Rough Set Approach to Conflict Analysis

Rafał Deja

rd@alta.pl, ul. Tokarskiego 4/14, 40-749 Katowice

Abstract: The Rough Set Theory and the Boolean reasoning were found the effective methods for conflict analysis. The starting point is the conflict model introduced by Pawlak. The disadvantages of this model are outlined and the new enhanced model is introduced. The consensus problem and agents strategies are discussed as examples of conflict analysis within the model.

1. Introduction

The importance of multi-agents systems, models of agents' interaction is increasing nowadays as distributed systems of computers started to play a significant role in society. An interaction occurs when two or more agents, which have to act in order to attain their objectives, are brought into a dynamic relationship. This relationship is the consequence of the limited resources which are available to them in a situation. If the number of resources is insufficient to attain agents' goals it often comes into the conflicts. This can happen in almost all industrial activities requiring distributed approach, such as network control, the design and manufacture of industrial products or the distributed regulation of autonomous robots. However, distributed systems is only one from many different areas where a conflict can arise and where it is worth to apply computer aided conflict analysis. Just to mention some human activities like business, government, political or military operations, labour-management negotiations etc.

In the paper the conflict situation model is defined based on the real data – data gathered from the conflict sides. We propose some methods to solve the most fundamental problems related to conflicts.

1.1. Pawlak Model

The model introduced in this paper is an enhancement of the model proposed by Pawlak in papers e.g. [7, 9]. In the Pawlak model, some issues are chosen, and the agents are asked to specify their views: are they favourable, neutral or against. Thus the analysis are naturally restricted to outermost conclusions like finding the most conflicting attributes or the coalitions of agents if more than two take part in the conflict [9]. In the real world, views on the issues to vote are consequences of the decision taken, based on the local issues, the current state and some background knowledge using some strategy. Therefore, the Pawlak model is enhanced here by adding to the model some local aspects of conflicts.

Example 1. Let us consider a conflict between an employer and employees represented by two different trade unions TU1 and TU2. The example of conflict is taken from the author's observation, though it has been simplified to present the defined notions rather than resolve a real conflict. Employer is interested mainly in factory profit (denoted by *k*), good investment level (*l*) and, maybe, worker's satisfaction (*m*). Job attributes considered for the workers from TU1 are compensation (*a*) and work conditions (*b*). The most important factors for the workers from TU2 are salary (*s*), social care policy (*t*) but also the level of employment (*u*) (some reductions has been proposed). We can think about these attributes quite generally, for example, compensation can consist of the worker's salary and all his income but it also can include the repeated profit division like the social fund. We analyse the conflict presented in this example more deeply in the paper.

2. Conflict model

The information about the *local states* U_{ag} of an agent *ag* can be presented in the form of an information table, creating the agent *ag*'s information system $I_{ag}=(U_{ag}, A_{ag})$, where a: $U_{ag}\rightarrow V_a$ for any $a\in A_{ag}$ and V_a is the value set of attribute *a*. We assume: $V_{ag} = \bigcup_{a\in A_{ag}} V_a$ Any local state $s\in U_{ag}$ is explicitly described by its *information vector* $Inf_{A_{ag}}(s)$, where $Inf_{A_{ag}}(s)=\{(a, a(s)): a\in A_{ag}\}$. The set $Inf_{A_{ag}}(s)$: $s\in U_{ag}$ is denoted by $INF_{A_{ag}}$ and it is called the *information vector set of ag*. We assume that sets $\{A_{ag}\}$ are pairwise disjoint, i.e., $A_{ag}\cap A_{ag'}=\varnothing$ for $ag\neq ag'$. This condition emphasizes that any agent is describing the situation in its own way. Relationships among attributes of different agents will be defined by constraints as shown in section 2.4. Example 2 illustrates local states for the labour-management conflict.

