Multi-Index Cooperative Mixed Strategy for Service Selection Problem in Service-Oriented Architecture

Yimin Shen
Department of Computer Engineering, Chengdu Electromechanical College, Chengdu, China
Email: shen_yimin@sina.com

Jing Zhang
Department of Computer Engineering, Chengdu Electromechanical College, Chengdu, China
Email: cdzjcd@hotmail.com

Yushun Fan
Automation Department, Tsinghua University, Beijing, China
Email: fanyus@tsinghua.edu.cn

***Abstract*—Under the environment of Service-Oriented Architecture (SOA), service users have to decide how to choose the proper provider from the candidates to obtain optimal service level. The existing approaches based on the optimization theories usually ignore the decision conflicts among users, which result in periodic QoS fluctuation and poor system performance. This paper treated the service selection problem as an inconstant-sum non-cooperative n-person dynamic game with incomplete information. The mechanism that generated the decision conflicts is discussed at first, and then Multi-index Cooperative Mixed Strategy (MCMS) with corresponding algorithm for finding its optimal solution is developed. Simulation shows that MCMS offers service users an optimal strategy that reduces the fluctuation of QoS and improves performance of SOA. Moreover, the stability and equilibrium of MCMS make it acceptable to most of the users, thus MCMS is able to improve predictability and controllability of SOA significantly.**

***Index Terms*—service oriented architecture (SOA), multi-index, service selection, mixed strategy**

I. Introduction

Service Oriented Architecture (SOA) is changing the traditional software engineering by the way of loose coupling, implementation neutrality, flexible configuration, etc [1]. It is regarded as the main software engineering practice in future to fit in with the needs of large-scale, high-complexity software [2].

Unfortunately, the optimism of the academic community is a little far from the reality. For example, it was reported by Universal Business Registry (UBR) that there were less than 1,200 registry services during 2003~2004, and only 34% of them were available [3]. One of the important reasons of this phenomenon was that some Quality of Service (QoS) indexes, like response time, were not optimized or guaranteed. Therefore the users had to suffer the unbearable waiting time.

Since in the three main operations of SOA, service-finding is the exclusive one that coordinates all of the three roles (user, provider and broker) in SOA [4], the researches on the service level improvement concentrate primarily in the service-finding process. In this case, the service selection problem (SSP) existing in service-finding has caused extensive concern [5]-[13].

SSP is defined as the problem that service requester has to make the optimal decision of selecting the service provider from multiple candidates discovered by broker. The providers serve in similar way and offer substitutable functions to the users, but with different QoS [5].

Many strategies have been developed to help the users to gain optimal performance by selecting correct provider. The most studied methods are dynamic programming [5][6], Genetic Algorithm [7][8], interactive method [9], fuzzy logic [7][10], and other performance evaluation approaches [11]-[13], etc.

However, all the above service selection approaches did not concern about the possible conflicts among the users' strategies. Since there is no administrator to determine or implement a global-optimized strategy in the peer to peer (P2P) world of SOA, the contention for "best" service providers forms a "winner takes all" situation, and thus makes the quality of services instability.

Such instability of service quality can be observed in P2P environment frequently. Fig. 1 shows the case that data is transferred peer to peer between 27 seeds (service providers) and clients (users and providers) by Bit Torrent Plus! 2.130.2.110 [14]. Here upload rate is tracked as a key QoS index, and it presents a large-amplitude periodic vibration between 50kbps and 220kbps.

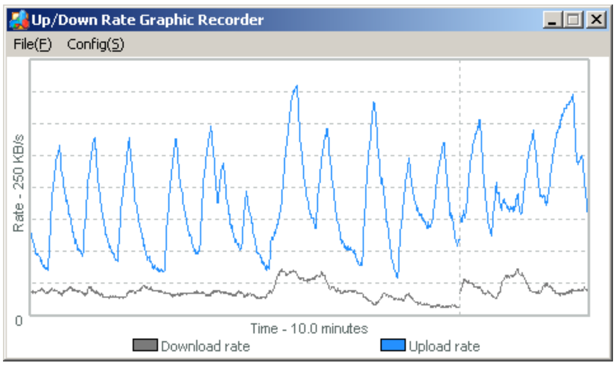

Figure 1. Fluctuation of transmission rate in P2P environment.

