World of Computer Science and Information Technology Journal (WCSIT)
ISSN: 2221-0741
Vol. 2, No. 7, 209-214, 2012

An Analytical Comparison of Rough Set Theory-based Grid Resource Matching Algorithms

Mehdi Bayat
Computer Engineering Department
Iran University of Science and Technology
Tehran, Iran

Morteza Analoui
Computer Engineering Department
Iran University of Science and Technology
Tehran, Iran

Abstract— Grid environment is an infrastructure in which many heterogeneous resources participate in solving large scale and high computational problems on the internet. Heterogeneity and large number of resources makes resource matching an important issue in the field of grid networks. Recently, rough set theory has been widely considered for efficient resource matching in the literature. In this paper, we analyze and compare three state-of-the-art rough set-based resource matching methods: ROSSE, DRSRD, and CRSRD. These methods are evaluated from the aspects of accuracy and time complexity. The evaluations show ROSSE is fast and has much better performance than DRSRD and CRSRD.

Keywords- grid computing; Resource Discovery; Rough set theory; matchmaking

I. Introduction

Grid technologies enable sharing, exchange, discovery, selection, and aggregation of geographically distributed or Internet-wide heterogeneous resources (e.g. sensors, computers, databases [1], workstations, clusters, and mainframes) with various individual properties (e.g. main memory, CPU speed, bandwidth, virtual memory, hard disk, operating system, CPU vender, number of CPU elements, and etc) where they can independently join and leave the grid environment [2]. In large scale Grid environments, resource discovery is a challenging task due to a potentially large number of resources, users and considerable heterogeneity in resource types and user requests [3]. One of the most important issues in resource matching is the existence of uncertainty among advertised resource properties. ”An uncertain property is defined as a resource property that is explicitly used by one advertised service but does not appear in another service advertisement that belongs to the same service category” [4]. Recently, rough set theory has been widely considered in developing new resource matching methods. ROSSE [4], DRSRD [5], and CRSRD [2] are three rough set based methods that can deal with uncertainty and have been examined in this paper.

The remainder of the paper is organized as follows: Section 2 is a brief description of the three aforementioned methods. Section 3 provides an analytical evaluation of time complexity of each method. Section 4 demonstrates a functional comparison of the three methods ROSSE, DRSRD and CRSRD. Finally, we conclude this analysis in section 5.

II. Overview of ROSSE , DRSRD and CRSRD

In this section we briefly describe the three state-of-the-art methods ROSSE, DRSRD and CRSRD. This will enable us to further evaluate them in terms of operation efficiency and time complexity.

A. ROSSE

Reference [4] presents a search engine for grid service discovery, called ROSSE. ROSSE is built on rough set theory and has the capability to deal with uncertainty of service properties when matching services. This is achieved by dynamically identifying and reducing dependant properties that may be uncertain properties when matching a service query [4]. It also supports semantic based matching by using an ontology repository to infer the semantic relationships of properties when matching services. The process of service discovery in ROSSE is as follow:

First, it identifies and reduces the dependant properties because some uncertain properties of advertised services may be indecisive properties. Dependent property reduction algorithm is shown in Fig. 1. Algorithm is based on (1).

$$[Y]_{P_A} = [Y]_{P_A^D} = [Y]_{P_A^{IND}} \qquad (1)$$

Where, Y is a set of objective services that are relevant to the service query and $[Y]_{P_A^D}$ is a set of objective services that are defined by the decisive properties of set P_A^D.