2.1. Local set of goals (similarity of states)

Every agent evaluates the local states. The *subjective evaluation* corresponds to an order (or partial order) of the states in the agent information table. We assume that the function e_{ag} called the *target function*, assigns an evaluation score to each state; let for example $e_{ag}:U_{ag}\rightarrow[0,1]$. The states with score 1 are mostly preferred by the agent as target states, while the states with score 0 are not acceptable. The agent *ag*'s *set of goals* (*targets*) denoted by $T(ag)$ is defined as the set of target states of *ag*, which means $T_{ag}=\{s\in U_{ag}: e_{ag}(s)>\mu_{ag}\}$, and μ_{ag} is the *acceptance level*, chosen by the agent *ag* – it is subjective which evaluation level is acceptable by the agent.

The state evaluation can also help us to find the state similarity [5]. For any $\varepsilon>0$ and $s\in U_{ag}$, we define ε-*neighbourhood* of *s* by: $\tau_{ag,\varepsilon}(s)=\{s'\in U_{ag}: |e_{ag}(s) - e_{ag}(s')|\leq$

ε}. The family $\{\tau_{ag,\varepsilon}(s)\}_{s\in U_{ag}}$ defines a tolerance relation $\tau_{ag,\varepsilon}(s)$ in $U_{ag}\times U_{ag}$ by $s\,\tau_{ag,\varepsilon}\,s'$ iff $s'\in\tau_{ag,\varepsilon}(s)$.

Example 2. Let us consider the situation described in Example 1. *Ag* consists of three agents: ag_1 – TU1, ag_2 –TU2 and ag_3 – the employer. Table 1 shows agent's ag_1 states, i.e., views on local issues (attributes) *a*, *b* and the state subjective evaluation. Consequently Table 2 shows the agent's ag_2 states.

Table 1. Agent ag_1 local states with subjective evaluation.

local states	a	b	e_{ag_1}
s_1	2	2	1
s_2 (current)	2	1	2/3
s_3	1	2	2/3
s_4	1	1	0
s_5	1	0	0

Table 2. Agent ag_2 local states with subjective evaluation.

local states	*s*	*t*	*u*	e_{ag_2}
s_1	2	1	2	1
s_2 (current)	2	2	1	1
s_3	2	1	1	2/3
s_4	1	2	1	1/3
s_5	1	1	2	1/3
s_6	1	1	1	0
s_7	2	0	1	0
s_8	0	1	2	0

We assume that all attributes' domains are the same, and values belong to the set V_{ag}={0, 1, 2}. One can interpret the values from set *V* as *small*, *medium* and *high* levels, respectively. Let in the considered situation, the minimal acceptable level of evaluation by the agents will be a score greater than 1/3. Accordingly set of goals of e.g. agent ag_1 is as follows: $T_{ag1}=\{s_1, s_2, s_3\}$. The set of goals can be presented in the propositional form. The information table is converted to the decision table in which the decision 1 means that the state belongs to the set of goals, while 0 that it does not (see Table 3). Then the decision rules are generated [5].

Table 3. Decision table of agent ag_3 local states.

local states	k	l	m	e_{ag_3}	decision d
s_1	2	2	2	1	1
s_2	1	2	2	2/3	1
s_3	1	1	2	1/3	0
s_4	1	1	1	1/3	0
s_5 (current)	2	0	1	0	0

Rule for *d*=1: $l_2 \vee (k_2\wedge m_2)\rightarrow d_1$, and for *d*=0: $l_0 \vee l_1 \vee m_1\rightarrow d_0$

Note 1. One can easily distinguish, depending on the context, between $a=v$, which denotes $\{ag \in U_{ag}: a(ag)=v\}$ from the boolean variable $(a=v)^*$ corresponded to $a=v$ (and denoted by $a=v$ or a_v).

The decision class for d=1 describes the agent ag_3 local set of goals: $t_{ag_3}= l_2 \vee (k_2 \wedge m_2)$. Similarly the local set of goals of agents ag_1 and ag_2 can be described respectively by: $t_{ag_1}=a_2 \vee b_2$ and $t_{ag_2}=(s_2 \wedge t_1) \vee (s_2 \wedge t_2) \vee (s_2 \wedge u_2)$.

