International Journal of Database Theory and Application
Vol.10, No.5 (2017), pp.59-68
http://dx.doi.org/10.14257/ijdta.2017.10.5.06

Dynamic QoS evaluation for Web Services Using Data Envelopment Analysis on Real-time Status

Luda Wang[1,2] and Peng Zhang[2]
[1]*School of Information Science and Engineering, Central South University, Changsha, China*
[2]*Xiangnan University, Chenzhou, China*
wang_luda@163.com, mimazp@126.com

Abstract

Service run-status monitoring can provide treal-time status for service QoS evaluation as service properties. In this work, dynamic QoS evaluation for Web services are based on DEA. Proposed methods could be used to analyze real-time status of Web services. DEA-based service performance evaluation is implemented by a multi-objective model, and DEA-based service QoS evaluation is implemented by a multi-objective model with critical real-time status performance. Both models are effective depending on particular argument and validation. Dynamic QoS evaluation for Web services are based on DEA could provide performance and QoS information to service composition.

Keywords*: Web Service, Service composition, QoS, DEA, Real-time Status*

1. Introduction

QoS evaluation for Web services is to analyze the decision value of describing the quality of service to reflect the difference of service quality among different service composition components, and to provide reference for service invoker. The decision value of QoS is needed to construct the QoS dynamic evaluation decision process of the service to realize the evaluation model.

Service QoS evaluation requires QoS-related service properties. Service run-status monitoring [1] provides the relevant real-time status for service QoS evaluation as service properties.

In service composition, an effective method of service QoS service composition should focus on factors that relate to the needs of the invoker. QoS needs to reflect the service to meet the service invoker's relative degree of demand. Service QoS evaluation focus on a comprehensive analysis of services performance and critical real-time status.

The QoS of service means that the service can meet the needs of the invoker. Service QoS evaluation will comprehensively examine the service properties related to the invoker's needs, including the performance of the service and the real-time status. The service selection based on QoS depends on the satisfaction of multiple service properties. Service QoS properties is a multi-objective decision-making problem, or multi-objective optimization [2]. Construction of service QoS decision process can achieve the evaluation model. It needs perceiving service objectives as decision-making objects, and its decision-making process will involve performance evaluation.

The research work will first build the decision-making process of service performance evaluation. Then, based on it, the service QoS evaluation decision process with performance and critical real-time status is constructed. The two decision-making processes are multi-objective decision-making, using multi-objective mathematical programming [3] method, specifically based on data envelopment analysis (DEA) [2] theory.

ISSN: 2005-4270 IJDTA

In the follow-up, the multi-objective setting of service QoS evaluation and the multi-objective decision-making of DEA-based service QoS evaluation will be discussed in detail.

2. Multi-object Setting of Service QoS Evaluation

The purpose of Service QoS dynamic evaluation is to provide a clear reference value to the service invoker, called the decision value. Therefore, it is necessary to define the mathematical programming objectives (indicators) for calculating the decision values according to the service properties. Real-time status is service properties acquired by the Web service run-status monitoring tool [1].

On the basis of the real-time status, the evaluation mathematical programming problem of service QoS is to provide a clear evaluation results to the service invoker, that is, decision value. The mathematical programming needs to formalize the system of the evaluation object composed of each target service, and set the real-time status as the objects of the mathematical programming.

Real-time status include the service IP address and port number, the invoker IP address and port number, Throughput, Latency and Connections, the Availability, System CPU utility and memory usage rate, [1]. The real-time status associated with the service QoS dynamic evaluation of the mathematical programming problem is shown in Table 1.

Table 1. Service Properties in Evaluated Object System

Service real-time status acquired by the Web service run status monitoring	
Loads	Overheads
Throughput	Latency
Connections	Memory usage rate
	System CPU utility
	Availability

In Table 1, the real-time status are service properties that is associated with the mathematical programming problem, all of which are objects for service QoS dynamic evaluation. Therefore, the mathematical programming of service QoS dynamic evaluation is multi-objective mathematical programming. Because the objects of service QoS evaluation assume different load and overhead tendentiousness, and loads effect on overheads, its decision-making process should build a fractional programming method [2] of multi-objective decision.

