International Journal of Computer Science and Informatics

Volume 3 | Issue 4 Article 10

April 2014

APPLYING ROUGH SET THEORY TO GENETIC ALGORITHM FOR WEB SERVICE COMPOSITION

CARIAPPA M.M
Department of Information Science and Engineering, M.S.Ramaiah Institute of Technology, Bangalore, India, cariappa24merianda@gmail.com

MYDHILI .K. NAIR
Department of Information Science and Engineering, M.S.Ramaiah Institute of Technology, Bangalore, India, mydhili.nair@gmail.com

APPLYING ROUGH SET THEORY TO GENETIC ALGORITHM FOR WEB SERVICE COMPOSITION

CARIAPPA M.M[1] & MYDHILI .K.NAIR[2]

[1,2]Department of Information Science and Engineering, M.S.Ramaiah Institute of Technology, Bangalore, India
E-mail : cariappa24merianda@gmail.com, mydhili.nair@gmail.com

Abstract - Rough set theory is a very efficient tool for imperfect data analysis, especially to resolve ambiguities, classify raw data and generate rules based on the input data. It can be applied to multiple domains such as banking, medicine etc., wherever it is essential to make decisions dynamically and generate appropriate rules. In this paper, we have focused on the travel and tourism domain, specifically, Web-based applications, whose business processes are run by Web Services. At present, the trend is towards deploying business processes as composed web services, thereby providing value-added services to the application developers, who consumes these composed services. In this paper, we have used Genetic Algorithm (GA), an evolutionary computing technique, for composing web services. GA suffers from the innate problem of larger execution time when the initial population (input data) is high, as well as lower hit rate (success rate). In this paper, we present implementation results of a new technique of solving this problem by applying two key concepts of rough set theory, namely, lower and upper approximation and equivalence class to generate if-then decision support rules, which will restrict the initial population of web services given to the genetic algorithm for composition.

Keywords-*Rough Set Theory; Genetic Algorithm; Web Service Composition; Rule Generation; Decision Support*

I. INTRODUCTION

In today's world of changing business there is an inherent need for collaboration among various organization for the purpose of rapid delivery of customer oriented solutions and internal growth of organization business[7].

Web service, which are essentially services offered on the web, play a vital role in the integration of large scale applications and dynamic participation among organization to provide essential solutions to organizational concerns [2].

Web service as defined by W3C architecture working group, is a software application identified by a URI, whose interfaces and bindings can be defined, described, and discovered as XML artifacts. Web service in general is an implementation of Service Oriented Architecture (SOA). Service Oriented Architecture is defined as set of concepts and methodology for developing software as interoperable component objects [2].Realizing SOA principles using web services for large scale business integrations is a proven trend in today's business world.

The web service include two operating components which are service consumer and service provider .The components work in distributed client–server pattern where service consumer behaves as client and service provider act as the server end[9].

There are a tremendously large number of web services, fulfilling different purposes and providing viable solutions to organizations and consumers. To provide a suitable and optimal solution to consumers and to have better collaboration among participating organizations there is a growing demand for *composing these services*[11][12] as a means of providing value added service that benefit all parties involved.

For composing web services and providing useful solution we need to consider many non-functional requirements of individual web services such as (reliability, performance, execution time etc.).

Genetic algorithm is a sub set evolutionary algorithm which is used to solve optimization problems. As indicated by [5][11][12]*"A genetic algorithm (GA) is a search heuristic that resembles the process of natural evolution."* Genetic algorithm solves optimization problems using its tool set which comprises of inheritance, mutation, selection, and crossover operations.

Basic purpose of using genetic algorithm in web service composition is generating optimal chromosomes. Each of these chromosomes is a collection of genes. Each gene represents a web service and chromosome represents the composed web service.

Genetic algorithm progresses in generations (i.e. iterations), in each generation the fitness of every individual is checked (i.e. Web services).In each generation some individual are combined (or mutated) to form new population used in the next generation of the algorithm [11][12]. Finally after a certain number of generations we obtain optimal composed web service.

