IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 5, Ver. III (Sep.- Oct. 2017), PP 16-22
www.iosrjournals.org

Design and Implementation of a General System for Conflict Analysis

Prof. Dr. Hussein K. Khafaji[1], Huda Ahmed Abed[2]
[1]*AL-Rafidain University College/Computer Communication Eng.Dept*
dr.hkm1811@yahoo.com
[2]*Iraqi Commission for Computers and Informatics/ Informatics Institute for Postgraduate Studies*

Abstract: *To strengthen the dominant's dominance or to weaken that dominance, the conflict's agents seek to create alliances. The conflict is a life phenomenon; individuals, companies, states, organizations, governments, etc., are in conflict. Due to the importance of the conflict and its great impact on human interests and relations, many methods have been suggested to study and analyze it. The rough set theory is one of the important tools to deal with inconsistency, which characterizes conflict. Inconsistency in conflict represents uncertainty about alliance, neutrality and struggle among agents in a conflict situation. This paper presents the design and implementation of a proposed general system for conflict analysis that depends on rough set theory and conflict theory. It discusses the overall process of this system from obtaining dataset, factors computation, computing the distance function, and then visualizing the results. Real conflict datasets were used to test the system. Furthermore, synthetic datasets, with different sizes, were utilized to study the scalability of the proposed system.*

Keywords: *Rough set, conflict system, conflict analysis.*

Date of Submission: 13-09-2017 Date of acceptance: 02-10-2017

I. Introduction

Conflict analysis has an important role both in theory and in practice. It has been utilized in different prominent domains like commerce, economy, political contention, governmental, management negotiations, military attacks and etc. [1]. There is many methods to process these problems using different approaches like Rough set theory [2], fuzzy set[3] , Vague set theory[4], Soft set theory[5].

Rough set theory is considered a powerful mathematical, non-statistical tool that suitable to analyze various types of data, particularly, when it processes inaccurate, uncertain or inexplicit knowledge. It is representing and extracting knowledge from deficient or noisy data [6] [7], it was developed by Zdzislaw Pawlak [2][8][9] [10] in (1982). Pawlak suggested a to analyze a conflict depending on rough set theory [11]. Next sections will present a well demonstration about the overall model, due to the proposed system depends on this model.

The Pawlak's conflict model consists of agent AGs, and ISS represents issues, the domain of the values to this issues is Vs = {−1, 0, 1}. By (-1) is meant "in conflict with", "no", "negative opinion", "disagree", etc. (0) is neutral case, while the value (1) means "alliance", "coalition", "yes", "agree", etc. The value s(x) is view of agent x to issue s where x ∈ AG, and s ∈ ISS

$$\varphi s(x, y) = \begin{cases} 1 & \text{if } s(x)a(y) = 1 \text{ or } x = y, \text{ // agents x and y are in alliance.} \\ 0 & \text{if } s(x)s(y) = 0 \text{ and } x \neq y, \text{ // neutral case} \\ -1 & \text{if } s(x)s(y) = -1. \text{ // agents x and y are in conflict.} \end{cases}$$

A distance function ρ∗ between agents x and y with consideration the set of issues A ⊆ ISS
A : AG × AG → [0, 1] is cleared

$$\rho^*_A(x, y) = \frac{\sum_{s \in A} \phi s(x, y)}{card\,\{A\}}$$

Where:

$$\phi^* s(x, y) = \frac{1 - \phi s(x, y)}{2}$$

Then apply $\varphi^{*}s(x, y)$ depending on the value of auxiliary function $\varphi s(x, y)$ to get:

$$= \begin{cases} 0 & \text{if } s(x) \times s(y) = 1 \text{ or } x = y, \\ 0.5 & \text{if } s(x) \times s(y) = 0 \text{ and } x \neq y, \\ 1 & \text{if } s(x) \times s(y) = -1. \end{cases}$$

II. Proposed Design and Implementation of Conflict System

A general system has been built depending on rough set theory (RST). It has the capability to deal with any kind of the agents with different types of issues. The system consists of two main stages:

1- Factors computing stage: The first stage is the process of choosing only one issue, then calculating the factors to know the final voting for each group of agents. After that, information system will be built on which the conflict theory will be applying to discover alliances and disputes depending on a specific issue.

