in any order we obtain the same values for the decisions $F=\text{yes}$, $F=\text{no}$ equal to $1/48$ (see Fig. 6). This means that an unambiguous decision does not exist in this case. In the net model with parameters (and this is the model presented in the paper) the problem of ambiguity of decisions is easier to solve than in the model without parameters. In a situation like this, the ambiguity of decisions could be relatively easily resolved if the weights of the output arcs for t_4 and t_5 were different. This situation is possible with a different interpretation of the weights of the input and/or output arcs in this net model. We intend to address this problem in more detail in our future research work.

It is also visible in this figure that in the current marking the transitions t_4 and t_5 are still enabled. Firing these two transitions in the current marking does not change this marking, therefore the simulation of the net operation is already completed. We omit the particular calculation in this case, because it runs similarly as in Example 5 (Sect. 3).

6 Conclusion

Trying to make fuzzy Petri nets more realistic with regard to the perception of physical reality, in this paper we established the relationship between fuzzy Petri nets and rough set theory. This link is of a methodological nature and shows the possible application of rough set methodology to transform the WGFP-net into a more realistic net model. In the proposed model, the weights of arcs and the function β are interpreted using appropriate concepts from the rough set theory, thanks to which their values are calculated from data tables. Decision rules are also automatically generated from these tables, which are the basis for building the net model of the decision algorithm. In addition, the considered net model allows the use of any triangular norms to describe the behavior of the WGFP-nets. The approach developed seems promising and one could try to apply it to problems that can be solved in a similar way.

It is worth noting that the presented net model allows relatively quickly identify the objects specified in a given decision table. However, the algorithm described does not propagate information from attributes to other attributes as soon as possible. If such an algorithm did this, we would achieve even faster decision making in the net model. It is well known that this aspect is extremely important in real-time systems. This is the reason to consider in the next study the rules in minimal form, i.e. with a minimal number of descriptors on its left hand side. Another interesting problem arises when we are unable to determine the exact membership or value of truth, then we should focus our attention on e.g. interval fuzzy sets [19] to indicate their scope instead of exact values. Therefore, it seems useful to examine the WGFP-net in the context of interval t-norms. This should make the model proposed here even more flexible, general and practical. These are just some examples of problems that we would like to examine using the approach presented in the paper.

Acknowledgement. This work was partially supported by the Center for Innovation and Transfer of Natural Sciences and Engineering Knowledge at the University of Rzeszów. We would like to thank the anonymous referees for critical remarks and useful suggestions to improve the quality of the paper.

References

1. Bandyopadhyay, S., Suraj, Z., Nayak, P.K.: Soft Petri net. In: Mihálydeák, T., et al. (eds.) IJCRS 2019. LNCS (LNAI), vol. 11499, pp. 253–264. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22815-6_20
2. Bandyopadhyay, S., Suraj, Z., Grochowalski, P.: Modified generalized weighted fuzzy Petri net in intuitionistic fuzzy environment. In: Flores, V., et al. (eds.) IJCRS 2016. LNCS (LNAI), vol. 9920, pp. 342–351. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47160-0_31
3. Cardoso, J., Camargo, H. (eds.): Fuzziness in Petri Nets. Springer, Heidelberg (1999)
4. Chen, S.M.: Weighted fuzzy reasoning using weighted fuzzy Petri nets. IEEE Trans. Knowl. Data Eng. **14**(2), 386–397 (2002)