2.2. Local conflict

The agent ag is in the *ε-local conflict* in a state s iff s does not belong to the ε-neighbourhood of s', for any s' from the set of ag-targets where ε is a given threshold. Local conflicts for an agent ag arise from the low level of subjective evaluation of the current state by ag. In other words the state s does not belong to the ε-environs of the set of goals T_{ag} i.e.: $s \notin \bigcup_{s' \in T_{ag}} \tau_{ag,\varepsilon}(s')$, where $\tau_{ag,\varepsilon}(s')=\{s'': s'' \tau_{ag,\varepsilon} s'\}$.

2.3. Situation

Let us consider a set Ag consisting of n ordered agents $ag_1,\dots,ag_n$. A *situation* of Ag is any element of the Cartesian product $S(Ag)=\prod_{i=1}^{n} INF^*(ag_i)$, where $INF^*(ag_i)$ is the set of all possible information vectors of agent ag_i, defined by: $\mathrm{INF}^*(\mathrm{ag})=\{f: A_{ag} \to \bigcup_{a \in A_{ag}} V_a(ag): f(a) \in V_a(ag) \text{ for } a \in A_{ag}\}$. The *situation* S corresponding to a global state $\overline{s}=(s_1, \dots, s_n) \in U_{ag1} \times \dots \times U_{agn}$ is defined by $(Inf_{A_{ag1}}(s_1),\dots, Inf_{A_{agn}}(s_n))$.

2.4. Constraints

Constraints are described by some dependencies among local states of agents. Without any dependencies, any agent could take the state freely and there is no conflict at all. Dependencies come from the bound on the number of resources (any kind of a resource may be considered, e.g. water on Golan Hills see [9] or an international position [6], everything that is essential for agents). Constraining relations are introduced to express which local states of agents can coexist in the (global) situation. More precisely, *constraints* are used to define a subset $S(Ag)$ of global situations. Constraints restrict the set of possible situations to admissible situations satisfying constraints.

Example 3. The following dependencies are constraints in our example. Attribute names here stand for the variables corresponding to attribute values. Constants have been taken experimentally to express relationships and to allow comparison of any two variables.

1. $a > 0$ (compensation must be medium at least)
2. $u > 0$ (the level of employment must be at least medium too)
3. $l + m \geq u$ (and depends on the investment level and workers satisfaction)
4. $2 + a \geq s + t$ (compensation includes the salary and the social care)
5. $2 + m = a + b$ (workers' satisfaction comes from a good compensation and work conditions)
6. $3 \cdot k \geq a + l + s + t$ (profit division – a very simple case, i.e., the company uses its current profit for all expenses)

Constraints above can be converted to propositional formulas ($f_{\varphi 1}$, $f_{\varphi 2}$ …, $f_{\varphi 6}$) accordingly. The conjunction of formulas $f_{\varphi} = f_{\varphi 1} \wedge f_{\varphi 2} \wedge f_{\varphi 3} \wedge f_{\varphi 4} \wedge f_{\varphi 5} \wedge f_{\varphi 6}$ defines all admissible situations in our example, i.e. all local states can coexist in the admissible situation. For example, the situation $a=2$, $b=2$, $s=2$, $t=2$, $u=2$, $k=2$, $l=2$, $m=2$ is not admissible because of constraint 6.

2.5. Situations evaluation

Usually agents tend to attain the best states without taking care about the *global good*. However, the negotiators experience shows that the real, stable consensus can only be found when the global good is considered. Thus the *objective evaluation of situations* is introduced - the expert judgement. For example the United Nation Organisation can be thought as an expert in the military conflicts.

We assume there is a function q:S(Ag)→[0, 1], called the *quality function*, which assigns a score to each situation. The set of situations satisfying a given level of quality t is defined by: $Score_{Ag}(t) = \{S \in S(Ag) : q(S) \geq t\}$.

Example 4. Table 4 presents some situations scored by an expert in our conflict.