Genetic algorithm is not effective when the input domain of population is large. Rough set theory generates rules that resolve ambiguity in the information system and generate final decision rules that are used by genetic algorithm to process input population domain. Rough set theory plays a vital role in adding that intelligence to the Genetic Algorithm that makes it effective.

One of the inherent problems with genetic algorithms is that it does not impose any restriction on its initial population of genes. This leads to a massive number of genes which qualify as the initial population set, many of them unwanted and not meeting customer specifications [12].

Rough set theory addresses this problem by filtering out the genes which meets the customer specifications and classifying the genes as per the domain they need to be applied. In this paper, we have filtered out the genes which does not meet the customer specifications of non-functional characteristics such as high reliability, high performance and low cost. We have also classified the genes under three domains, namely, airline, car rental and hotel which can potentially be composed for usage in a travel and tourism applications.

As indicated by [11][12], the time of execution is much lesser and the hit rate of getting the optimal solution is much higher, when the initial population of genes is restricted using rough set theory. In this paper, we have leveraged on the finding [5][11], and applied rough set theory to generate certain rules based on the input data. Only those genes, in this case web services that satisfy these rules can appear as initial population for the genetic algorithm, which is used to compose the web services.

The rest of the paper is organized as follows: Section II provides a conceptual foundation for rough set theory. Section III describes the implementation and analysis of rough set theory. The paper ends with Section IV, the conclusion and future scope of work.

II. CONCEPTUAL FOUNDATION

In this section we describe the conceptual details of the rough set module. Section 2.1 briefly describes the basic concepts of rough set theory. Section 2.2 defines and elaborates on the implementation steps involved in classification and rule generation in our implementation of rough set theory.

2.1 Basic concepts of Rough Set Theory

2.1.1 Concept of Approximation space

Let IS = (U, A) be an approximation space or (Information System), where: U is non- empty finite set of objects.(i.e. U={u1, u2, u3,u4,….,u10}) and A is non-empty finite set of attributes.(i.e. A={a1,a2,a3,a4,…..,a10}).

The information function or(approximation function) is defined as $f_a : U \longrightarrow V_a$,where : a={an attribute defined by ,a$\in$ A}and V_a={domain of attribute a}[1][3][4][10].

2.1.2 Equivalence relation

Consider a set of attributes α, such that α $\subset$ A. We define the equivalence relation or(indiscernibility relation) as follows:

$\forall$ (x ,y) $\in$IND (α), x and y are indiscernible from each other based on the set of attributes defined in α.

The objects that are indiscernible from each other are grouped into equivalence class or(elementary set).

The equivalence classes for set U ,in the attribute space A is given by the notation U/A[1][10], similarly the equivalence classes for set U, in attribute space α (α$\subset$ A) is given by U/α.

2.1.3 Lower and upper approximation

Let X be our target set such that X$\subset$U, then the lower approximation of set X in attribute space α (α $\subset$ A) is given by,

$$\underline{\alpha} X = \{x \mid [x]_\alpha \subseteq X\}$$

Where:

$[x]_\alpha$ = {Set of equivalence class of α –Indiscernible relation}.

In simple works the lower approximation of a set is defined by the set of objects that surely belongs to the set [1][13].

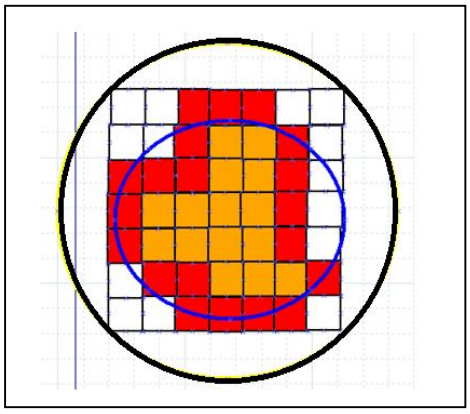


Fig.1 Diagrammatic view of lower and upper approximation.

In the Fig.1 the light coloured portion corresponds to the lower approximation and the outer dark coloured portion correspond to the upper approximation. The boundary region is the difference between the upper and the lower approximation.