3. Proposed Program

DEA [2] is a method of evaluating the relative validity (called DEA effective) between ‘department’ or ‘unit’ with multiple inputs and outputs using mathematical programming. As the evaluated targets, the "department’ or "unit" are called the decision-making unit (DMU) [2]. The initial DEA model C^2R is a fractional programming that transforms the fractional programming into an equivalent linear programming problem [4] [5] [6].

According to the mathematical planning problem of service QoS evaluation and the service properties of the evaluation object system, the multi-objective decision making of service QoS evaluation will construct the fractional programming model based on DEA theory. The construction of the model is based on analogy of the C^2R model [2], and the fractional programming model of the object system is evaluated according to the service attributes in Table 1.

3.1. Multiple Loads-overheads Model of services

The mathematical programming problem of the C^{2}R model is that DMUs are comparatively comparable to each of the *n* DMUs. Each evaluation object unite has *m* ‘inputs’ (representing the consumption of the DMU for the ‘resources’), and *s* ‘outputs’ (which are the ‘outcomes’ of decision units that, after consuming "resources"). A general understanding of input and output is that the smaller inputs and larger output the better [4].

Based on the DEA theory, the multiple decision-making objects of service QoS evaluation are to evaluate the service properties of the object system. Compared with the C^{2}R model [4], as shown in Fig. 1, the DEA fractional programming model has *n* target services (regarded as DMU). The ‘loads’ of the *m* types (indicating the invoking pressure on the target service), and the ‘overheads’ of the *s* types (indicating the consumption of the target service under the invoking pressure). The ‘loads’ of the service result in the ‘overheads’.

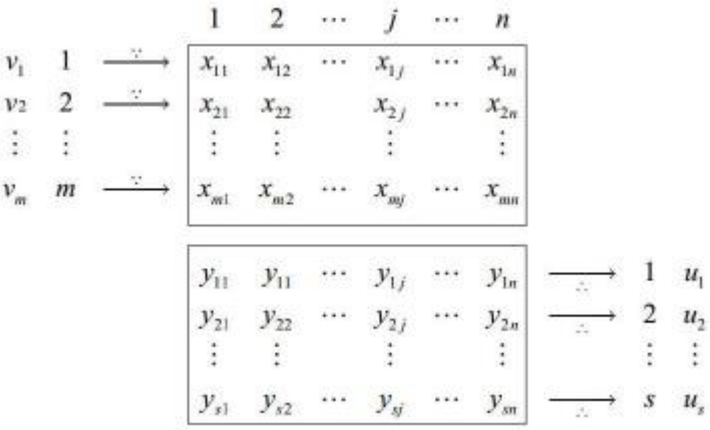
1 2 ⋯ j ⋯ n
v_1 1
v_2 2
v_m m
x_{11} x_{12} ⋯ x_{1j} ⋯ x_{1n}
x_{21} x_{22} x_{2j} ⋯ x_{2n}
x_{m1} x_{m2} ⋯ x_{mj} ⋯ x_{mn}
y_{11} y_{11} ⋯ y_{1j} ⋯ y_{1n}
y_{21} y_{22} ⋯ y_{2j} ⋯ y_{2n}
y_{s1} y_{s2} ⋯ y_{sj} ⋯ y_{sn}
1 u_1
2 u_2
s u_s

Figure 1. Multiple Loads and Overheads of DMU services

In Figure. 1, evaluation target service *j* is denoted by DMU$_j$, $1 \le j \le n$; x_{ij} is the *i-th* load of DMU$_j$, $x_{ij} > 0$; y_{rj} is the index of the *r-th* overhead of DMU$_j$, $x_{ij} > 0$. $i = \{1，2，…，m\}$, $j = \{1，2，…，n\}$, $r = \{1，2，…，$s$\}$. v_i is the weight to the *i-th* load, and u_r is the weight to the *r-th* overhead. To simplify the representation, the DMU system can be recorded as equation (1).

$$\begin{aligned} X_i &= \left(x_{1j}, x_{2j}, \cdots, x_{mj}\right)^T, \ j = 1, \cdots, n, \\ Y_i &= \left(y_{1j}, y_{2j}, \cdots, y_{sj}\right)^T, \ j = 1, \cdots, n, \\ v &= (v_1, v_2, \cdots v_m)^T, \\ u &= (u_1, u_2, \cdots u_s)^T. \end{aligned} \tag{1}$$

In this case, X_j and Y_j are the ‘load’ vector and ‘overhead’ vectors for DMU$_j$, respectively. X_j and Y_j are known data, and can be obtained from the service properties in table 1. *v* and *u* are weight vectors to terms of *m* inputs and *s* outputs, respectively. *v* and *u* are variable vectors.