Table 4. Objective situation evaluation (decision table)

situations	a	b	s	t	u	k	l	m	$q(S)$	decision
S_1	1	2	1	2	1	2	2	2	1	1
S_2	1	2	2	2	2	2	2	1	1	1
S_4	1	1	2	1	1	1	1	1	2/3	1
S_5	1	2	1	2	1	1	2	2	2/3	1
S_8	1	2	1	2	1	2	1	2	1/3	0
S_{11}	1	2	2	2	2	1	2	1	0	0
S_{13}	1	2	1	1	1	2	2	0	1/3	0
S_{19} (current)	1	2	2	2	1	2	0	1	0	0
		…	…	…	…	…	…	…		
S_{22}	1	2	1	2	1	1	2	1	0	0
S_{23}	1	2	1	1	2	2	1	1	1/3	0
S_{28}	2	0	1	1	1	2	2	1	0	0

The set $Score_{Ag}(2/3)$ is described by the following formula (we have generated the decision class based on minimal relative reducts).

$a_1k_2l_2m_2 \vee a_1s_2k_2l_2m_1 \vee b_2s_2k_2l_2m_1 \vee a_1u_2k_2l_2m_1 \vee b_2u_2k_2l_2m_1 \vee a_1t_1k_2l_2m_1 \vee b_2t_1k_2l_2m_1 \vee k_1l_1m_1 \vee k_1l_2m_2 \vee t_1u_2k_2l_2m_1 \vee s_1u_2k_2l_2m_1 \leftrightarrow d_1$

3. System with constraints

The multi-agent system, with local states for each agent defined and the global situations satisfying constraints, will be called *the system with constraints*. We denote our system with constraints by M_{Ag}.

4. Analysis

The conflict model introduced above gives us possibility, first to understand and, then, to analyse different kinds of conflicts. Particularly, the most fundamental problem can be widely investigated, that is, the possibility to achieve the consensus. Because of the lack of space only the consensus problem on local preferences is described in this paper. We propose Boolean reasoning [1] and Rough Set methodology [8] for all analysis. The elementary Boolean formula is usually obtained here by transforming the information table into the decision table, generating rules (minimal with respect of number of attributes on left side) and determining the description of decision class [10]. From the elementary formulas the final formula describing the problem is shaped.

4.1. Consensus problem on local and global level

INPUT

The system with constraints M_{Ag} defined in Section 3.

t - an acceptable threshold of the objective global conflict for Ag.

OUTPUT

All situations with the objective evaluation reduced to degree at most t, and without local conflict for any agent. (it is required that any new situation is constructed in the way that all local states in this situation are favourable for the agents).

ALGORITHM

The algorithm is based on verification of global situations from $Score_{Ag}(t)$ with the local set of agents goals and constraints. The problem is described by the formula f: $f = \bigwedge_{ag \in Ag} t_{ag} \wedge f_C \wedge f_\varphi$, where t_{ag} describes the set of goals of the agent ag, and f_C describes $Score_{Ag}(t)$ and f_φ the constraints. The formula $f_C \wedge f_\varphi$ representing all admissible situations without the *global conflict* regarding the threshold t.

Example 5. The way of constructing the formulas t_{ag}, f_{φ}, f_C has been presented in Example 2, Example 3, Example 4, respectively. Including the formulas describing the local set of goals the formula f has the following form: $f=(a_2\vee b_2) \wedge (s_2t_1 \vee s_2t_2\vee s_2u_2) \wedge (l_2 \vee k_2m_2) \wedge f_C \wedge f_{\varphi}$.

After reduction we have four prime implicants and four solutions – non-conflicting situations: $f=a_1b_2s_2t_1u_1k_2l_2m_1 \vee a_1b_2s_2t_1u_2k_2l_2m_1 \vee a_1b_2s_2t_1u_1k_2l_2m_2 \vee a_1b_2s_2t_1u_2k_2l_2m_2$

4.2. Calculation strategies

The reduction (calculating prime implicants) of formulas described in the previous section can be exhausted or time consuming. In the consensus problems we have to verify the local goals $f_1,...,f_n$ against the formula of core situations (belonging to *Score(t)*) and constraints f_{ϕ}. This usually yields long formulas looked like this: $f=f_1, \wedge...\wedge f_n \wedge f_C \wedge f_{\phi}$.