The upper approximation of set X in attribute space α (α $\subset$ A) is given by,

$$\overline{\alpha} X = \{x \mid [x]_\alpha \cap X \neq \phi\}$$

In simple words the upper approximation of a set is defined by the set of objects that possibly belongs to the set.

The difference between the upper and the lower approximation defines the boundary region of the given set.

The boundary region is defined as follows [3][6]:
$\bar{\alpha} X - \underline{\alpha} X$.

2.1.4 Reducts
A reduct is defined as the minimum set of attributes that is required to discern one object from all other objects based on a specific decision[1][3][14].

In formal terms, a reduct is subset of attributes; REDUCT $\subseteq$ A (i.e. Full set of attributes) such that the equivalence classes generated by the reduct is same as the equivalence class generated by full set of attributes [8]:

$$[x]_{reduct} = [x]_{A.}$$

2.1.5 Decision rules
Decision rules are the *if-then* rules, where the *if* part corresponds to the attributes and their corresponding values in the Information system. The then part corresponds to one or more decision class [1][14][15].

The generated decision rules are evaluated using three supporting concepts: *Support*, *Accuracy* and *Coverage*.

Support is defined as the number of objects in the Information system that matches the corresponding decision rule.

Accuracy is defined as the ratio of number of objects that match the *if* part of the decision rule to that of number of objects that match the *if* part as well as the *then* part of the decision rule.

Coverage is defined as the ration of number objects that match the *then* part of the decision rule to that of the number of objects that match the *then* part as well as the *if* part of the rule.

The order of priority is of the supporting concepts for decision rules are given as follows:

Priority level 1= Support, Priority level 2 = Accuracy, Priority level 3= Coverage.

2.2 Classification amd rule generation
The following are steps involved in classification and rule generation based on our implementation rough set theory.

STEP1: Define the Information table

- The information table is defined as a n*n matrix (where n corresponds to number of objects or number of features, variables, characteristics or properties).
- The row of the table or (matrix) corresponds to the object of interest and the column defines the attributes or (variables or features or properties).
- The attributes are classified into conditional and decision attributes, where conditional attributes defines feature or property or characteristics and decision attribute defines the final outcome or result.

STEP2: Construction of Coded Information table

- The coded information table is constructed by quantification of the information table generated in the previous step.
- In this step we map each attribute value to corresponding numerical value.
- This process is carried out until all the entries in the table have been mapped to their corresponding numerical alternative.

STEP3: Generation of Equivalence class table

- In this step the coded information table generated in the previous step is processed to generate the equivalence class table.
- Each of the objects in the coded information table is compared with all other objects and grouped into equivalence class based on the indiscernibility condition that exits among their conditional attributes.
- If any ambiguity among the decision attribute is observed in any equivalence class, then all the decision attributes or outcome are grouped into a generalized decision for that equivalence class.

STEP4: Generation of Discernibility matrix

- In this step the discernibility matrix is generated by specifying which set of attributes that discern different equivalence class.
- This is done by processing the outcome of the previous step (i.e. the equivalence class).This is achieved by pair wise comparisons of all the equivalence classes.
- Equivalence classes that have the same generalized decision cannot be discerned from each other.
- For those equivalence classes that belong to different generalized decision class the discernibility matrix is generated by identifying one or more which have a different entry in the equivalence class table.

STEP5: Generation of Reducts based on the Discernibility function

- The minimum information needed to discern a equivalence class from all other equivalence classes with different decision is defined as a discernibility function for that equivalence class.

- Each column of the discernibility matrix corresponds to a discernibility function for that particular equivalence class.

For each of this discernibility function the reducts are generated by considering a attribute from each entry for a equivalence class and composing them to getminimal set of attributes that discern that equivalence class from all other objects (i.e. equivalence class).

STEP6: Reduct classification

- All the reducts generated in the previous step, is classified based on the outcome into High Performance reducts or High Reputation reducts or Low Cost reducts.

STEP7: Rule generation and optimization

- For each of the reducts that is generated in the previous step if-then rules are constructed by considering the outcome of STEP 2, STEP 3, and STEP 5.The ambiguities that exist in the then part of the rule are neutralized by calculating the Support, Accuracy and coverage for each rule.
- Each rule is classified based on the decision attribute or outcome it supports.
- The final decision rules are calculated for each category of the rule (i.e. based on the decision attribute it supports) by summing up the values of Support, Accuracy and Coverage:

Resultant = Support + Accuracy + Coverage.

