Multi-objective Service Monitoring Rate Optimization using Memetic Algorithm

Pan He, Junhao Wen, Kaigui Wu and Peng Li
College of Computer Science, Chongqing University, Chongqing, China
Email: {hepan, jhwen, kaiguiwu, cheney}@cqu.edu.cn

Haijun Ren
School of Software Engineering, Chongqing University, Chongqing, China
Email: jhren@cqu.edu.cn

Abstract— In dynamic service-oriented environment, service monitoring could provide reliability improvement to service composition as well as cost increase. To reduce the overall cost brought by monitoring, existing literatures proposed to decrease the number of monitors through monitoring the most reliability-sensitive services. However, the optimal monitoring rate for those monitors was not taken into account at the same time. Aiming at choosing optimal monitoring rate for minimal number of monitors, this paper proposed to search appropriate monitoring rate to minimize multi kinds of resources cost by monitoring under reliability constraints. Firstly, two multi-objective optimization problems were presented with the reliability and cost models of service composition under monitoring analyzed through Markov chain. Then a multi-objective memetic algorithm (MOMA) was used to search the near-optimal solutions of monitoring rate for services. This algorithm employed nondominated sorting genetic algorithm II (NSGA II) [4] in the global search process and iterative random walk with direction exploitation method as local search operator. Experimental studies results showed that multi-objective approach for service monitoring rate optimization could provide solutions with a variety of trade-offs between the system reliability and cost comparing with existing greedy sensitivity-based method. Comparison with other multi-objective evolutionary algorithms showed that, in terms of both the coverage rate and hypervolume indicator, MOMA searched more effectively than several state-of-art algorithms including NSGA II, PHC-NSGA-II and HaD-MOEA.

Index Terms—service monitoring; Markov chain; software reliability; memetic algorithm; multi-objective optimization

I. INTRODUCTION

Although the development techniques for service-oriented architecture have been extensively investigated, how to improve the reliability of service composition has not been comprehensively studied. In traditional software systems, testing time increase and redundant components configuration are two common ways to improve system reliability [1]. In dynamic service-oriented environments, it is difficult to test all services in a composition in advance, so monitoring is often used to improve service reliability [2]. Like traditional reliability assurance methods, monitoring mechanism also increases system cost, so it may be advantageous to achieve a balance between reliability improvement and cost increase. Since a service-oriented system typically comprises of several services, a natural question is how to deploy monitors on them and how to optimize the monitoring rates.

Our previous work [3] employed a greedy method to select the most reliability-sensitive services in a composition to monitor but it couldn’t guarantee the minimal monitoring rate for these monitors at the same time. In this paper, we treat the problem of “finding the minimal number of monitors with minimal monitoring rate” as multi-objective optimization problems of “minimizing the cost under reliability constraints” and a memetic algorithm is used to search the near-optimal solutions. In the memetic algorithm, nondominated sorting genetic algorithm II (NSGA II) [4] is introduced in the global search process and iterative random walk strategy with direction exploitation [5] is employed as local search operator. Empirical studies results showed that in multi-objective monitoring rate optimization, our method outperformed the sensitivity-based method and several evolutionary algorithms including NSGA II, PHC-NSGA-II[6] and HaD-MOEA[7], which are state-of-art approaches for multi-objective reliability allocation.

The whole paper is organized as follows: Section II summarized related work on optimal reliability allocation and multi-objective memetic algorithms. Section III analyzed reliability model of service composition under monitoring and presented the problem formulation of monitoring rate optimization. Section IV introduced our memetic algorithm to find minimal monitors with minimal monitoring rate. Section V listed experimental studies results followed by a conclusion.

II. RELATED WORK

Optimal reliability allocation has been researched for decades [1] but existing researches often deal with single-objective reliability allocation problems. Moreover, few approaches on monitoring resources allocation have been proposed. Reference [3] presented a greedy sensitivity-
based method to select most sensitive services to monitor according to monitoring mechanisms. However, it didn’t consider the trade-off between the number of monitors and the monitoring rate. Zai Wang[7] proposed a harmonic distance based multi-objective evolutionary algorithm(HaD-MOEA). It outperformed NSGA II but aimed at multi-objective testing time allocation. In a word, there is no existing work on efficient multi-objective algorithms for monitoring rate optimization, so we plan to build a memetic algorithm to solve this problem.