2- Conflict model: The second stage is the studying and discovering of the conflicts and alliances depending on all selected issues. It is worth mentioning that it is possible to start analyzing the issues from this stage in a state of availability of the information system, which contains one opinion for each agent's group. The algorithm in Figure.1 shows the main steps of the proposed system. The architecture of the proposed system is presented in Figure.2. This architecture shows the data flow and execution flow as dashed and solid lines respectively.

General Conflict System Algorithm

{

1. Connect to specific dataset(decision table) //**first stage**
2. Read an issue, from the issues, from dataset, Current Issue.
3. Identify condition and decision attribute from decision table.
4. Apply the granularity and indiscernibility using the strength, certainty, and coverage factors for all agents and Current Issue.
5. Repeat steps 2, 3, and 4.for another issue if exists.
6. Construct the Information System
7. Apply the conflict theory depending on discernibility matrix to know the degree of conflict, alliance, and neutral. //**second stage**
8. Discover conflict and alliances using flow graph.

}

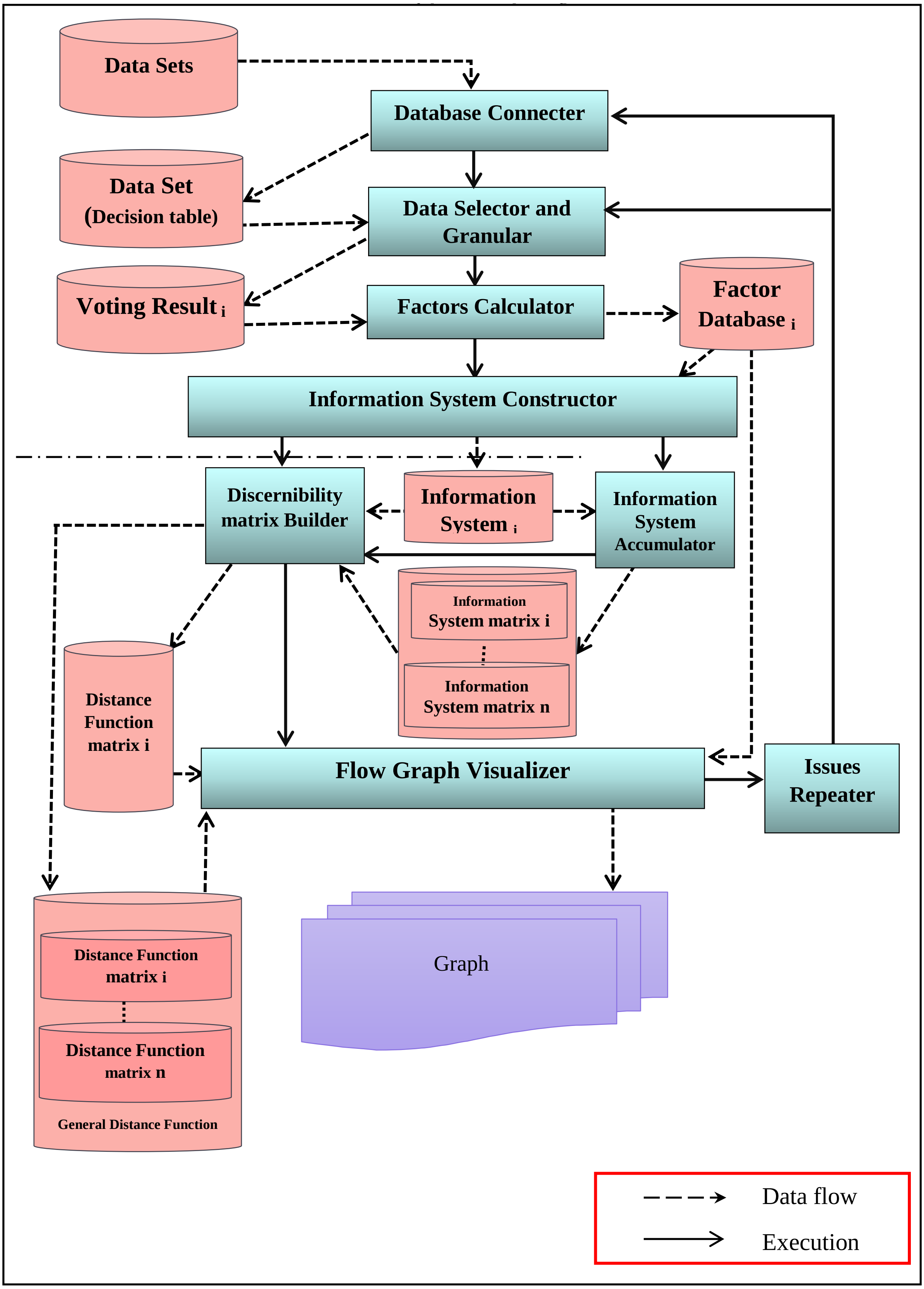

Figure .2 Conflict System Architecture

In addition, it elucidates the input and output of each module. The system consists of many modules such as Database Connecter (DBC), Data Selector& granular (DS&G), Factors Calculator (FC), Information System Constructor (ISC) and so on. Generally, the system can mine the conflict according to one or more issues datasets that can be managed by different DBMSs. The system graphically visualizes the conflict among the agents according to the issue(s) under consideration. The process of conflict mining is accomplished through many trajectory operations. The next subsections will explain the design and implementation of each module.

2.1 Database Connecter (DBC)

The database Connecter (DBC) offers the capability to import data from various types of DBMSs as data sources. It deals with the most important and well-known types that always used in this field of study like (csv, xlsx, spss, accdb). The output of this module is a connected dataset. The general schema of the dataset is as follows:

Agent1	…	$Agent_n$	Issue1	….	$Issue_m$