Without considering the service critical real-time status, according to the concept of software performance [7] [8] [9], the relative efficiency of service multiple loads and overheads system reflects the performance of the target service. A common understanding of loads and overheads about software performance is that the larger loads and smaller overheads the better. As a result, the multiple loads-overheads model of the service (DMU), which is modeled after the C^{2}R model [4], is built according to the fractional programming. The efficiency index of the target service j (that is, DMU$_j$) is shown in the efficiency evaluation (2), where weight vector $v \in E^m$ and $u \in E^s$.

$$h_j = \frac{v^{\mathrm{T}} X_j}{u^{\mathrm{T}} Y_j}, \quad j=1,\cdots,n \tag{2}$$

Existing appropriate weight vectors v and u make $h_j \leq 1$，$j = \{1，2，\ldots，n\}$. The efficiency index h_j means the ratio of ‘loads’ to ‘overheads’, on condition that loads is $v^T X_j$, and overheads is $u^T Y_j$. It is written for convenience as equation (3) ($1 \leq j_0 \leq n$).

$$X_0 = X_{j_0},\ Y_0 = Y_{j_0} \tag{3}$$

To evaluate the efficiency of DMU_{j_0}, aiming on the efficiency index of DMU_{j_0},

$$h_{j_0} = \frac{v^{\mathrm{T}} X_0}{u^{\mathrm{T}} Y_0}. \tag{4}$$

and subjected to efficiency index of target service DMU$_j$,

$$h_j = \frac{v^{\mathrm{T}} X_j}{u^{\mathrm{T}} Y_j} \leq 1, \ j = 1,\cdots,n\,,, \tag{5}$$

we construct fractional programming, multiple loads and overheads model of services as follow.

$$(\text{Load-Overhead})^I \begin{cases} \max \dfrac{v^T X_0}{u^T Y_0} = V_P^{I^P}, \\ \dfrac{v^T X_j}{u^T Y_j} \leq 1, \quad j = 1,\cdots,n, \\ v \geq 0, \quad u \geq 0 \end{cases} \tag{6}$$

Fractional programming problem (Load-Overhead)^I has software and engineering background. It defines the efficiency of the science and engineering to a multiple loads and overheads system of service (DMU). The relative efficiency is perceived as target service performance. In the condition of (Load-Overhead)^I, the solution of the efficiency index in equation (4) is DEA-based evaluation decision process of the service performance.

On consider of requirements for service invocations, performance and real-time statues are considerable. Latency is the critical real-time status that is preferred by the service invokers.

Considering the critical real-time status, according to the concept of QoS, relative efficiency of service multiple loads and overheads system with real-time status preference, meet the requirements for service invocations. As a critical real-time status for a service invokers, latency is much more important than other decision object.

Based on the (Load-Overhead)^I model, a multiple loads and overheads model for requirements for service invocations is built in fractional programming.

To evaluate the efficiency of DMU_{j_0}, aiming on the efficiency index of DMU_{j_0}, and subjected to efficiency index of target service DMU$_j$, we construct fractional programming, multiple loads and overheads model of requirements for service invocations as follow.

$$(\text{QoS\_ Load-Overhead})^{I}\begin{cases} \max \dfrac{v^{T}X_{0}}{u^{T}Y_{0}} = \overset{Q}{V}{}_{P}^{I}, \\ \dfrac{v^{T}X_{j}}{u^{T}Y_{j}} \leq 1, \quad j = 1,\cdots,n, \\ \\ v \geq 0, \quad u \geq 0, \\ \begin{pmatrix} v \\ \\ u \end{pmatrix} \in W \end{cases} \tag{7}$$

$W \subseteq E_{+}^{m+s}$ is closed convex cone, Int W旲 . In the multi-objective decision problem, the closed cone W embodies the preference for the importance of decision objects.

Fractional programming problem (QoS_Load-Overhead)^{I} has the characteristics of service composition technology background. It defines the efficiency of the science and engineering to a preference-attached multiple loads and overheads system of service (DMU). The relative efficiency is calculated as QoS of service. In equation (4), the solution of the efficiency index is the QoS evaluation decision of service DMU$_{j}$ for requirements.