III. PROBLEM DESCRIPTION

In this section, we first analyze the reliability and cost models for service composition under monitoring and then present the monitoring rate optimization model.

A. Reliability and Cost Models

1) Reliability model

Monitoring process checks the status of in-use service periodically and replaces unavailable ones in background. Since service running and monitoring process are both continuous process, we first build Continuous Time Markov Chain (CTMC)[12] model to analyze the reliability of single service under monitoring, as shown by Fig. 1. In this model, state(1,0) and state(0,0) represent the normal execution and failure state of service. State(1,1) and state(0,1) represent that the service is in the monitoring state. 1/λ and 1/μ represent normal time to failure and mean time to restore the service respectively.

We assume that 1) λ(μ) of new service and the replaced service are the same because they are often selected according to certain QoS rules; 2) new changed services are always in the working state. λm represents the monitoring rate and 1/μ0 denotes the mean time taken to wait for the response of the service. 1/μ0 denotes the mean time used to wait for the response of services, select and change service. To simplify the calculation process, we consider the steady state of this model. Let πi,j denotes the steady probability of the service pool in state (i,j).

Figure 1. CTMC model for single service under monitoring.

According to the rule of “Rate of flow in=Rate of flow out for each state” [12], we can get (1) from Fig. 1.

\[
\begin{align*}
\pi_{1,0} (\lambda + \lambda_m) &= \pi_{1,0} \mu_0 + \pi_{1,1} \mu_0 + \pi_{0,0} \mu \\
\pi_{0,0} (\lambda_m + \mu) &= \pi_{0,0} \lambda \\
\pi_{0,1} \mu &= \pi_{0,0} \lambda_m, \text{ or } \pi_{1,1} \mu_0 = \pi_{1,0} \lambda_m \\
\pi_{1,1} + \pi_{1,0} + \pi_{0,1} + \pi_{0,0} &= 1
\end{align*}
\]

The reliability of this service could be perceived as the probability that the service in working states (1, i): $R_i = \pi_{1,0} \cdot \pi_{1,1}$, shown by (2). All the parameters in R could be determined except for the monitoring rate $\lambda_m$, so we could solve (2) to get $R = R(\lambda_m)$.

\[
R = (1 + \frac{\lambda_m}{\mu_0}) / (1 + \frac{\lambda_m}{\mu_0} + \frac{\lambda}{\lambda_m \cdot \mu_0} + \frac{\lambda_m}{\mu_0} + \frac{\lambda}{\mu_0} + \frac{\lambda_m}{\mu + \lambda_m + \mu})
\]

Then we analyze the reliability of service composition under monitoring. Although there are some differences between the service composition and traditional modular software, it could be transformed to DTMC [14,15]. After building the DTMC model on the composition, we could get the number of invocations to each service $V_i$. The overall reliability model of service composition with monitored services could be calculated from each service using hierarchy approach of architecture-based software reliability model[13], as shown by (3). $R_c$ denotes the reliability of the service composition, $R_i$ denotes individual services’ reliability under monitoring, m denotes the number of services in the composition and $\lambda_{mi}$ denotes monitoring rate for the ith service (1≤i≤m).

\[
R_c = \prod_{i=1}^{m} R_i^{V_i} = \prod_{i=1}^{m} R_i(\lambda_{mi})^{V_i}
\]

2) Cost model

There are generally two kinds of resources cost by monitoring. On one hand, additional resources are needed to store information about backup services, deploy the monitor and set the timer. Let $S_i$ denotes this kind of resource cost by the ith monitor and $S_c$ denotes the overall cost resources. $S_c$ could be calculated from $S_i$, as shown by (4), where $k_i$ denotes whether the ith monitor is deployed. If monitor is deployed on the ith service, $k_i=1$, else $k_i=0$. On the other hand, additional network bandwidth will be taken to transmit monitoring query and response messages. Let $L_c$ denotes the bandwidth cost in a unit time and $L_c$ denotes the length of query and response message of the ith monitor. $L_c$ is determined by the monitoring rate $\lambda_{mi}$, as shown by (5). If monitor is deployed on the ith service, $\lambda_{mi}>0$, else $\lambda_{mi}=0$.

\[
S_i = \sum_{i=1}^{m} S_i k_i, \quad k_i = \{0,1\}
\]