Shen and Fan developed the Cooperative Mixed Strategy (CMS) to eliminate the fluctuation of queue length, and shorten the time for users to wait for service [15]. This approach is efficient for single-index cases in which only time is to be optimized. However, in more complicated cases, e.g. the users prefer a combination object function of time, cost, and quality, etc. the mathematical model of CMS is no longer applicable.

In Section II of this paper, the mechanism of QoS fluctuation in SOA is studied. And then the Multi-Index Cooperative Mixed Strategy (MCMS) is presented in Section III for the cases with multiple QoS indexes. In Section IV, the approach to obtain optimal solution is discussed in detail. And the performance, stability, and equilibrium of MCMS are simulated and evaluated in Section V. Finally, Section VI concludes the paper.

II. Mechanism of QoS Fluctuation

Suppose that m parallel providers $M = \{M_i\}_{i=1}^{m}$ are able to provide same service in SOA in the similar way but with different QoS. The arrival process of the service requests follows the Poison process with a mean arrival rate λ. When one of the requests, r, arrives at time t, it requires QoS parameters of all the m providers from the QoS broker, and then select a provider from the m candidates. After making such decision, it takes T_p time for r to complete the preparatory tasks like binding service and transmitting data. So r starts queuing at the moment $t+T_p$. The process is demonstrated in Fig. 2.

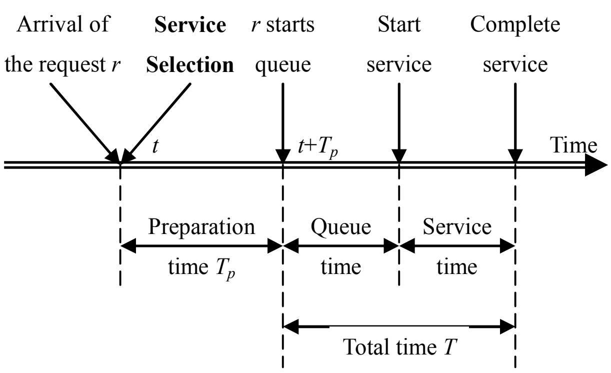

Figure 2. Procedure of service selection and binding in SOA

Let us focus on the multiple indexes in the decision-making process. Because the total time that the user stays in the queue relies on the strategies of all the related users, the n QoS indexes are divided into two sets. Total time T belongs to the Decision Dependent Index (DDI) set, while the other n–1 indexes belong to the Decision Independent Index (DII) set. Here "decision independent" means that the index is the stationary process of time t. Indexes like quality level and list price are typical DIIs. On the other hand, "decision dependent" means the index depends on both the time t and other users' selection strategies. In order to simplify the mathematical model, total time T is the unique DDI considered in this paper.

Accordingly, QoS of M_i at the time t is an n-dimension random process $Q_i^t = (q_{i1}^t, q_{i2}^t, \ldots, q_{i,n-1}^t, T_i^t)$. Here waiting time T_i^t is the DDI value, composing of the queuing time and service time, whereas q_{ij}^t (j=1,2,…n–1) is the DII value. The service time of a request on M_i is a negative exponential distributed variable with mean μ_i.

Suppose that the object of optimizing the service is a linear function f, i.e., to find a M_i to maximize $f = Q_i^{t+T_p}\alpha$. Here α is an n-dimension constant weight vector. Please refer to the literatures like [16] for how to select the appropriate α value when solving practical problems. As r is unable to predict Q^{t+Tp} at the time t, therefore, the only way is to choose a service provider M_i which maximizes $f = Q_i^t\alpha$ based on the known Q_i^t. This is the widely-applied optimization-based pure strategy.

However, M_i is the optimal solution, according to f, so the majority of requests arriving at the time interval (t–Δt, t) tend to choose M_i, like r does. This consistency of decision just forms a competition for the optimal resource M_i, thus results in a serious decision conflict. Because the requests arriving at (t–Δt, t) are more possible to complete the preparatory work and start queuing earlier than r which arrives at time t. So according to the principle of first-come-first-serve (FCFS), they will be served ahead of r. In this situation, when the request r starts queuing up after time $t+T_p$, it will find that the length of the queue and the waiting time is longer than expected, so the optimization object $Q_i^{t+T_p}\alpha$ is worse than expected value $Q_i^t\alpha$.