The evaluation decision process of the service performance and QoS evaluation decision process of service for requirements are constructed in (Load-Overhead)^{I} and (QoS_Load-Overhead)^{I} respectively.

3.2. Multi-objective Decision Argumentation of Service QoS Evaluation

In DEA theory, the fractional programming are equivalent in the form of linear programming. That can prove fractional programming implementation of the decision making process is effective [10] [4]. A fractional programming efficiency index can be solved by the equivalent linear programming solution.

The multi-objective decision-making process of service performance evaluation is proved for effectiveness as follow.

Using Charnes-Cooper transformation [11][12][13], fractional programming (Load-Overhead)^{I} is transformed in a equivalent linear programming.

Make

$$t = \frac{1}{u^{\mathrm{T}}Y_{0}}, \quad \omega = tv, \quad \mu = tu \,, \tag{8}$$

where $u^{\mathrm{T}}Y_{0} > 0, \; t > 0$, then objective function is

$$\frac{v^{\mathrm{T}}X_{0}}{u^{\mathrm{T}}Y_{0}} = \omega^{T}X_{0} \,, \tag{9}$$

and subjected to

$$\begin{aligned} &\frac{\omega^{\mathrm{T}}X_{j}}{\mu^{\mathrm{T}}Y_{j}} = \frac{v^{\mathrm{T}}X_{j}}{u^{\mathrm{T}}Y_{j}} \;?\; 1, \;\; j \quad 1,\cdots,n \\ &\omega, \mu\; ^{3} \;\; 0 \end{aligned} \, . \tag{10}$$

Because equation (8), $\mu^{T}Y_{0} = 1$. Then, fractional programming (Load-Overhead)^{I} is transformed in

$$(\, P_{\text{Load-Overhead}}^{I}) \begin{cases} \max \omega^{T}X_{0} = \overset{P}{V}{}_{\text{Load-Overhead}}^{I} \\ \mu^{T}Y_{j} - \omega^{T}X_{j} \geq 0, \quad j = 1,\cdots,n \\ \mu^{T}Y_{0} = 1, \\ \omega \geq 0, \quad \mu \geq 0 \end{cases} \, . \tag{11}$$

（$P^I_{\text{Load-Overhead}}$） is a equivalent linear programming of (Load-Overhead)^I, proving fractional programming (Load-Overhead)^I is effective.

The multi-objective decision-making process of service QoS for requirements is proved for effectiveness as follow.

Similarly, using Charnes-Cooper transformation [11][12][13], fractional programming (QoS_Load-Overhead)^I can be transformed in equation (13), omitting the process.

$$(P^I_{\text{QoS\_Load-Overhead}})\begin{cases} \max \omega^T X_0 = \overset{Q}{V}{}^I_{\text{QoS\_Load-Overhead}}, \\ \mu^T Y_j - \omega^T X_j, \quad j = 1,\cdots,n, \\ \qquad \mu^T Y_0 = 1, \\ \qquad \omega \geq 0, \quad \mu \geq 0 \\ \qquad \begin{pmatrix} \omega \\ \\ \mu \end{pmatrix} \in W \end{cases} \tag{13}$$

（$P^I_{\text{QoS\_Load-Overhead}}$） is a equivalent linear programming of (QoS_Load-Overhead)^I, proving fractional programming (QoS_Load-Overhead)^I is effective. In fractional programming (QoS_Load-Overhead)^I, efficiency index $\overset{Q}{V}{}^I_P$ can be solved by $\overset{Q}{V}{}^I_{\text{QoS\_Load-Overhead}}$ solution in the equivalent linear programming. The closed cone W embodies the preference for the importance of decision objects.

On the basis of equivalent linear programming（$P^I_{\text{Load-Overhead}}$） and（$P^I_{\text{QoS\_Load-Overhead}}$）, two presented multi-objective decision-making processes are effective.

4. Experiment and Result

The experiments of service QoS evaluation methods is performed using simulation with the service described in the dataset OWLS-TC v2 [14], which simulates the corresponding environment. The simulated Web services have 7 classes and 578 services. There are 28 queries that represent specific user functional requirements.

We allocate one virtual machine installation environment for each Web service. It limits the system resources of overheads, and implements the Web services simulation release on top of Windows Server. Invocation requests are implemented by program simulation to perform the load-overhead simulation of the service.