2.2 Data Selector and Granular (DS&G)

The input for this module is a connected dataset (decision table) and the output is voting result to a specific issue. The decision system consists of a set of conditions C and decision d. The process of selection begins by choosing a specific condition C to be analyzed from a set of condition attributes, which can be considered as an agent leading to obtaining an appropriate decision. After manipulation of the selected C and d to mine conflicts and coalitions, the system enables the user to select another decision (opinion/or voting) to another issue until processing of all required issues.

So to make the granularity, a specific condition (in this situation called agent) will be selected that be fixed for all issues and during the run of the system.

After the selection process, the turn is for the following computation:

- Compute the number of objects or voters.
- Compute the support or frequency that contains the number of voters for each decision.

It has been obtained for each granular by applying the indiscernibility concept of RST.

Finally, the selection and granularity process will be repeated for the same agent but to another issue (if exists) according to the number of selected issues.

2.3 Factors Calculator (FC)

It is well known that there are many values in one condition or issue according to their domains of values; otherwise, the decision rules become redundant due to its high confidentiality which equivalent to 100%. The number of objects may represent all people in country, members of parliaments, individuals within a party, or any other groups. Therefore, a decision rule can be produced with an approximation. Factors Calculator (FC) takes the voting result table for specific issue as input and then calculates factors to each decision rule. When calculating strength factor to each row, FC takes support to row or object under study and divide it by a summation of support for all rows in factor database. To compute the certainty factor to each row, FC takes support to row or object under study and divide it by the summation of support for just rows that have identical conditions. Finally, to compute a coverage factor to each row, FC takes support to row or object under study and divide it by summation of support for just rows that have an identical decision.