© 2012 ACADEMY PUBLISHER
\[ L_c = \sum_{j=1}^{n} L \lambda_m, \lambda_m \geq 0 \]  \hspace{1cm} (5) 

**B. Service Monitoring Rate Optimization Problem**

To minimize overall monitoring cost \( S_c \) and \( L_c \), we aim at choosing appropriate \( k \) and \( \lambda_m \) under reliability constraints \( R_o \). To simplify the problem, we assume that \( S_c \) and \( L_c \) for each monitor is the same and denote them by \( S_0 \) and \( L_0 \). Let \( k \) denote the number of monitors set in the composition. The optimal problem is shown in (6) and (7), depending on whether we need to maximize the reliability. Equation (6) is a bi-objective optimization problem; 3)  \( \text{set offspring population } Q_0 = \emptyset, \text{and generation counter } t = 0. \) 

Begin 

The pseudo code of MOMA’s framework is shown below. 

**IV. MULTI-OBJECTIVE MEMETIC ALGORITHM**

In this section, we proposed a multi-objective memetic algorithm (MOMA) to solve the above two problems. 

Our MOMA is built as follows: 1) the simulated binary crossover operator and polynomial mutation operator are employed as genetic operator to generate new individuals; 2) nondominated sorting strategy [4] is used to sort combined populations and to select needed solution; 3) random walk with direction exploitation method [5] is introduced to refine local best solutions. The pseudo code of MOMA’s framework is shown below. 

Begin 

Set offspring population \( Q_0 = \emptyset \), and generation counter \( t = 0. \) Initialize population \( P_0 \) with \( N \) individuals, \( P_0 = \{p_1, p_2, \ldots, p_N\} \). 

While \( t < t_{\text{max}} \), \( ( \text{the maximum generation number} ) \) 

Use genetic operator to generate \( Q_t \) from \( P_t \). 

Combine parent and offspring population \( R_t = P_t \cup Q_t \). 

Calculate objective values and constraints values of individuals in \( R_t \). 

Calculate nondominated fonts and crowding distance for elements in \( R_t \). 

Sort \( R_t \) in descending order using nondominated fonts and crowding distance and select the first individual. 

Apply local search operator to the selected one. 

Fast non dominated sort and select() 

Begin 

Fast non dominated sort \( R_t \) to get nondominated fonts \( F \) of \( R_t \), \( F = \{F_1, F_2, \ldots\} \). 

If (before local search) 

\( \text{crowding distance assignment}(F_1) \) calculate crowding distance in \( F_1 \) (the first nondominated font). 

Sort \( F_1 \) in descending order using crowding distance. 

Local best solution = \( F_1[1] \) (the first solution in the first nondominated font). 

Else (for global search) 

\( P_{t+1} = \emptyset \) and \( i = 1 \).
Begin
Local_search_procedure(Input local best individual x)

code of local search method is shown below and the process will continue until it reaches the maximal number of iterations. The pseudo random array will be reduced by half and the process will continue until it reaches the maximal number of iterations. In the search process, a random array is generated and adjusted to local its neighbors and include feasible solutions. In the search algorithm, and crowding distance calculation. For tri-objective problem, one more objective value, the reliability R=R(λ_m) is added.

C. Local Search Strategy

After the local best solution is found through nondominated sorting, the local search strategy takes use of random walk with direction exploitation method to search its neighbors and include feasible solutions. In the search process, a random array is generated and adjusted to local best solution. The new individual is then checked whether to be a nondominated solution. If so, it will be included into acceptable solutions, otherwise, the random array will be reduced by half and the process will continue until it reaches the maximal number of iterations. The pseudo code of local search method is shown below and the algorithm will return a set of nondominated solutions.

Local_search_procedure(Input local best individual x)

Begin
i=1 and Q=Ø.
While i<i_max(predefined maximal search number)
Randomly generate vector λ={λ_1,λ_2,…}. Set t=1, x_t=λ and ratio=1.
While t<t_max(predefined maximal iteration number)
x_t=x_t+ratio*λ.
If x_t constraint dominate x
Include x_t into Q and break;
Else
ratio=ratio/2.
End
Set t=t+1.
End
Set i=i+1.
End
Output Q. (Include Q into combined population R_t)
End

V. EXPERIMENTAL STUDIES

In this section, our MOMA is compared with the sensitivity-based method and other multi-objective algorithms including original NSGA II [4], PHC-NSGA-II[6] and HaD-MOEA[7]. NSGA II is a most widely used algorithm in multi-objective optimization. PHC-NSGA-II is a new memetic algorithm built upon NSGA II and HaD-MOEA is a state-of-art evolutionary algorithm for multi-objective testing time allocation. The experimental studies were designed to consist of two parts. First, the MOMA was compared with other approaches using a single group of parameters with different reliability constraint values. Second, several groups of system parameters with a certain reliability constraint value were used for comparison.