The effectiveness of the service performance evaluation method will be analyzed in a way that compares the DEA-based service performance evaluation with the results of service pressure test. Service pressure tests refer to the software pressure method [15]. It treat a Web service as an AUT (Application Under Test) [16]. As the service invoking request load increasing, when cost reach the upper limit of system resource or service is overtime (latency < 30000ms), it gets the service maximum capacity, that is the actual statistical result of service performance (unit for 1000 concurrent/s). DEA-based service performance evaluation implements multiple loads and overheads model (Load-Overhead)^I shown in equation (6).

In the simulation environment of Web service loads and overheads, there are 578 services. DEA-based service performance evaluation and service pressure test performance results are shown in Figure 2.

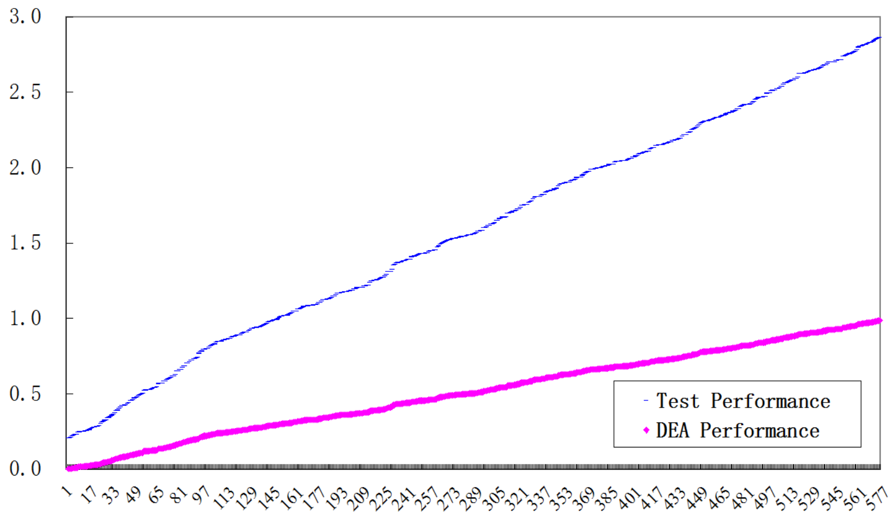

Figure 2. DEA-based Performance and Pressure Test Performance of Services

In Figure 2, n=578. Decision values of DEA-based service performance evaluation, DEA Performance are series $\{P_n\}=\{p_j \mid j=1,2,\cdots,n\}$ （$0\le p_j\le 1$）, which are in the descending order. Service pressure test performance, Test Performance are series $\{P_n'\}=\{p_j' \mid j=1,2,\cdots,n\}$, in which services (DMU$_j$) are in order of series $\{P_n\}$.

The linear fitting degree of $\{P_n\}$ and $\{P_n'\}$ is in the small deviation range, then the DEA Performance of service is basically consistent with the Test Performance. It proves that The service performance evaluation implemented by the service multiple loads and overheads model decision process is valid.

Effectiveness analysis of the DEA-based service QoS evaluation method is carried out in the manner of analyzing service latency and performance. The ratio of service QoS meeting invoking requirements in latency and performance is the Select Accuracy. Invokers selects services based on the higher service QoS decision value. Service selection condition is service performance is higher than the median value of services in the same class, and service latency is superior to the median value of services in the same class.

DEA-based service QoS evaluation is implemented by model (QoS_Load-Overhead)^I shown in equation (11). In decision-making process of fractional programming (QoS_Load-Overhead)^I, closed convex cone W is assigned by equation (13), realizing invokers perform the critical real-time status latency.

$$W=\left\{\begin{pmatrix}\omega\\ \\ \mu\end{pmatrix}\left|\begin{pmatrix}\infty & \infty & \infty & -1 & \infty & \infty & \infty\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ \infty & \infty & \infty & -1 & \infty & \infty & \infty\\ -t_1 & -t_1 & -t_1 & -1 & -t_1 & -t_1 & -t_1\\ \infty & \infty & \infty & -1 & \infty & \infty & \infty\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ \infty & \infty & \infty & -1 & \infty & \infty & \infty\end{pmatrix}\begin{pmatrix}\omega_{Throughput}\\ \omega_{Connections}\\ \\ \mu_{Latency}\\ \mu_{Memory-rate}\\ \mu_{CPU-rate}\\ \mu_{Availability}\end{pmatrix}\ge 0\right.\right\} \qquad (13)$$

In equation (13), $(\omega,\ \mu)^T \ge 0$ means each element in the vector is greater than 0.