And then, the coming users will notice the services congestion on M_i, and then the subsequent service requests will turn to chase other service providers instead of M_i. The queue length and waiting time on M_i starts to decrease, and QoS resumes until the next congestion. But during this period, the congestion appears gradually on other service providers. The external manifestation of this kind of periodic phenomenon is the fluctuation of response time or transmission speed, as observed in Fig. 1.

In order to simulate the mechanism of the periodic phenomenon, a SOA environment with 10 service providers are established. (For the configurations detail of the simulation, please refer to Section V of the paper). 10,000 service requests are generated in accordance with the Poisson process, and are assigned to the providers by pure strategy. The delay time T_p is set to be 0.2 minute only, and then an "orderly" phenomenon, i.e. the periodic fluctuation of the queue length, appears. Fig. 3 shows the time series of the queue length of provider M_3, as an example. Its periodic vibration in the experiment is very similar with that in the real world demonstrated in Fig. 1.

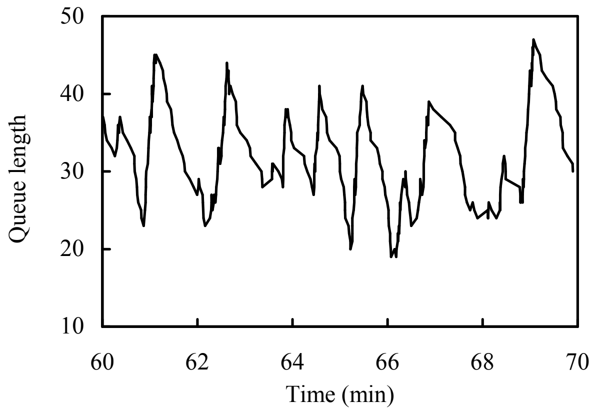

Figure 3. QoS Fluctuation simulated in P2P environment

Moreover, the delay time T_p is expected to be prolonged because SOA recommends one-time transmission process using text data to establish service connection, instead of traditional remote procedure call (RPC). This increases the complexity and difficulty of the optimization problem related to DDI. If users still apply any pure strategies, then the incomplete information during this period will aggravate the performance of service providers.

It can be concluded from the above analysis that the following three factors cause the service level declination and QoS fluctuation:

1) *P2P environment*. Among the three parties in SOA, the broker seems unequal to the others. But in fact, its main function is to provide services like registration management, service finding or matching, etc. Each service requestor has the right to make its own decision, independent of the broker or QoS broker.

2) *Incomplete information*. When a user arrives, it does not know the decisions that the other users have made already. So the queue lengths after T_p are not predictable for it. This kind of imperfect information can easily result in decision conflict.

3) *Pure strategy*. Since all the users adapt optimization methodology to obtain optimal result, they can not work together to reach the equilibrium of the n-person game.

Because the first three factors are unavoidable, this paper takes the mixed strategy instead of pure strategy to solve this problem.

III. Multi-Index Cooperative Mixed Strategy

In order to avoid the conflicts described above, service users have to consider the possible strategies of other users. Therefore, this problem becomes an inconstant-sum, non-cooperative, n-person dynamic game with incomplete information. In traditional situations, the conflict can be resolved by job scheduling approaches. In the P2P world of SOA, however, the users are not expected to avoid the decision conflict by sacrificing their own interests. In this case, the main idea and logic of this paper are:

- A Multi-index Cooperative Mixed Strategy (MCMS) is developed to replace the traditional pure strategy;
- If all users adapt MCMS, the object function f (i.e., payoff function) will be optimized significantly with weakened QoS fluctuation and improved service level;
- In case that some users (the obedient) adopt MCMS, but other users (the disobedient) still apply pure strategy, the payoff of the obedient is still than before, and higher than that of the disobedient.

The above mechanism makes MCMS coincidence with the users' principle of maximizing their own interests. So the rational users will make their decision using MCMS, which forms a stable and harmonious situation of the dynamic game. The procedure of service matching and selection using MCMS is shown in Fig. 4.

MCMS is outlined as follows. When a service request r arrives at time t, it predicts the QoS value of time $t+T_p$, and calculates the expected payoff Ef for each components of its pure strategy set. And then r does not choose the M_i that minimizes the object function, but chooses from $M=(M_1,M_2,...,M_m)$ using the roulette wheel approach with the probability $\tilde{p}=(\tilde{p}_1,\tilde{p}_2,...,\tilde{p}_m)$. Here $\sum_{i=1}^{m}\tilde{p}_i=1$, and $\tilde{p}$ is elaborated to maximize the mean payoff Ef.