A. Experimental Design

A sample service composition, containing six types of basic workflow patterns, taken from [17] is used as the example in this section, shown by Fig. 2. The transition probability information is listed in [17] and we could obtain \( V_f = \{1,1,3,0,3,1,1,0,7,0,2,0,1,1,1,1\} \). \( R_e = R_9 R_{12} R_7 R_5 R_3 \). \( R_e R_9 R_6 R_7 R_5 R_8 R_9 R_{10} R_{12} \). 10 groups of failure rate, repair rate are simulated for 12 services in it. The data is randomly chosen from the sample data collected from real-world services through Membrane, an open source web service registry and monitoring tool [18].

![Figure 2. Sample workflow of service composition.](image)

Assuming each service waits at most 10 seconds for response [19] and 10 seconds for retrying, the value of \( \mu_0=1/t_0=3 \). Generally service selection process takes about 0.5-2 seconds according to different algorithms [20]. We don’t take the service replacement time into consideration since it is often much less than the time for service selection. As a result, the value of \( \mu_0=1/t_0 \in (30,120) \) and we choose \( \mu_0=60 \) in the following experiments. \( t_m=1/\mu_m=0.99 \). Let \( S_0 \) and \( L_0 \) to be unit 1 and omit them in the objective values calculation. In MOMA, the population size is set to 100, the maximum generation is 200, the crossover rate is set to 0.9, and the mutation rate is set to 0.1. The initial value of monitoring rate of each service \( \lambda_m (1 \leq i \leq 5m) \) is randomly generated from 0 to 1/min. The minimal monitoring interval is 1 minute and the maximal interval is \( \infty \), which implies that no monitor is set. If the monitoring rate value after genetic operations exceeds the range of [0, 1], it will be replaced by 0 or 1 respectively.

B. Comparison using Different Reliability Constraint

In this experiment, a single group of service failure rate and repair rate was randomly chosen from the dataset and the reliability threshold value was set from 0.90 to 0.99. Fig. 3 shows the comparison of a single group of solutions obtained by different algorithms for bi-objective problem under different constraint value, where \( dR_0 \) denotes the minimal reliability improvement for sensitivity-based method. Sensitivity-based method only returns a single group of solution and it is dominated by the Pareto sets obtained by MOMA, which provides more kinds of solutions. Under the same number of monitors, the monitoring rate obtained by MOMA is generally smaller than those obtained by other solutions. Fig. 4 illustrates a set of nondominated solutions obtained by multi-objective algorithms on both the bi-objective problem and the tri-objective problem. According to...
Fig. 3. Comparison of solutions obtained by different algorithms for bi-objective problem using different reliability constraint (one case study).

Fig. 4. Nondominated sets obtained by multi-objective algorithms for bi-objective and tri-objective problems using different reliability constraint (one case study).

For each reliability constraint value was set to 0.95. Fig. 5 shows the comparison of a single group of solutions obtained by different algorithms for bi-objective problem using different groups of parameters and Fig. 6 illustrates the nondominated solutions obtained on both the bi-objective and tri-objective problem. Similarly, sensitivity-based method only returns single solution for test cases, the hypervolume indicator of MOMA is higher than that other algorithms.

C. Comparion Using Different Service Reliability

In this experiment, different groups of service failure rate and repair rate from the dataset were used and reliability constraint value was set to 0.95. Fig. 5 shows the comparison of a single group of solutions obtained by different algorithms for bi-objective problem using different groups of parameters and Fig. 6 illustrates the nondominated solutions obtained on both the bi-objective and tri-objective problem. Similarly, sensitivity-based method only returns single solution for each experiment and the solution is dominated by the Pareto sets of MOMA. From Fig. 6, it is difficult to determine which algorithm dominates another, so we will compare the coverage fraction and hypervolume indicator of solutions obtained by different algorithms.