- The rule with highest resultant is the final rule for that category of decision attribute.
- The above step is performed for all other categories of web services and if any deadlock condition exist,(i.e. two or more resultant have same highest value) it is broken using the priority level criteria described in Section 2.1.5: Priority Level=Support > Accuracy > Coverage.

III. IMPLEMENTATIONAL DETAILS

In this section we describe the implemental details of the rough set module which is implemented in JAVA. Figures in this section shows the actual screenshot of the output obtained after implementing the rough set theory. Section 3.1 describes rough set theory taking travel reservation process as an example. Section 3.2 briefly describes experimental results and analysis.

3.1 Example Illustration of Rough Set Theory

In this section we briefly describe a example of travel reservation process which is implemented based on rough set theory.

The travel reservation process consists of *three* types of service: *Airline Service, Hotel Service and Car-rental Service.*

For each service we first define the information table and follow all the steps that were discussed in the Section 2.2.

For the sake of brevity in illustration we only describe the classification and rule generation for airline service.

STEP1: Define the Information table

First we need to define the information table for Airline Service.

The Airline Services is listed under the column U.

i.e. U={AS1,AS2,AS3,AS4,…,AS9}

The Information table for Airline Service consist of conditional attributes defined by *:{ SUPPLIER, TYPE, IMPLEMENTATION*}. The decision attribute is defined by the *OUTCOME*.

STEP2: Construction of Coded Information table

The construction of coded information system is done by mapping of the codes for each entry in the information table as given in the listing below:

The code for the various entities is as follows:

SUPPLIER = {1-DSC, 2-THMOSAB, 3-CODOHERTY}

TYPE = {1-E-COMMERCE, 2-ERP}

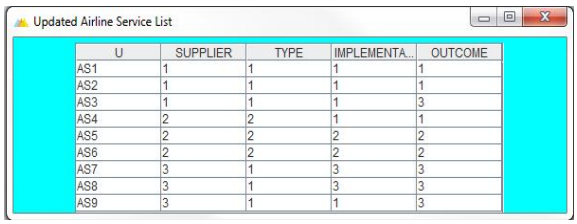
IMPLEMENTATION = {1-JAVA, 2-VB, 3- C++}

OUTCOME = {1-HR, 2-HP, 3-LC}

Where, HP= {High Performance}, HR= {High Reputation}

LC= {Low Cost}.

For the purpose of ease in computation and analysis the coded information system is generated for the information system. Each row in the coded information system corresponds to unique object or airline web service. Of all the attributes for airline service we have considered three conditional and one decision attribute.



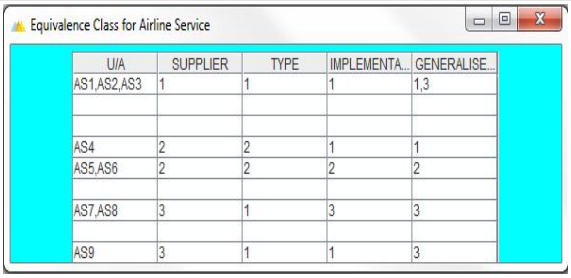
U	SUPPLIER	TYPE	IMPLEMENTA..	OUTCOME
AS1	1	1	1	1
AS2	1	1	1	1
AS3	1	1	1	3
AS4	2	2	1	1
AS5	2	2	2	2
AS6	2	2	2	2
AS7	3	1	3	3
AS8	3	1	3	3
AS9	3	1	1	3

STEP3: Generation of Equivalence class table

The Equivalence class table is generated from the coded information table by following the steps discussed in Section 2.2.

There are *5 Equivalence Class*es for Airline Service.

We see from the Fig.3 that the first row in the table has a ambiguity among the generalized decision (i.e. it could be either **"1"** or **"3"**).All other equivalence class have no such ambiguity in their generalized decision. As we can see from Fig.3 the column 'U/A' represent the equivalence class.