MCMS can be divided into the following steps.

1) The service providers register to the service broker;

2) The QoS broker accesses the providers to get their QoS information updated in an event-driven or a period manner;

3) When a request emerges to the architecture, it fetches updated QoS information from the QoS broker, particularly the parameters $\{\mu_i\}$ and the current queue length $\{L_i\}$;

4) Then the user calculates the probability $\tilde{p}_i$ corresponding with each provider M_i, with the method described in the next section;

5) Finally the requester uses a biased roulette wheel approach based on the above distribution to determine which provider that it shall choose.

The main characteristic of MCMS is that it becomes a random process with certain optimal distribution other than a certain process. The service level that the service users obtain is determined by this distribution in essence. Therefore, the problem of selecting optimal service providers is generalized to find out the proper distribution. The algorithm of optimal distribution is presented in the next section.

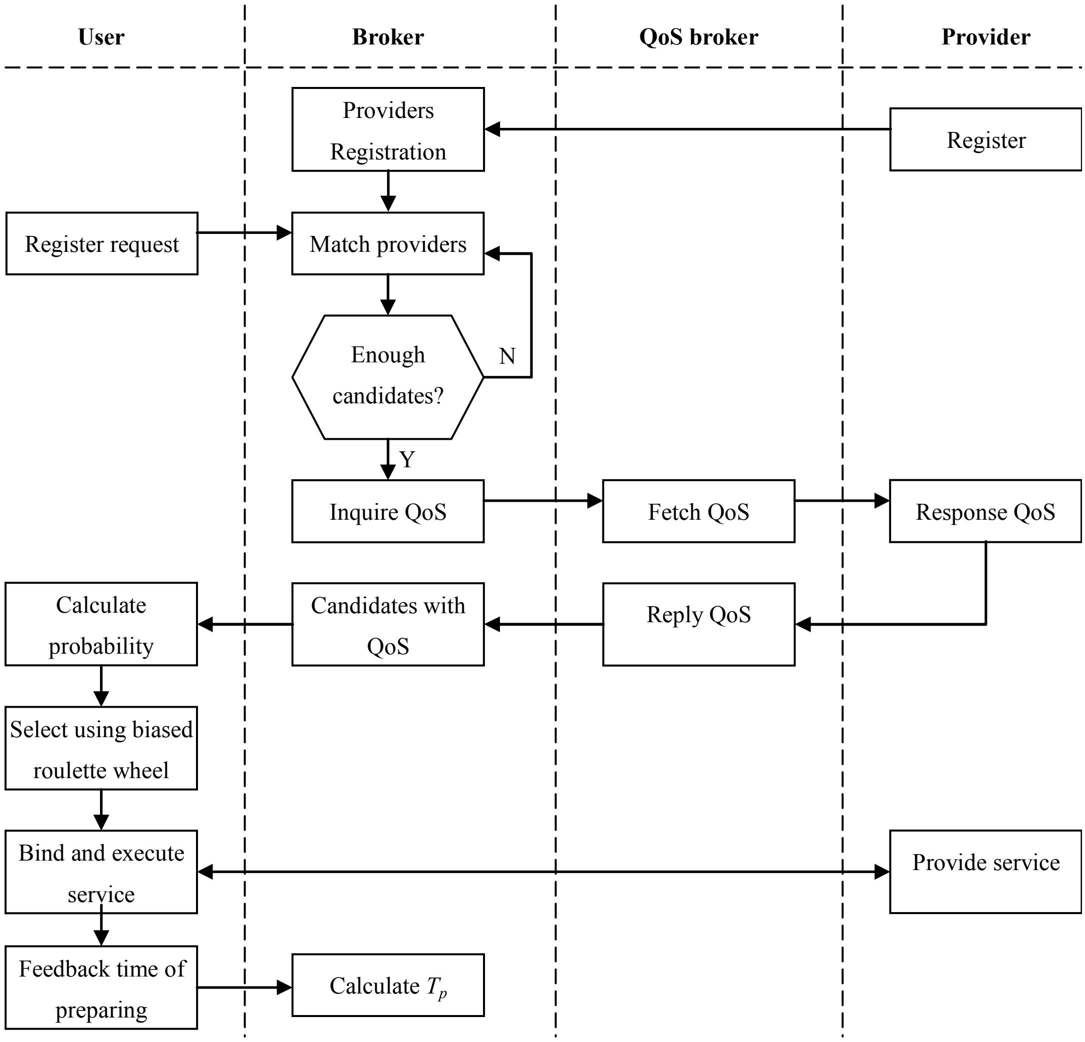

Figure 4. Procedure of mixed strategy for service selection

IV. Global Optimized Solution

According to the definition of DII, q_{ij}^t is an unbiased estimation of $Eq_{ij}^{t+T_p}$ because of its stationary. So the challenge for r to maximize Ef is defined as the nonlinear programming problem (1).