2.4 Information System Constructor (ISC)

After computing, the degree of each factor that effects on decision-making. Factor Database for specific issue becomes ready to choose one object from each group to avoid repetition in condition or object. For this reason, one opinion has been taken which represents the majority voting for agents belong to the same group. However, ISC receives the Factor Database as input, to obtain the last decision for each group without redundancy. The last decision represents an information system for a specific issue.

```
Information System Constructor (ISC) Algorithm
Input:     Factor Database_i //for issue#i
Output:    Information System_i //for issue#i
  1. {
  2.  While not end of Factor Database_i
  3.   {
  4.     Compare condition objects in one group;
  5.     Take one condition that have maximum certainty value in each
         group ;
  6.     Add maximum attribute condition to information system_i ;
  7.   }//End while
  8. } //ISC
```

Figure.3 algorithm to construct Information System

2.5 Information System Accumulator (ISA)

In ISC stage, the information system of one particular issue has been computed. But what about the information systems for issues previously calculated and analyzed and where are they saved?. Information System Accumulator (ISA) is collecting the information system for all issues under analysis in a single system.

Each time the input for this module is one attribute that represents opinions of agents to the selected issue. ISA adds this issue to previous calculated issues to the same option of agents. The last output to this module is information system that contains all opinions for all issues under analysis.

2.6 Discernibility Matrix Builder (DMB)

The most important process in studying and analyzing conflict is discernibility matrix that is one of RST concepts. The Builder takes information system to issue(s) as input, then it computes binary relations between each pair of agents by giving specific values to (similarity=0, difference=1, and neutral=0.5) depending on conflict theory as an algorithm in Figure.4. The information system may contain opinions to the number of agents for one issue or more based on issues number and agents' number under analysis.

The algorithm in Figure.4 reflects more exactly the differences between views of agents. Its output represents a Distance Function matrix to either one issue or more. For more explanation, if the algorithm information system as a matrix then it scans all rows and column by comparing each row with others, (except with the agent himself), depending on the binary relations. The input information system contains agents in rows and issues in columns with values of the agent’s opinions which are restricted to the values (1, 0, and -1) as against, neutral, and favorable toward the issues respectively, If there was no conflict between two agents then do nothing. If one of them or both equal to 0 add 0.5, else add 1 to the distance matrix that represents conflict to distance matrix. Finally, to obtain a Distance matrix as output, divide its elements by the number of selected agents. DMB algorithm does not obtain conflict or alliance only, but also the approximated degree of conflict or alliance in. The degree of conflict can decrease or increase depending on specific issues.

```
Discernibility matrix Builder Algorithm //DMBA
Input:      Information System [Agent#] [Issues#];
Output:     Distance [Agent#] [Agent #]; // Distance Function matrix
{
for (i=1; i<Agent#; i++)
    for (j=i+1; j<= Agent#; j++)  {
        for(k=1; k< =Issues# ;k++)
         If   (Information System[i] [k] = 0) OR (Information System[j] [k] = 0
           Distance [j] [i] = Distance [j] [i] + 0.5;
         Else if Information System[i] [k]! = Information System[j] [k]
           Distance [j] [i] = Distance [j] [i] + 1;
      Distance [j] [i] = Distance [j] [i] / Issue#
    }//end for j
}//end of Distance Function
```

Figure.4 algorithm to build distance function

2.7 Flow Graph Visualizer (FGV)

This module can be considered as the most interactive stage. It displays all results in an attractive, simple, and understandable manner. These results show all agents and their opinions as vertices, and the relationships between them as branches (links). This module was implemented in a way to be a general converter, (drawer), of adjacency matrix to the graph. The output of some modules in the system is represented as bipartite graph when the data are separated into two sets. There is more than one input to this module for example:

1. Factor Database

After computing the factor database to a particular issue, FGV receives this database as input. It displays each factor separately using the algorithm presented in Figure (3.9). As it is mentioned previously there are three types of factors; strength, certainty, and coverage, their visualization manipulation are as follows:

a- **Strength factor**: The strength depicts the relation between the agents and the type of voting, i.e. -1, 0, and 1. FGV shows all agents, their opinions, and the values of strength as a weight for each branch.

b- **Certainty factor**: FGV displays all agents that have certainty degree of voting more than 0.5 by using a graph.

c- **Coverage factor**: The visualize works in the same way by displaying just agents that have coverage degree of voting more than 0.5 by using a graph.