Input: P_A, which is a set of service properties that are relevant to a service query

Output: P_A^D, which is a set of decisive service properties that are relevant to a service query

1: for each property $P \in P_A$

2: if P is an indecisive property based on (1)

3: add P into P_A^D

4: $P_A^{IND\_Core} = \phi$

5: add P into $P_A^{IND\_Core}$

6: end if

7: end for

8: for i= sizeof (P_A^{IND})-1 to 2

9: compute all possible i combinations of the properties of P_A^{IND}

10: if any combined i properties are indecisive properties based on (1), then

11: $P_A^{IND\_Core} = \phi$

12: add the i properties into $P_A^{IND\_Core}$

13: break

14: end if

15: end for

16: $P_A^D = P_A - P_A^{IND\_Core}$

17: return P_A^D

Figure 1. Dependent property reduction algorithm of ROSSE

This algorithm shows the discovery of decisive properties in which, line 1-7 are used to identify individual indecisive properties, and line 8-15 are used to check all possible combinations of these individual indecisive properties with an aim to compute a maximal set of indecisive properties [4]. After reducing indecisive properties, it computes similarity degree of each advertised service to service query.

Let P_Q be a property used in a service query and

P_A be a property used in a service advertisement,

$dom(P_Q, P_A)$ be the degree of a match between P_Q and P_A

$\| P_Q, P_A \|$ be the semantic distance between P_Q and P_A In the terms of domain ontology Ω.

Let $S(Q,s)$ be the similarity degree of an advertised service P_Q and the service query P_A in the terms of Ω.

ROSSE defines $dom(P_Q, P_A)$ as:

$$dom(P_Q, P_A) = \begin{cases} 1 & \textit{exact match} \\ \frac{1}{2} + \frac{1}{e^{(\|P_Q, P_A\| - 1)}} & \textit{plugin match}, \|P_Q, P_A\| \geq 2 \\ \frac{1}{2 \times e^{(\|P_Q, P_A\| - 1)}} & \textit{subsume match}, \|P_Q, P_A\| \geq 1 \\ 0.5 & \textit{uncertain match} \\ 0 & \textit{no match} \end{cases} \quad (2)$$

Since every decisive property of an advertised service s has a maximal match degree with one of the properties among all the properties used in a service query, $S(Q,s)$ can be computed using the following [4]:

$$S(Q,s) = \frac{\sum_{j=1}^{L_D} \sum_{i=1}^{M} \max(dom(P_{Q_i}, P_{A_j}))}{L_D} \quad (3)$$

B. CRSRD

Ataollahi and Analoui propose a rough set theory-based method called CRSRD [2]. Like ROSSE, CRSRD can also deal with uncertainty in properties. First, it reduces dependent properties. It uses (4) to find dependent properties.

$$\alpha = \gamma(C, D) = \frac{|Pos_C(D)|}{|U|} \quad (4)$$

Here, C and D are subsets of the property set P, and α is the dependency degree of D to C. If $\alpha = 1$, D totally depends on C. According to rough set theory:

$$Pos_C(D) = \cup \underline{C}X \quad (5)$$

Where $\underline{C}X$ is the lower approximation of X in the terms of the property set C.

Dependent-properties reduction algorithm has been shown in Fig. 2. In the first step, the algorithm marks indecisive advertised properties which are relevant to the requested resource in the terms of the decision property set D and add them to Q_{DEP} which is the dependent property set. Step 2 removes all the decisive properties from Q_{DEP}. Step 3 checks all the combinations of the properties in Q_{DEP} and for any $Q_i \in Q_{DEP}$ if one of its combinations with other properties in Q_{DEP} is not indecisive, it will be removed from Q_{DEP} [2].

Input: condition attributes set Q={Q1,Q2,….,Qm}.
Input: decision attribute D.
Output: dependent attributes.
Step1:
$Q_{DEP} = \phi$;
For all $Q_i \in Q$ do
If $\gamma_Q(D) = \gamma_{Q-\{Q_i\}}(D)$ then
$Q_{DEP} = Q_{DEP} \cup Q_i$;
End for
Step 2:
If $Q_{DEP} \neq \phi$ then
For all $Q_i \in Q_{DEP}$
If $\gamma_{Q-\{Q_i\}}(Q_i) \neq 1$ then
$Q_{DEP} = Q_{DEP} - \{C_i\}$
End for
End if
Step 3:
If $Q_{DEP} \neq \phi$ then
For all $Q_i \in Q_{DEP}$
For any Combination Set of Q_i with other attributes in Q_{DEP}
If $\gamma_{Combination\_set}(D) = \gamma_{Q_{DEP}}(D)$ then
$Q_{DEP} = Q_{DEP} - Q_i$
End for
End for
Return Q_{DEP}.