Simple strategies can be based on the Boolean algebra rules. First, the absorption rule has to be considered when choosing the formulas to calculate the formulas conjunction - a shorter formula can strongly reduce the longer formula being an extension of the shorter one. Thus the order of formulas conjunction is important and appropriate strategy can be built. Another important notice, which can be useful in calculation strategy is that the result (if exists) is a disjunction of f_{ϕ} components. On the other hand formulas $f_1...f_n$ consist of components based on different Boolean variables (set of attributes $\{A_{ag}\}$ are pairewise disjoint). Thus any prime implicant must contain a component from each agent formula $f_1...f_n$. In the following strategy we are verifying all remaining formulas (component after component) against agents formulas. The considered component can be removed if it does not contain any component of a given agent formula. More precisely this strategy of *preliminary reduction* can be described by the following algorithm.

Let f[][] denotes the two dimensional array for storing formulas – for each formula the components are stored and *nAg* will be the number of agents. Let first *nAg* formulas of array f[][] describe agents sets of targets.

```
for i=nAg to formulas number do
    for j=1 to components number of f[i] do
        bRemove =false;
        for z=1 to nAg do
            for k=1 to components number of f[z] do // for one agent
                if f[i][j] ∧ f[z][k] = f[z][k]  then
                    break;  // do not remove this component
                if f[i][j] ∧ f[z][k] =∅ then bRemove = true;
            endfor;
            if k<= components number of f[z] then
                break;
        endfor;
        if bRemove then remove component f[i][j];
    endfor;
```

```
endfor;
for i=1 to formulas number - 1 do      // conjunction of remaining formulas
        f [formulas number] = f[i] ∧ f[formulas number];
endfor;
print f [formulas number];  // result- reduced formula
```

The preliminary reduction strategy allows in a time depended on components number (pessimistic calculation time $O(n^2)$) to check and possibly remove these components, which are normally reduced during conjunction. However in the algorithm without preliminary reduction the number of components can at the start exponentially grow during conjunction. Unfortunately not all components which have to be reduced are removed within the proposed algorithm.

Example 6 The formula describing consensus problem on local and global level is as follows: $f=(a_2 \vee b_2) \wedge (s_2t_1 \vee s_2t_2 \vee s_2u_2) \wedge (l_2 \vee k_2m_2) \wedge (a_1k_2l_2m_2 \vee a_1s_2k_2l_2m_1 \vee b_2s_2k_2l_2m_1 \vee a_1u_2k_2l_2m_1 \vee b_2u_2k_2l_2m_1 \vee a_1t_1k_2l_2m_1 \vee b_2t_1k_2l_2m_1 \vee k_1l_1m_1 \vee k_1l_2m_2 \vee t_1u_2k_2l_2m_1 \vee s_1u_2k_2l_2m_1) \wedge f_\varphi$

Let us consider the core formula as an example. Because the first component $a_1k_2l_2m_2$ consists of component l_2 of agent ag_3 it cannot be removed. Similarly with next components. The only component which can be reduced is $k_1l_1m_1$ because $k_1l_1m_1 \wedge (l_2 \vee k_2m_2)=\varnothing$.

5. Agents' strategy analysis

The constraints imply such a relationship that the change of the local state by one agent can cause the change of local states of the other agents. The agents' transition from one state to another can come from the predefined strategy. In our model agents' strategies are extracted based on historical data – analysing situations' changes caused by agents' transition in a past (e.g. computers' log files). Because the complete description of an agent answer on the other agents' change of the state cannot be expected we are approximating the agents' strategies in the form of decision rules based on real data and inductive inference.

Example 7. Let the historical data are give in the form of an information table - Table 5. The table describes the transition from situation 1 to situation 2 with agent ag_1, ag_2 and ag_3 taking part.