5. Chen, S.M., Ke, J.S., Chang, J.F.: Knowledge representation using fuzzy Petri nets. *IEEE Trans. Knowl. Data Eng.* **2**(3), 311–319 (1990)
6. David, R., Alla, H.: *Petri Nets and Grafcet: Tools for Modelling Discrete Event Systems*. Prentice-Hall, Upper Saddle River (1992)
7. Ha, M.H., Li, Y., Wang, X.F.: Fuzzy knowledge representation and reasoning using a generalized fuzzy Petri net and a similarity measure. *Soft Comput.* **11**(4), 323–327 (2007)
8. Klement, E.P., Mesiar, R., Pap, E.: *Triangular Norms*. Kluwer, Dordrecht (2000)
9. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: a tutorial. In: Pal, S.K., Skowron, A. (eds.): *Rough Fuzzy Hybridization: A New Trend in Decision-making*, pp. 3–98, Springer, Singapore (1999)
10. Liu, H.-C., You, J.-X., Li, Z.-W., Tian, G.: Fuzzy Petri nets for knowledge representation and reasoning: a literature review. *Eng. Appl. Artif. Intell.* **60**, 45–56 (2017)
11. Liu, H.-C., You, J.-X., You, X.Y., Su, Q.: Fuzzy Petri nets using intuitionistic fuzzy sets and ordered weighted averaging operators. *IEEE Trans. Cybern.* **46**(8), 1839–1850 (2016)
12. Looney, C.G.: Fuzzy Petri nets for rule-based decision-making. *IEEE Trans. Syst. Man Cybern.* **18**(1), 178–183 (1988)
13. Pancarz, K., Suraj, Z.: Rough sets for discovering concurrent systems models from data tables. In: Hassanien, A., Suraj, Z., Ślęzak, D., Lingras, P. (eds.) *Rough Computing: Theories, Technologies and Applications*, pp. 239–268. IGI Global, Pennsylvania (2008)
14. Pawlak, Z., Skowron, A.: Rudiments of rough sets. *Inf. Sci.* **177**, 3–27 (2007)
15. Pawlak, Z., Skowron, A.: Rough sets and boolean reasoning. *Inf. Sci.* **177**, 41–73 (2007)
16. Pawlak, Z.: Concurrent versus sequential the rough sets perspective. *Bull. EATCS* **48**, 178–190 (1992)
17. Pawlak, Z.: *Rough Sets - Theoretical Aspects of Reasoning about Data*. Kluwer, Dordrecht (1991)
18. Pawlak, Z.: Rough sets. *Int. J. Comput. Inf. Sci.* **11**, 341–356 (1982)
19. Pedrycz, W.: *Fuzzy Control and Fuzzy Systems*, 2nd extended edn. Wiley, Hoboken (1993)
20. Peters, J., Skowron, A., Suraj, Z., Pedrycz, W., Ramanna, S.: Approximate real-time decision making: concepts and rough fuzzy Petri net models. *Int. J. Intell. Syst.* **14**, 805–839 (1999)
21. Skowron, A.: Synthesis of adaptive decision systems from experimental data. In: *Proceedings of the Fifth Scandinavian Conference on Artificial Intelligence*, Trondheim, Norway, 29–31 May 1995. IOS Press (1995). *Frontiers. Artif. Intell. Appl.* **28**: 220–238
22. Skowron, A.: A synthesis of decision rules: applications of discernibility matrices. In: *Proceedings of a Workshop on Intelligent Information Systems, Practical Aspects of AI II*, Augustów, Poland, 7–11 June 1993, pp. 30–46 (1993)
23. Skowron, A., Suraj, Z.: A parallel algorithm for real-time decision making: a rough set approach. *J. Intell. Inf. Syst.* **7**, 5–28 (1996)
24. Skowron, A., Suraj, Z.: Rough sets and concurrency. *Bull. PAS Tech. Sci.* **41**(3), 237–254 (1993)
25. Suraj, Z.: A new class of fuzzy Petri nets for knowledge representation and reasoning. *Fundam. Inform.* **128**(1–2), 193–207 (2013)

26. 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), Vancouver, Canada, 25–29 July 2016, pp. 2385–2392. IEEE (2016)
27. Suraj, Z., Fryc, B.: Timed approximate Petri net. *Fundam. Inform.* **71**, 83–99 (2006)
28. Suraj, Z., Grochowalski, P.: Petri nets and PNeS in modeling and analysis of concurrent systems. In: Proceedings International Workshop on Concurrency, Specification and Programming (CS& P 2017), Warsaw, Poland, 25–27 September 2017, pp. 1–12 (2017)
29. Suraj, Z., Hassanien, A.E.: Fuzzy Petri nets and interval analysis working together. In: Bello, R., Falcon, R., Verdegay, J.L. (eds.) *Uncertainty Management with Fuzzy and Rough Sets*. SFSC, vol. 377, pp. 395–413. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-10463-4_20
30. Zhou, K.-Q., Zain, A.M.: Fuzzy Petri nets and industrial applications: a review. *Artif. Intell. Rev.* **45**(4), 405–446 (2015). <https://doi.org/10.1007/s10462-015-9451-9>