Figure 2. Dependent property reduction algorithm at CRSRD[2]

After reducing the dependent advertised properties, the resource matching component calculates the degree of the match between the requested resource and the advertised resources. $m(Q_j, R_i)$ is defined as the match degree of the requested resource property R_i and the advertised resource property Q_j. In this algorithm, properties are divided into two classes: first class are the properties of type string. For this class of properties, if Q_j is an exact match with R_i, the match degree would be 1, but if Q_j is a plug-in match with R_i with a match generation of d, the match degree would be [2]:

$$\begin{cases} m(Q_j, R_i) = 1 - ((d-1)\times 0.1) & 2 \le d \le 5 \\ m(Q_j, R_i) = 0.5 & d > 5 \end{cases} \tag{6}$$

For the case of the subsume match, if Q_j is a subsume match with the match generation of d [2]:

$$\begin{cases} m(Q_j, R_i) = 0.8 - ((d-1)\times 0.1) & 1 \le d \le 3 \\ m(Q_j, R_i) = 0.5 & d > 3 \end{cases} \tag{7}$$

An advertised property with empty value is regarded as a null property. For any null property Q_j the match degree is 0.5 [2].

The second class is the property set of types other than string. For this class, the match degree is defined by [2]:

$$\begin{cases} m(Q_j, R_i) = 1 - (V(Q_j)/V(R_i)\times 0.1) & V(Q_j)/V(R_i) \le 5 \\ m(Q_j, R_i) = 0.5 & V(Q_j)/V(R_i) > 5 \end{cases} \tag{8}$$

In which $V(Q_j)$ is the value of the attribute.

For calculating the match between the requested resource and the advertised resources, we use equation (9) which calculates the maximum match degree between the requested resource and the advertised resources [2].

$$M(R_R, R_A) = \frac{\sum_{j=1}^{L_D}\sum_{i=1}^{M}\max(m(Q_j, R_i))}{L_D} \tag{9}$$

Where R_R and R_A are the requested resource and the advertised resource respectively.

C. DRSRD

Ataollahi and Analoui propose DRSRD [5] which is a dynamic rough set-based method that contains two main steps. The first step is *candidates optimization* which uses dynamic rough set theory in order to determine the optimum set of candidate resources [5]. This selected optimum set of resources are most likely to satisfy the requested service. The second step is resource matching which only applies to the optimized candidate set and ranks resources of the optimum set according to their match degree. The matching algorithm in this method is the same as the match making algorithm of CRSRD.

Suppose $A=(U,P)$ is an information system in which U is the universal set of resources and P is the property set that describes resources. Let $T\subseteq P$ and $X\subseteq U$, for any $x\in U$ we have :

$$\rho^{-}_{(X,T)}(x)=\frac{\left|[x]_T - X\right|}{\left|[x]_T\right|}\, as\; x\in X \tag{10}$$

$$\rho^{+}_{(X,T)}(x)=1-\frac{\left|[x]_T - X\right|}{\left|[x]_T\right|}\, as\; x\in \sim X \tag{11}$$