To show the preference for latency for the service invoker, set $t_1 = 10$ in closed convex cone W. Based on the service selection condition previously, the DEA-based service QoS evaluation the Select Accuracy are shown in Table 2.

Table 2. The Select Accuracy on QoS Evaluation

Top5 select accuracy	Top10 select accuracy	Top20 select accuracy	Average select accuracy
95%	87.2%	66.7%	52.4%

By the experiment of DEA-based service QoS evaluation method, on the basis of Selection Accuracy, service multiple loads and overheads model with real-time status preference is valid.

5. Conclusion

In this work, dynamic QoS evaluation for Web services are based on DEA. Proposed methods could be used to analyze real-time status of Web services.

DEA-based service performance evaluation is implemented by multiple loads and overheads model (Load-Overhead)I, and DEA-based service QoS evaluation is implemented by model (QoS_Load-Overhead)I. Both models are effective depending on particular argument and validation.

Dynamic QoS evaluation for Web services are based on DEA could provide performance and QoS information to service composition.

Acknowledgements

This project was supported by A Project Supported by Scientific Research Fund of Hunan Provincial Education Department (No. 15C1288).

References

[1] Predic8 GmbH. Membrane Service Proxy [EB/OL] Moltkestr. http://www.membrane-soa.org/service-proxy/

[2] T, J, Coelli, D.S. Prasada Rao, C.J. O'Donnell, *et al*., "Data Envelopment Analysis", An Introduction to Efficiency and Productivity Analysis, **(2005)**, pp. 161-181.

[3] E L. Hannan "On the efficiency of the product operator in fuzzy programming with multiple objectives", Fuzzy Sets and Systems, **(1979)**, vol. 2, no. 3, pp. 259-262.

[4] Q. Wei, "Data envelopment analysis", Science Press, Bejing, **(2004)**.

[5] W, J. Wu, "Rendering a Decision - Making Unit DEA Efficient by Changing Its Outputs only", Systems Engineering, vol. 13, no. 2, **(1995)**, pp. 17-20.

[6] C. Shi, J. Chen, Y. D. Zhang, "Crossing-evaluation on Logistics Company Performance", Systems Engineering, vol. 28, no. 1, **(2010)**, pp. 47-52.

[7] C. Symons, "Software industry performance: What you measure is what you get", Software, IEEE, **(2010)**, vol. 27, no. 6, pp. 66-72.

[8] M. Atkins, R. Subramaniam, "PC software performance tuning", Computer, vol. 8, **(1996)**, pp. 47-54.

[9] M. Woodside, G. Franks, D.C. Petriu, "The future of software performance engineering", Future of Software Engineering, FOSE'07. IEEE, **(2007)**, pp. 171-187.

[10] M. Ying, "The stability of multi-objective mathematical programming", Acta Mathematica Sinica, vol. 24, no. 3, **(1981)**, pp. 321-330.

[11] C. A. K. Lovell, L. C. Walters, L. L. Wood, "Stratified models of education production using modified DEA and regression analysis", Data Envelopment Analysis: Theory, Methodology, and Applications. Springer Netherlands, **(1994)**, pp. 329-351.

[12] L. Carosi, L. Martein, "Some classes of pseudoconvex fractional functions via the Charnes-Cooper transformation", Berlin: Springer Berlin Heidelberg, **(2006)**.

[13] J.C. Chen, H.C. Lai, S. Schaible, "Complex fractional programming and the charnes-Cooper transformation", Journal of optimization theory and applications, vol. 126, no. 1, **(2005)**, pp. 203-213.

[14] M. Klusch, B. Fries, K. Sycara, “Automated semantic web service discovery with OWLS-MX”, Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems. New YorkACM, **(2006)**, pp. 915-922.

Authors

Luda Wang, he received the M.S. degree of Computer Application Technology from Hunan University in 2009, and he is currently a Ph. D. candidate of Computer Science and Technology in Central South University (CSU). His research interests include AI, Data Analysis and Information Retrieval.

Peng Zhang, she received the M.S. degree of Computer Application Technology from Hunan University in 2010. She has been a faculty member of Computer Science at Xiangnan University, China, since 2004, where she is currently a lecturer. Her research interests include Information Retrieval and Information Fusion.