Equivalence Class for Airline Service

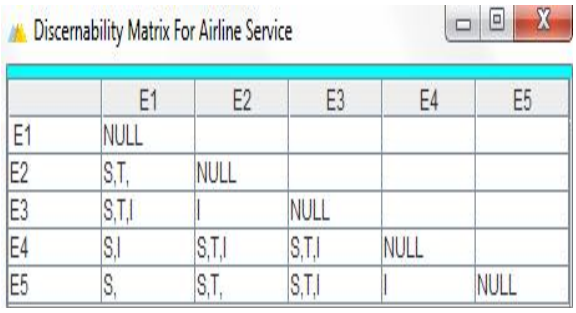
U/A	SUPPLIER	TYPE	IMPLEMENTA...	GENERALISE...
AS1,AS2,AS3	1	1	1	1,3
AS4	2	2	1	1
AS5,AS6	2	2	2	2
AS7,AS8	3	1	3	3
AS9	3	1	1	3

Fig.3. Screenshot of the Equivalence Class: Airline Service

STEP4: Generation of Discernibility Matrix

The Generation of Discernibility matrix for Airline Service is based on the steps describe in Section 2.2. The NULL value along the diagonal of the Discernibility matrix table emphasize on the fact that an equivalence class cannot be discerned from itself as show in Fig.4.As a example, E1 is discernible from E2 by the attributes{S,T},E1 is discernible from E3 by the attributes{S,T,I},E1 is discernible from E4 by the attributes{S,I},E1 is discernible from E5 by the attributes {S}.

As we can observe from the Fig.4 , if we considered the figure as a matrix then the upper triangular matrix is not filled as it same as lower triangular matrix.


Discernability Matrix For Airline Service

	E1	E2	E3	E4	E5
E1	NULL				
E2	S,T,	NULL			
E3	S,T,I	I	NULL		
E4	S,I	S,T,I	S,T,I	NULL	
E5	S,	S,T,	S,T,I	I	NULL

Fig.4.Screenshot of the Discernibility Matrix: Airline service

STEP5:Generation of Reducts Using Discernibility Function

The step is carried out as described in Section 2.2

As an example, we first define the discernibility function for each of the equivalence class using the data from Fig.4 as follows:

Discernibility function for E1= {(S,T),(S,T,I),(S,I),(S)}

Discernibility function for E2= {(I),(S,T,I),(S,T)}

Discernibility function for E3 = {(S,T,I),(S,T,I)}

Discernibility function for E4= {(I)}

Discernibility function for E5= {NULL} or no discernibility function exist for E5 equivalence class.

As an example of reduct generation the first row of Fig.5 has a reduct:

S,T,1,1,3,FIRST here,

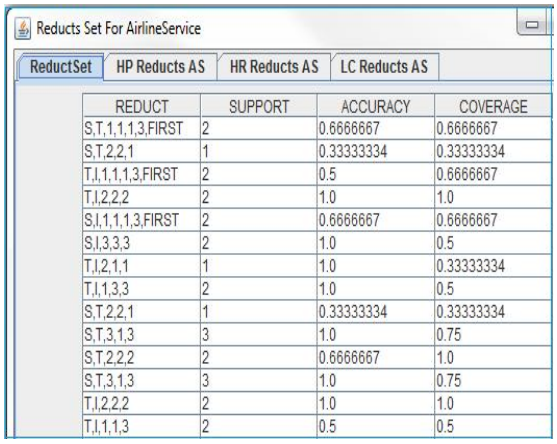
S=Supplier, T=Type, 1=Supplier code, 1=Type code, 1=Outcome code and 3=Outcome code. The *FIRST* resolves the ambiguity among outcome 1 or 3 and sets the outcome for the reduct as 1 based on the computation of *Support, Accuracy* and *Coverage.*

STEP6: Reduct classification

In Fig 5, depicting the screenshot of all the reducts, under the *ReductSet* tab, the reducts are classified into *High Performance, High Reputation and Low Cost reducts.*

The classification of reducts into different category is based on the outcome of the reduct.