$\rho^{-}_{(X,T)}(x)$ is called the outward transfer coefficient and $\rho^{+}_{(X,T)}(x)$ is called the inward transfer coefficient of element x about T [5]. $[x]_T$ is the set of all resources that have the same value for the property set T . Outward transfer coefficient of resource x is the percent of the resources that are member of the set $[x]_T$ and are not a member of the set X and $x\in X$. Inward transfer coefficient of resource x is the percent of the resources that are a member of both the sets $[x]_T$ and X and that $x\notin X$. Let R be the requested resource properties set, the properties set T having $T\subseteq R$, is defined as bellow:

$$T=\{(r_i,w_i)\,|\,(r_i,w_i)\in R, w_i\geq 0.5\}, 1\leq i\leq L1 \tag{12}$$

In fact the set T contains properties with priority factor (weight) more than 0.5 [5]. Inward transfer standard $d^{+}_{T}(x)$ is defined as follows:

$$d^{+}_{T}(X)=\frac{\sum_{i=1}^{L1} W_i}{|T|}\; which\,(t_i,W_i)\in T \tag{13}$$

And T' is defined as a set of properties that their weight is less than 0.5 :

$$T'=\{(r_i,w_i)\,|\,(r_i,w_i)\in R, w_i< 0.5\}, 1\leq i\leq L2 \tag{14}$$

The outward transfer standard $d^{-}_{T}(x)$ is defined as bellow [5]:

$$d^{-}_{T'}(X)=\frac{\sum_{i=1}^{L2} W_i}{|T'|}\; which\,(t_i,W_i)\in T' \tag{15}$$

Inflated dynamic main set of X is defined as [5]:

$$M^{+}_{T}(X)=\{x\,|\,x\in(\sim X), d^{+}_{T}(X)\leq \rho^{+}_{(X,T)}<1\} \tag{16}$$

And contracted dynamic main set of X is defined as [5]:

$$M^{-}_{T}(X)=\{x\,|\,x\in X, d^{-}_{T}(X)\leq \rho^{-}_{(X,T)}<1\} \tag{17}$$

Candidate optimization algorithm is shown in Fig. 3.

In step 2 resources which have inward transfer coefficient greater than $d^{+}_{T}(x)$ are added to set I .

There could be a resource x' that is not a member of set X but is a member of set $[x']_P$ (the most of members of which are a member of set X). Considering that the members of set $[x']_P$ are similar resources, there is a high chance that resource x' can be a member of set X . In fact, resources like x' are found and added to set I .

Furthermore in this step, resources which have an outward transfer coefficient less than $d^{-}_{T}(x)$ are added to set C . There can be a resource x'' that is a member of set X which is also a member of set $[x'']_P$ (the most of members of which are not members of set X). Considering that members of set $[x'']_P$ are similar resources, there is a high probability that resource x'' won't be a member of the set X . Resources like x'' are found and add to the set C .

In step 3, we remove the resources of set C from set X and add the resources of set I to set X . The result set X^* is the two direction dynamic set of X according to T and T' .

In step 4, we calculate the lower approximation set X^* ($\underline{X^*}$) according to the requested resource properties set R . In fact $\underline{X^*}$ is the set of resources that are most likely to be selected as the matched resources [5].

III. TIME COMPLEXITY ANALYSIS

In this section, we evaluate the time-complexity of the three mentioned algorithms.

Input: requested properties set R= {(r1, w1), (r2, w2), …, (rL,wL)}.

Input: candidates set X.

Output: candidates optimized set

I : Inflated dynamic main set of X about T .

C : contracted dynamic set of X about T' .

X^* : Two direction dynamic set of X according to the T and T' .

$\underline{X}^*$: Lower approximation of X^* according to requested resource properties R .

Step 1:

Compute $d_T^+(x)$ and $d_T^-(x)$.

Step 2:

For all $x \notin X$

If $\rho^+_{(X,T)}(x) \ge d_T^+(X)$

Add x to the I .

End for.

For all $x \in X$

If $\rho^-_{(X,T)}(x) \ge d_T^-(X)$

Add x to the C .

End for.

Step 3:

$X^* = (X - C) \cup I$.

Step 4:

Compute $\underline{X}^*$ according to the R .

Return X^* .

Figure 3. Candidate optimization algorithm

Let P be the number of all properties that describe resources,

P_C the number of conditional properties which are used to construct equivalence classes on the universe set of resources,

P_{dep} as the number of properties that are dependent on a subset of properties set,

R as the number of all available resources, and

cls as the number of equivalence classes ($[x]_{P_C}$) on the universe of the set of resources.