$$\max Ef = \sum_{i=1}^{m} f(\{q_{ij}^t\}_{j=1}^{n-1}, ET_i^{t+T_p})\tilde{p}_i$$
$$st.\begin{cases} \sum_{i=1}^{m} \tilde{p}_i = 1 \\ \tilde{p}_i \ge 0 \qquad i = 1,2,\dots,m \end{cases} \tag{1}$$

Generally speaking, (1) is irresolvable for r because $ET_i^{t+T_p}$ is incomputable when lack of other users' strategies. Nevertheless, if all the users hold a kind of cooperative attitude, i.e., applying MCMS, they can not only forecast the mean $ET_i^{t+T_p}$, but also optimize their expected payoff (1).

Moreover, this kind of cooperative attitude is not an extravagant hope because, as shown in Section V of the paper, users holding uncooperative attitude will make their own payoff reduced. This special feature of MCMS forces the users to follow the rules proposed by MCMS.

Let p_i be the probability that M_i is chosen at the neighborhood of the moment t. It is clear that on the one hand, $\tilde{p}$ is a sample of $p=(p_1,p_2,\dots,p_m)$ thus they are related so closely that the distribution of p is determined by the many $\tilde{p}$ s. But on the other hand, $\tilde{p}$ and p usually have different mathematical characters because $\tilde{p}$ does not have to apply the same strategy as p.

Let $Q = \begin{pmatrix} q_{1,1}^t & q_{1,2}^t & \cdots & q_{1,n-1}^t & ET_1^{t+T_p} \\ q_{2,1}^t & q_{2,2}^t & \cdots & q_{2,n-1}^t & ET_2^{t+T_p} \\ \cdots & \cdots & \cdots & \cdots & \\ q_{m,1}^t & q_{m,2}^t & & q_{m,n-1}^t & ET_m^{t+T_p} \end{pmatrix}$, then Ef can be written in a matrix expression: $Ef = \tilde{p}^T Q\alpha$. Here $\tilde{p}^T$ is the transposition of $\tilde{p}$.

Assuming the users waiting for M_i at the moment t is L_i^t, it can be deduced from Little formula [17] that,

$$ET_i^{t+T_p} = \frac{1+L_i^{t+T_p}}{\mu_i} = \frac{1+\max(L_i^t + p_i\lambda - \mu_i)T_p, 0)}{\mu_i}$$

Under the cooperative situation, the parties in the game follows the same distribution, i.e. $\tilde{p}=p$. In this case, the problem (1) turns to be the nonlinear programming problem (2).

$$\min -Ef = -p^T Q\alpha = -\sum_{i=1}^{m}\left[\left(\sum_{j=1}^{n-1}\alpha_j q_{ij}^t + \frac{\alpha_n}{\mu_i}\right)p_i + \alpha_n \max\left(\frac{\lambda T}{\mu_i}p_i^2 + \frac{L_i - \mu_i T}{\mu_i}p_i, 0\right)\right] \quad (2)$$

$$st.\begin{cases}\sum_{i=1}^{m} p_i = 1 \\ p_i \ge 0 \qquad i = 1,2,...,m\end{cases}$$

It can be verified that (2) is a convex programming problem with m variables and m+1 linear constraints, so (2) has good computability. Moreover, because of the continuity of Ef on [0,1], the left-hand and right-hand partial derivatives exist:

$$\frac{\partial Ef}{\partial p_i^-} = \begin{cases} c_i & 0 < p_i \le -2b_i \\ 2a_i p_i + 2a_i b_i + c_i & else \end{cases}$$

$$\frac{\partial Ef}{\partial p_i^+} = \begin{cases} c_i & 0 \le p_i < -2b_i \\ 2a_i p_i + 2a_i b_i + c_i & else \end{cases}$$

Here $a_i = \frac{\alpha_n \lambda T}{\mu_i}$, $b_i = \frac{L_i - \mu_i T}{2\lambda T}$, and $c_i = \sum_{j=1}^{n-1}\alpha_j q_{ij}^t + \frac{\alpha_n}{\mu_i}$.