Each of these 14 reducts as show in Fig.5 is classified into reducts categories as show by the tabs (HP Reducts AS,HR Reducts AS,LC Reducts AS).


Reducts Set For AirlineService

ReductSet | HP Reducts AS | HR Reducts AS | LC Reducts AS

REDUCT	SUPPORT	ACCURACY	COVERAGE
S,T,1,1,1,3,FIRST	2	0.6666667	0.6666667
S,T,2,2,1	1	0.33333334	0.33333334
T,I,1,1,1,3,FIRST	2	0.5	0.6666667
T,I,2,2,2	2	1.0	1.0
S,I,1,1,1,3,FIRST	2	0.6666667	0.6666667
S,I,3,3,3	2	1.0	0.5
T,I,2,1,1	1	1.0	0.33333334
T,I,1,3,3	2	1.0	0.5
S,T,2,2,1	1	0.33333334	0.33333334
S,T,3,1,3	3	1.0	0.75
S,T,2,2,2	2	0.6666667	1.0
S,T,3,1,3	3	1.0	0.75
T,I,2,2,2	2	1.0	1.0
T,I,1,1,3	2	0.5	0.5

Fig.5.Screenshot of full set of reducts: Airline Service

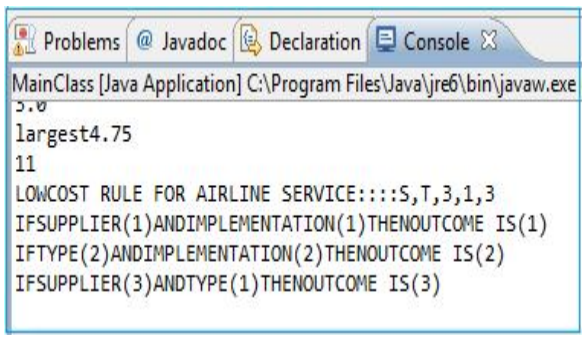
STEP7: Rule generation and optimization

The final *if- then* rules are generated by making use of supporting concepts: *Support, Accuracy and Coverage.*

There are two approaches of generating the final *if-then* rules:

One is based on the decision attribute and another is based on conditional attribute. But we chose the first approach as it gives consistent rules based on the input.

As we can see in Fig.6 the *if-then* rules are show in the console of eclipse IDE:


Problems | Javadoc | Declaration | Console

MainClass [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe

```
largest4.75
11
LOWCOST RULE FOR AIRLINE SERVICE::::S,T,3,1,3
IFSUPPLIER(1)ANDIMPLEMENTATION(1)THENOUTCOME IS(1)
IFTYPE(2)ANDIMPLEMENTATION(2)THENOUTCOME IS(2)
IFSUPPLIER(3)ANDTYPE(1)THENOUTCOME IS(3)
```

Fig.6 Screenshot of final if-then rules

Rule 1:(For High Performance)

If Supplier is 1 and Implementation is 1 *then* Outcome is 1.

Rule 2:(For High Reputation)
If Type is 2 and Implementation is 2 *then* Outcome is 2.
Rule 3:(For Low Cost)
If Supplier is 3 and Type is 1 *then* Outcome is 3.
Referring to the codes given in Section 3.2 (STEP 2 - Construction of the coded information table) the rules can be represented as:
Rule 1: If Supplier is DSC and Implementation is JAVA then Outcome is High Performance.
Rule 2: If Type is ERP and Implementation is VB then Outcome is High Reputation.
Rule3: If Supplier is CODOHERTY and Type is E-COMMERCE then Outcome is Low Cost.

3.2 Experimental results and analysis
The rough set theory as a aiding tool for genetic algorithm was analyzed by using the application that we developed in JAVA. We considered 9 web services of three categories (i.e. *Airline Service, Hotel Service, Car-rental service*).The processing of the Information table that was provided as into the application is done in 7 steps as it was described in previous section.

For each category of web service (Airline service, Hotel service and Car rental service) 3 rules (if-then) were generated. Out of the 14 reducts that were generated, 6 of them were classified as High Performance reducts, 3 were classified as High Reputation reducts and 5 were classified as Low Cost reducts. For each of these categories of reduct set the final rules were generated by relative comparison of supporting concept values: *Support, Accuracy* and *Coverage*. These *if-then* rules will be used by Genetic Algorithm as input for web service composition.