Table 5. Transition table (historical data)

Situation 1								Situation 2							
a	*b*	*s*	*t*	*u*	*k*	*l*	*m*	*a*	*b*	*s*	*t*	*u*	*k*	*l*	*m*
1	1	2	1	1	2	2	0	1	0	2	1	1	2	2	1
1	1	2	1	1	2	0	1	1	1	1	2	1	2	2	1
1	2	1	2	1	2	0	1	1	0	2	1	2	2	1	2
1	0	1	2	2	2	2	2	1	1	2	1	2	2	2	2
2	1	2	2	2	2	0	2	1	2	1	2	2	2	0	2
1	0	2	1	1	2	2	1	1	1	2	1	1	2	2	0

Situation 1								Situation 2							
a	*b*	*s*	*t*	*u*	*k*	*l*	*m*	*a*	*b*	*s*	*t*	*u*	*k*	*l*	*m*
1	1	1	2	1	2	2	1	1	1	2	1	1	2	0	1
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

The decision rules are generated in the way that each time a different attribute from "situation 2" is taken as decision. For example we have the following rules:

$b_1 \vee b_0 \vee s_2 \vee t_1 \vee u_1 \vee m_0 \vee m_1 \vee a_2 \vee l_1 \rightarrow a_1$

$b_2u_2 \vee a_1u_2l_0 \vee s_1u_2l_0 \vee a_1b_2m_2 \vee b_2s_1m_2 \vee a_1l_0m_2 \vee s_1l_0m_2 \vee b_2l_2 \rightarrow a_2$

$b_1s_2l_2 \vee b_1t_1l_2 \vee m_0 \vee b_1u_2l_2 \vee b_2m_1 \vee a_1b_2u_1 \vee a_1b_1u_2 \vee b_2s_1u_1 \vee b_0s_1u_1 \vee s_1u_1l_0 \vee s_2u_2l_2 \vee s_1l_0m_1 \vee s_2l_2m_2 \vee a_1t_2u_1l_0 \vee t_2l_0m_1 \vee t_1l_2m_2 \vee b_1t_1u_2 \vee b_0t_2u_1 \vee b_1t_1m_2 \vee b_1l_2m_2 \vee u_1l_1 \vee t_1u_2l_2 \vee a_1b_1m_2 \vee b_0u_1m_2 \vee s_1u_1m_2 \vee u_1l_2m_2 \vee a_1u_1m_2 \rightarrow b_0$

etc. etc.

The meaning of the rules above is for example that the agent ag_1 can (based on strategy) move to the local state where a=1 iff e.g. b=1 or b=0 or s=2 etc.

5.1. Strategic situation space

Strategies introduce the new constraints into the set of admissible situations. In a given situation the available situations in strategy can be determined. To this end the conjunction of a given situation with the antecedent part of rules describing strategy must be constructed. The conclusion of the rules without contradiction forms (describes) the strategic situation space. Algorithm above can be described as follows:

Let $fst(a_i)$ is a description (antecedent part of a rule) for decision a_i, and f_S is the formula describing current situation i.e.: $f(a_i) = f_S \wedge fst(a_i)$ .If $f(a_i) \equiv f_S$ then the situation S can be changed into such a situation S', where a_i exists. The conjunction of a_i for $a \in A$ is describing the new situation S'. The set of situations S' is described by formula $f_Z(S)$: $f_Z(S) = \bigwedge_{\substack{a \in A \\ f(a_i) \equiv f_S}} (\bigvee_{i \in V_a} a_i) \wedge f_\varphi$, where f_φ describes constraints. The formula f_Z describes strategic situation space admissible from situation S. The set of situations described by f_Z will be denoted by $Z(S)$.

Example 8. Let us find the strategic situation space from the situation $f_S = a_2b_1s_2t_2u_1k_2l_0m_1$ based on strategies described in the previous example.