Each experiment (for different failure rate and repair rate) was carried out for 10 times and the average coverage fraction of nondominate solutions obtained by four algorithms was shown in Table I. The result obtained by MOMA that is significantly better than the other is emphasized in boldface. Again, MOMA covers greater fraction of the Pareto sets achieved by other algorithms, which implies that the solution space of MOMA is bigger than the others. Table II presents the mean values of hypervolume indicator obtained by four algorithms in 10 independent runs for each reliability constraint value. Except for few test cases, the hypervolume indicator of MOMA is higher than that other algorithms.
TABLE I.
The fraction of nondominated solutions covered by other nondominated points (the last column comprises the mean value for each row).

<table>
<thead>
<tr>
<th>Coverage (A cover B)</th>
<th>bi-objective problem</th>
<th>Reliability constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>0.90</td>
</tr>
<tr>
<td>NSGA II</td>
<td>MOMA</td>
<td>52.9%</td>
</tr>
<tr>
<td>MOMA</td>
<td>NSGA II</td>
<td>47.3%</td>
</tr>
<tr>
<td>PHC-NSGA-II</td>
<td>MOMA</td>
<td>64.7%</td>
</tr>
<tr>
<td>MOMA</td>
<td>PHC-NSGA-II</td>
<td>17.3%</td>
</tr>
<tr>
<td>HaD-MOEA</td>
<td>MOMA</td>
<td>43.3%</td>
</tr>
<tr>
<td>MOMA</td>
<td>HaD-MOEA</td>
<td>62.3%</td>
</tr>
</tbody>
</table>

TABLE II.
Values of hypervolume indicator of obtained results from different algorithms.

<table>
<thead>
<tr>
<th>Hypervolume Indicator</th>
<th>bi-objective problem</th>
<th>tri-objective problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliability constraint</td>
<td>NSGA II</td>
<td>PHC-NSGA-II</td>
</tr>
<tr>
<td>0.90</td>
<td>0.3478</td>
<td>0.1238</td>
</tr>
<tr>
<td>0.91</td>
<td>0.4879</td>
<td>0.3811</td>
</tr>
<tr>
<td>0.92</td>
<td>0.4222</td>
<td>0.2931</td>
</tr>
<tr>
<td>0.93</td>
<td>0.6669</td>
<td>0.2657</td>
</tr>
<tr>
<td>0.94</td>
<td>0.5705</td>
<td>0.3965</td>
</tr>
<tr>
<td>0.95</td>
<td>0.2664</td>
<td>0.3213</td>
</tr>
<tr>
<td>0.96</td>
<td>0.1943</td>
<td>0.2944</td>
</tr>
<tr>
<td>0.97</td>
<td>0.2353</td>
<td>0.1840</td>
</tr>
<tr>
<td>0.98</td>
<td>0.0914</td>
<td>0.0281</td>
</tr>
<tr>
<td>0.99</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Mean</td>
<td>0.3283</td>
<td>0.2288</td>
</tr>
</tbody>
</table>

Figure 5. Comparison of solutions obtained by different algorithms for bi-objective problem using different groups of parameters (one case study).

10 independent runs for each group of parameter. For most test cases, the hypervolume indicator of MOMA outperforms other algorithms.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose to optimally deploy monitors and set monitoring rate in service composition to improve the reliability as well as to save unnecessary cost. To improve the optimization efficiency, we formulate this problem as two multi-objective optimization problems...
Figure 6. Nondominated sets obtained by multi-objective algorithms for bi-objective and tri-objective problem using different groups of parameters (one case study).

TABLE III.
The fraction of nondominated solutions covered by other nondominated points (The last column comprises the mean value for each row).

<table>
<thead>
<tr>
<th>Coverage (A cover B)</th>
<th>Test case no.</th>
<th>bi-objective problem</th>
<th>tri-objective problem</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>NSGA II</td>
<td>MOMA</td>
<td>6.3%</td>
<td>17.8%</td>
</tr>
<tr>
<td>MOMA</td>
<td>NSGA II</td>
<td>91.5%</td>
<td>77.5%</td>
</tr>
<tr>
<td>PHC-NSGA-II</td>
<td>MOMA</td>
<td>2.9%</td>
<td>21.2%</td>
</tr>
<tr>
<td>MOMA</td>
<td>PHC-NSGA-II</td>
<td>97.1%</td>
<td>72.5%</td>
</tr>
<tr>
<td>HaD-MOEA</td>
<td>MOMA</td>
<td>11.0%</td>
<td>16.6%</td>
</tr>
<tr>
<td>MOMA</td>
<td>HaD-MOEA</td>
<td>66.9%</td>
<td>59.4%</td>
</tr>
</tbody>
</table>

TABLE IV.
Values of hypervolume indicator of obtained results from different algorithms.