2- Distance function to issue(s) visualization:

The algorithm in Figure.5 designed and implemented to visualize distance functions. Remember that distance function consists of values ranged between 0 and 1. FGV will display the conflict and alliance between agents depending on the issue(s) understudy. Note that the neutrality case does not mean alliance since agents may be have contrary views; also, it does not mean conflict. The lack of knowledge to the real opinion may lead to neutrality. Therefore, there is no line for neutral case.

```
Flow Graph Visualizer for conflict&aliance situation
Input:   Distance Function [Agent#] [Agent#]; //Distance Function_i Matrix
Output: drawing (Conflict, Alliance, & neutral graph);
{
for (i = 1,i<= (#Agent-1), i++)
  {
      x= i+1;
      for (j = x,: j <= #Agent, j++)
        {
          if(Distance Function [j][i]< 0.5) //allied
             draw dotted line between nodes;
          if(Distance Function [j][i]> 0.5)//conflict
             draw solid line between nodes;
          // else if(Distance Function [j,i]== 0.5) neutral
           // Don't draw any line;
        }
  }
}//end
```

Figure .5 Algorithm for drawing conflict, alliance situation using Flow Graph Visualizer

2.8 Issue Repeater (IR)

After the conflict and alliance have been discovered depending on the selected issue, and display them in a graph, it is possible to repeat all previous operations for another issue. It is possible that IR either goes to DC (Database Connecter) to select a new issue from another dataset (decision table), taking into consideration that the same agent's type should be selected or simply goes to DS&G (Data Selector and Granular) for selecting a new issue from the same dataset.

III. Conclusions

A general system has been built depending on rough set theory. It has the capability to deal with any kind of the agents with different types of issues. The limitations of the deliberated issues number are influenced by the capabilities of the device used. Therefore, if the number of issues increased, they may cause a delay in the calculation relayed on the memory and processor used. It was applied to discover the alliances and conflicts among the agents who have effective role the in the Syrian conflict problem, also a dataset of USA presidential election was used to test the proposed system [12]. A synthetic database of gradually increased numbers of issues and agents is used to test the scalability of the proposed system. The results showed the scalability and efficiency of the system.

References

[1] B. Sun, W. Ma, and H. Zhao, "Rough set-based conflict analysis model and method over two universes," *Information Sciences*, vol. 372. pp. 111–125, 2016.
[2] Z. Pawlak, "Rough sets," *Int. J. Comput. Inf. Sci.*, vol. 11, no. 5, pp. 341–356, 1982.
[3] L. a. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965.
[4] B. D. Gau W-L, "Vague sets," *IEEE Trans. Syst. Man. Cybern.*, vol. 23, 1993.
[5] D. MOLODTSOV, "Soft Set Theory First Results." An international Journal Computers and Mathematics with Applications - Elsevier, pp. 19–31, 1999.
[6] M. Podsiadlo and H. Rybinski, "Financial time series forecasting using rough sets with time-weighted rule voting," *Expert Systems with Applications*, vol. 66. pp. 1339–1351, 2016.

[7] M. Sudha, "Intelligent decision support system based on rough set and fuzzy logic approach for efficacious precipitation forecast," *Decision Science Letters*, vol. 6. pp. 95–106, 2017.
[8] J. Nasiri and M. Mashinchi, "Rough Set and Data Analysis in Decision Tables," *J. Uncertain Syst.*, vol. 3, no. 3, pp. 232–240, 2009.
[9] Z. Pawlak and A. Skowron, "Rough sets: Some extensions," *Inf. Sci. (Ny).*, vol. 177, no. 1, pp. 28–40, 2007.
[10] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Inf. Sci. (Ny).*, vol. 177, no. 1, pp. 3–27, 2007.
[11] Z. Pawlak, "An inquiry into anatomy of conflicts," *Inf. Sci. Elsevier scinece*, vol. 109, no. 1–4, pp. 65–78, 1998.
[12] **https://wiki.csc.calpoly.edu/datasets/wiki/LargeTree**

Prof. Dr. Hussein K. Khafaji. "Design and Implementation of a General System for Conflict Analysis ." IOSR Journal of Computer Engineering (IOSR-JCE) , vol. 19, no. 5, 2017, pp. 16–22.