We will have $R >> P \ge P_C > P_{dep}$, $R >> cls$.

A. CRSRD

The time complexity of constructing equivalence classes is $\theta(R \times P_C \times cls)$. Using a temporary table that contains the values of the conditional properties and the values of the decision properties for each class, the time complexity of the first step of the property reduction algorithm would be $O(P_C^2 \times cls^2)$, the time complexity of the Second step of the property reduction algorithm would be $O(cls^2 \times P_C \times P_{dep})$ and the time complexity of the third step of the property reduction algorithm would be $\theta(R \times P_{dep} \times cls)$. Matching the request with all the resources would cost $\theta(R \times P^2)$. So the overall time complexity of this method is $\theta(R \times (P^2 + P_C \times cls))$.

B. DRSRD

The time complexity of constructing equivalence classes is $\theta(R \times P_C \times cls)$. The time complexity of the first step of the candidates optimization algorithm is $\theta(P_C)$. Using a temporary table that contains $\rho^-_{(X,T)}(x)$ and $\rho^+_{(X,T)}(x)$ for each class, the cost of the second and third step of the candidates optimization algorithm would be $\theta(R^2)$ altogether. Using the same table, the time complexity of the fourth step of the candidates optimization algorithm would be $\theta(cls)$. Matching the request with all the resources would have a time complexity of $\theta(R \times P^2)$. Therefore the overall complexity of this approach is $\theta(R^2)$.

C. ROSSE

The time complexity of constructing equivalence classes is $\theta(R \times P \times cls)$. The first loop in its property reduction algorithm is very similar to the first step of the property reduction algorithm of CRSRD, so the cost of this section is $O(P_C^2 \times cls^2)$. The time complexity of the second loop in the property reduction algorithm is $O(2^{dep} \times cls^2 \times P)$. Matching the request with all the resources would cost $\theta(R \times P^2)$. So the total time complexity of ROSSE is $\theta(R \times P \times (P + cls))$.

D. Experimental Results

In this section we evaluate the performance of CRSRD, DRSRD, and ROSSE with respect to the accuracy of ranking available resources. In our simulations we assume a resource repository of six resources as shown in Table I. Each resource has four properties namely the number of CPU cores, the CPU clock, RAM size, and the Disk capacity respectively. We also assume that the range of each property is as bellow:

Feature 1 :{ 1, 2, 4}

Feature 2 :{ 0.133, 0.166, 0.233, 0.3, 0.333, 0.45, 0.533, 0.733, 1, 1.33, 1.4, 1.7, 1.8, 2, 2.2, 2.4, 2.53, 2.66, 2.83, 3, 3.2, 3.4, 3.6}

Feature 3 :{ 32, 64, 128, 256, 384, 512, 768, 1024, 1536, 2048, 2560, 3072, 3584, 4096}

Feature 4 :{ 5, 10, 20, 30, 40, 60, 80, 120, 160, 200, 250, 320, 400, 500, 750, 1000}

TABLE I. RESOURCE REPOSITORY

Resource No.	CPU Core	CPU Clock	RAM Size	Disk Capacity
1	4	2.66	4096	1000
2	4	2.83	3584	750
3	2	3.2	3072	750
4	2	2.4	2048	500
5	1	1.8	1024	160
6	1	0.45	128	20

We have also compared ROSSE, CRSRD and DRSRD from the aspect of efficiency of matchmaking. Suppose a request of {4, 2.4, 3584, 500} has been sent. These three methods rank the six resources to meet the request as shown in Table.II .