IV. CONCLUSION

As illustrated in Section 3.1 and 3.2, we have demonstrated that the initial population of genes (in this case, web services), used for genetic algorithms can be restricted by generating *if-then rules* from the input data (in this case information system), using rough set theory. This restricted or constrained input population would optimize the working of genetic algorithm making it more efficient with a higher hit rate and lesser execution time.

Thus, in this paper, an optimal approach for rough set theory in genetic algorithm for web service composition has been implemented. The approach in this paper is very straight forward and can be applied to various other domains dealing with imperfect knowledge such as fuzzy sets, evidence theory, machine learning [12], apart from web service composition. This method is easily scalable to a large number of web services. In our approach we have implemented two key concepts, in rough set theory, namely, upper and lower approximation and equivalence classes. In our future work, we suggest that the seven step process used in this work could be reduced into a more fine grained process using different set of conditional and decision attributes for web service composition.

V. REFERENCES

[1] B.Walczakand D.L.Massart,"Tutorial:Roughsets theory," Chemometrics and intelligent laboratory systems,vol.47,1999,pp.1-16.

[2] B. Srivastava and J.Konler, "Web Service Composition-Current Solutions and Open Problems,"ICAPS 2003 Workshop Planning for Web Services,2003,pp.1-8.

[3] C.Goh, and R.Law, "Incorporating the rough sets theory into travel demand analysis," TourismManagement,vol.24,no.5,2003,pp.511-517.

[4] D.Yamaguchi, G.D.Li, and M.Nagai, "A Grey-Rough Set Approach for Interval Data Reduction of Attributes",RSEISP 2007,LNAI 4585,2007,pp.400-410.

[5] G. Canfora, M.D.Penta , Raffaele Esposito and Maria Luisa Villani , "An Approach for QOS-aware Service Composition based on Genetic Algorithms,"GECCO '05 Proceedings of the 2005 conference on Genetic and evolutionary computing,pp.1069-1075.

[6] G.D.Li, D.Yamaguchi, and H.S.Lin, "A Grey-Based Rough Set Approach to Supplier Selection Problem,"RSCTC 2006,LNAI 4259,2006,pp.487-496.

[7] L.Xiangwei , X. Zhicai and Y.Li, "Independent Global Constraints-aware Web Services Composition Optimization Based on Genetic algorithm",2009 International Conference on Industrial and Information Systems,2009,pp.52-55.

[8] P.Pattaraintakorn,N.Cercone,and.Naruedomkul,"Rule learning:Ordinal prediction based on rough sets and soft-computing",Applied Mthematics Letters,vol.18,no.12,pp.1300-1307.

[9] R. Fileto, L. Liu, C. Pu, E.D. Assad, and C.B.Medeios, "POESIA: An ontological workflow approcach for composing Web services in agriculture,"The VLDB Journal,vol.12,no.4,2003,pp.352-367.

[10] W.P.Ziarko, "Rough Sets, Fuzzy Sets and Knowledge Discovery," Springer-Verlag, New York,1994.

[11] W.Y Liang , "Apply Rough Set Theory into the Web Services Composition,"22nd International Conference on Advanced InformationNetworkingand Applications,Ginowan,Okinawa,Japan,2008,pp.888-895.

[12] W.Y Liang,C.C Huang , "The generic genetic algorithm incorporates with rough set theory- An application of web services composition,"Expert Systems with Applications,Vol.36,no.3,2009,pp.5549-5556.

[13] X.Min and C.Liang, "Application of rough Set Theory in Coal Gangue Image Process,"Fifth International Conference on Information Assurance and security,Xian,China,2009,pp.87-90.

[14] Z.Pawlak, "Rough set theory and its applications,"Journal of Telecommunications and Information technology,2002.

[15] Z.Pawlak, "New look Bayes' theorem – the rough set outlook," RSTGC-2001, Matsue Shimane, Japan,May 2001, pp.1-8.

◈◈◈