$f(a_1) = (b_1 \vee b_0 \vee s_2 \vee t_1 \vee u_1 \vee m_0 \vee m_1 \vee a_2 \vee l_1) \wedge a_2b_1s_2t_2u_1k_2l_0m_1 \equiv (b_1 \vee s_2 \vee u_1 \vee m_1) \wedge a_2b_1s_2t_2u_1k_2l_0m_1 \equiv a_2b_1s_2t_2u_1k_2l_0m_1 \equiv f_S$ (no contradiction)

$f(a_2) = (b_2u_2 \vee a_1u_2l_0 \vee s_1u_2l_0 \vee a_1b_2m_2 \vee b_2s_1m_2 \vee a_1l_0m_2 \vee s_1l_0m_2 \vee b_2l_2) \wedge a_2b_1s_2t_2u_1k_2l_0m_1 \equiv 0$ (contradiction)

$f(b_0) = fst(b_0) \wedge f_S \equiv f_S$ (no contradiction)

etc. etc.

The strategic situation space from the considered situation is described by the following formula (it contains 66 situations after reduction):

$f_Z(S) = a_1 \wedge (b_0 \vee b_1 \vee b_2) \wedge (s_1 \vee s_2) \wedge (t_1 \vee t_2) \wedge (u_1 \vee u_2) \wedge k_2 \wedge (l_0 \vee l_1 \vee l_2) \wedge (m_1 \vee m_2) \wedge f_\varphi$

5.2. Conflict prediction

The conflict threat appears when agents' strategies yield into the global conflict. The conflict threat can be detected by verifying the strategic situation space after each agent move (each situation change). If the number of *Score* situation in the strategic situation space is less than the given safety level then the conflict threat appears.

6. Conclusions

We have presented and discussed the extension of the Pawlak conflict model. The understanding of the underlying local states as well as constraints in the given situation is the basis for any analysis of our world. The local goals and the evaluation of the global situation are observed as factors defining the strength of the conflict and can suggest the way to reach the consensus. Some problems e.g. reaching the consensus and conflict prediction based on agents' strategy has been pointed in the paper. The boolean reasoning and rough set theory has been successfully applied for analysing presented problems.

References

1. Brown FN (1990) Boolean Reasoning. Kluwer, Dordrecht.
2. Deja R (2000) Conflict Analysis. Rough Set Methods and Applications New Developments. In: L. Polkowski et al. (eds.) Studies in Fuzziness and Soft Computing, Physica-Verlag, pp.491-520.
3. Fraser NM Hipel KW (1984) Conflict Analysis: Models and Resolutions, North-Holland, New York.
4. Kleinberg J, Papadimitriou Ch, Raghavan P, (1998) A microeconomic View of Data Mining. Journal of Data Mining and Knowledge Discovery, vol. 2, issue 4, pp. 311-324.
5. Komorowski J, Pawlak Z, Polkowski L, Skowron A, (1999) Rough sets: A tutorial. In: SK Pal and A Skowron (eds.). Rough fuzzy hybridization: A new trend in decision making, Springer-Verlag, Singapore, pp. 3-98.
6. Nęcki Z (1994) Negotiations in business. Professional School of Business Edition. (The book in Polish), Krakow.
7. Pawlak Z (1984) On Conflicts. Int. J. of Man-Machine Studies, 21, pp. 127-134.
8. Pawlak Z (1991) Rough Sets - Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht.

9. Pawlak Z (1998) An Inquiry into Anatomy of Conflicts. Journal of Information Sciences 109 pp. 65-78.
10. Polkowski L, Skowron A (Eds.) (1998) Rough Sets in Knowledge Discovery (two parts). Physica-Verlag, Heidelberg.
11. Rosenheim JS, Zlotkin G (1994) Designing Conventions for Automated Negotiation. AI Magazine 15(3) pp. 29-46. American Association for Artificial Intelligence.
12. Sandholm T, Lesser V (1997) Coalitions among Computationally Bounded Agents. Artificial Intelligence 94(1), pp. 99-137, Special issue on Economic Principles of Multiagent Systems.
13. Skowron A, Rauszer C (1991) The Discernibility Matrix and Functions in Information Systems. Institute of Computer Science Reports, 1/91, Warsaw University of Technology, and Fundamenta Informaticae.
14. Sycara K (1996) Coordination of Multiple Intelligent Softwareagents. International Journal of Cooperative Information Systems 5(2-3) pp. 181-211.