<table>
<thead>
<tr>
<th>Test case no.</th>
<th>Hyipervolume Indicator</th>
<th>bi-objective problem</th>
<th>tri-objective problem</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NSGA II</td>
<td>PHC-NSGA-II</td>
</tr>
<tr>
<td>1</td>
<td>0.4290</td>
<td>0.3510</td>
<td>0.3160</td>
</tr>
<tr>
<td>2</td>
<td>0.2679</td>
<td>0.2257</td>
<td>0.2609</td>
</tr>
<tr>
<td>3</td>
<td>0.3249</td>
<td>0.1797</td>
<td>0.3878</td>
</tr>
<tr>
<td>4</td>
<td>0.5528</td>
<td>0.4499</td>
<td>0.5531</td>
</tr>
<tr>
<td>5</td>
<td>0.3873</td>
<td>0.3310</td>
<td>0.5550</td>
</tr>
<tr>
<td>6</td>
<td>0.2886</td>
<td>0.2550</td>
<td>0.4626</td>
</tr>
<tr>
<td>7</td>
<td>0.3781</td>
<td>0.2520</td>
<td>0.2613</td>
</tr>
<tr>
<td>8</td>
<td>0.4195</td>
<td>0.3815</td>
<td>0.3454</td>
</tr>
<tr>
<td>9</td>
<td>0.5934</td>
<td>0.3460</td>
<td>0.4272</td>
</tr>
<tr>
<td>10</td>
<td>0.2210</td>
<td>0.2555</td>
<td>0.1978</td>
</tr>
<tr>
<td>Mean</td>
<td>0.3863</td>
<td>0.3027</td>
<td>0.3568</td>
</tr>
</tbody>
</table>

and use memetic algorithm to search solutions. The memetic algorithm employs nondominate sorting strategy from NSGA II as the global search method and applies random walk with direct exploitation method as local search strategy. The experimental studies results showed that our MOMA outperformed both sensitivity-based method and other multi-objective evolutionary algorithms including NSGA II, PHC-NSGA-II and HaD-MOEA.

This paper presented a further step of work on monitoring resource allocation comparing with our previous work [3]. But there is still a lot of work to do in the future. For example, we take use of Markov chain
this paper to get the reliability model of service composition under monitoring but we only consider the steady state of Markov chain. In Internet environment, states of services may change too quickly to achieve the steady state. So, in the next step, we plan to consider real time processing of monitoring resources optimization and consider the usage of dynamic resources allocation.

ACKNOWLEDGMENT

The work reported in this paper has been supported by the Fundamental Research Funds for the Central Universities Project, under contact No.CDIXS11181161, the Fundamental Research Funds for Central Universities Project, under contact No.CDJRC10090002 and Major Program of the National Natural Science Foundation of China, under contact No.90818028.

REFERENCES


Pan He received the B.S. and M.S degree in software engineering from the School of Software Engineering, Chongqing University, Chongqing ,China, in 2006 and 2008; and is currently working toward the Ph.D degree at the College of Computer Science, Chongqing University, Chongqing, China.

Junhao Wen received the B.S. and M.S degree in computer science and application from the College of Computer Science, Chongqing University, Chongqing, China, in 1991 and 1999; and the Ph.D. degree from the College of Computer Science, Chongqing University, Chongqing, China, in 2008. Since 1991, he has been working in the College of Computer Science, Chongqing University, Chongqing, China. Since 2008, he has been a professor in the College of Computer Science, Chongqing University. From 2009, he has been working as a doctor supervisor. He is the co-author of more than 30 referred publications. His major research interests include service-oriented computing, data mining and real-world applications.

Dr. Wen is the chairman of branch forum of CCFYOCSF (CCF Young Computer Scientists & Engineers Forum) in Chongqing.