TABLE II. RESOURCE MATCHING RESULTS

ROSSE		CRSRD		DRSRD	
Res No.	*Score*	*Res No.*	*Score*	*Res No.*	*Score*
1	0.935	6	0.987	6	0.987
2	0.909	5	0.96	5	0.96
3	0.63	4	0.921	4	0.921
4	0.64	3	0.895	3	0.895
5	0.068	2	0.883	2	0.883
6	0.046	1	0.869	1	0.869

As you can see, resources #1 and #2 are more powerful than other resources and they can satisfy the request to a better degree than other resources. So we expect a higher score and better rank for these two resources. Resources #5 and #6 are two weak resources and they are not a good match for the requests which need a powerful resource. Resource #3 and #4 are comparatively two average resources.

ROSSE relatively ranks advertised resources in a reasonable way. The request needs a powerful resource and this method first suggests resources #1 and #2, then the two average resources (#3 and #4) and at last the two weak resources (#5 and #6).

As you can see, CRSRD and DRSRD have not ranked the advertised resources properly. These two methods first suggest the two weak resources then the average resources and at last the two powerful resources. Considering that the request needs a powerful resource, this is not a proper ranking for advertised resources. That is because of a problem which exists in their formulation of the match degree. As mentioned, the formula of the match degree between a non-string request property and a non-string advertised property for CRSRD and DRSRD is as in (8).

Suppose Q_1 and Q_2 be two available resources, for each available resource property $V(Q_{1j}) > V(Q_{2j})$ we have $m(Q_{1j}, R_i) < m(Q_{2j}, R_i)$. So it's obvious that using such formulation would lead to a higher score for weak resources and less score for powerful resources. As a result, weak resources get a better ranking than more powerful resources.

CRSRD and DRSRD use similar formula for computing the match degree. As mentioned, the difference is that DRSRD selects an optimized set of resources before suggesting them to meet the request. As DRSRD can't detect the proper resource for the request, choosing an optimized subset of resources would not improve the performance of this method.

IV. CONCLUSION

In this paper we analyzed and compared three state-of-the-art rough set theory-based resource matching methods ROSSE, CRSRD and DRSRD. Time complexity analysis of these methods shows that ROSSE and CRSRD have similar time complexities but they are faster than DRSRD. Our experimental results show that the ranking process in ROSSE is more effective in resource matching than the other two methods. We also analyzed the reasons and addressed the issue which exists in the match degree formulation of CRSRD and DRSRD which justifies the correctness of the experimental results.

REFERENCES

[1] Foster and C. Kesselman, The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 259-278, 1999.

[2] I. Ataollahi and M. Analoui, "Resource discovery using rough set in Grid environment," in 14th International CSI conference (CSICC2009), Tehran, 2009.

[3] K. R. Bindu and S. M. Bhanu, "Group Aware Semantic Matchmaker for Grid Resource Discovery," in International Conference on Advances in Recent Technologies in Communication and Computing, 2009.

[4] M. Li, B. Yu, O. Rana, and Z. Wang, "Grid Service Discovery with Rough Sets," IEEE transactions on knowledge and data engineering, vol. 20, June 2008.

[5] I. Ataollahi and M. Analoui, "Resource Matchmaking Algorithm using Dynamic Rough Set in Grid Environment," (IJCSIS) International Journal of Computer Science and Information Security., vol. 4, 2009.

[6] Pawlak Z, "Rough sets," in Int. Jour. on Computer and Information Science, 11(5): PP.341-356, 1982.

[7] Dong Ya Li , Bao Qing Hu, "A Kind of Dynamic Rough Sets," in Proc. 5th Int. Conf. on Fuzzy Systems and Knowledge Discovery, Vol.3, p.79-85, August, 2007.

[8] Xiaobing Pei, YuanZhen Wang, "an Approximate Approach to Attribute Reduction", in Int. Jour. on Information Technology, VOL. 12, NO.4, 2006.

[9] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron, Rough Sets: a tutorial, Rough Fuzzy Hybridization, Springer, pp. 3-98, 1999.

[10] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara. Semantic matching of web service capabilities. Proceedings of 1st International Semantic Web Conference. (ISWC2002), Berlin, 2002.