The following corollary is deduced from the left (right) continuity of the above left (right)-hand partial derivative.

Corollary. The pure strategy M_i is the optimal strategy if and only if $\forall i' \ne i \Rightarrow \left.\frac{\partial Ef}{\partial p_i^-}\right|_{p_i=1} \ge \left.\frac{\partial Ef}{\partial p_{i'}^+}\right|_{p_{i'}=0}$.

Proof. (Necessity, counterevidence) Without lose of generality, assuming M_1 is the optimal strategy, i.e. i =1. Thus the pure strategy is a special mixed strategy p=(1,0,0,…0). Without lose of generality also, assume $\left.\frac{\partial Ef}{\partial p_1^-}\right|_{p_1=1} < \left.\frac{\partial Ef}{\partial p_2^+}\right|_{p_2=0}$ for i'=2. If the strategy p changes to p'=(1–ε,ε,0,…,0), then according to the existence and left (right) continuity of left (right)-hand partial derivative of Ef, the difference of $-Ef$ is,

$$\Delta(-Ef) = -\int_0^{\varepsilon}\frac{\partial Ef}{\partial p_2^+} + \int_{1-\varepsilon}^{1}\frac{\partial Ef}{\partial p_1^-} + o(\varepsilon) = \varepsilon\left(\left.\frac{\partial Ef}{\partial p_1^-}\right|_{p_1=1} - \left.\frac{\partial Ef}{\partial p_2^+}\right|_{p_2=0}\right) + o(\varepsilon)$$

Because $\left.\frac{\partial Ef}{\partial p_1^-}\right|_{p_1=1} < \left.\frac{\partial Ef}{\partial p_2^+}\right|_{p_2=0}$, if ε is a sufficient small number, then $\Delta(-Ef)<0$. This shows that the mixed strategy p' is better than the pure strategy p, and brings a contradiction.

(Sufficiency) When $\forall i' \ne 1 \Rightarrow \left.\frac{\partial Ef}{\partial p_1^-}\right|_{p_1=1} \ge \left.\frac{\partial Ef}{\partial p_{i'}^+}\right|_{p_{i'}=0}$, any small change Δp=(1–ε, ε_2, ε_3,…,ε_m) ($\varepsilon = \sum_{i=2}^{m}\varepsilon_i$) in the neighborhood around the pure strategy p=(1,0,0,…0) will bring changes to $-Ef$ as,

$$\Delta(-Ef) = \int_{1-\varepsilon}^{1}\frac{\partial Ef}{\partial p_1^-} - \sum_{i=2}^{m}\int_0^{\varepsilon_i}\frac{\partial Ef}{\partial p_i^+} + \sum_{i=2}^{m}o(\varepsilon_i) = \sum_{i=2}^{m}\varepsilon_i\left(\frac{\partial Ef}{\partial p_1^-} - \frac{\partial Ef}{\partial p_i^+}\right) + \sum_{i=2}^{m}o(\varepsilon_i)$$

If ε_i is sufficient small, then $\Delta(-Ef)\ge0$, i.e. any mixed strategy is not superior to the pure strategy. So the pure strategy is a local maximum of the problem. Since local maximum of any convex programming problem is also a global maximum, the pure strategy is the optimal solution of the game.

Therefore the corollary holds. ∎

As $\left.\frac{\partial Ef}{\partial p_i^-}\right|_{p_i=1} \ge \left.\frac{\partial Ef}{\partial p_{i'}^+}\right|_{p_{i'}=0}$ equals to $\min_{p_i\in(0,1)}\frac{\partial Ef}{\partial p_i^-} \ge \max_{p_{i'}\in(0,1)}\frac{\partial Ef}{\partial p_{i'}^+}$ since the partial derivatives are monotonic decrease functions. Hence according to the corollary, pure strategy can not be optimal unless a provider is absolute superior to others. This ensures the predominance of mixed strategy in most cases.

V. SIMULATIONS

A. Service Performance Improvement

In order to verify the efficiency of MCMS in actual environment, a series of simulation experiments are undertaken. All the service users, service broker, QoS broker, and service providers are simulated on the computer with Intel Pentium IV 1.6-GHz using VBA language. The parameters of the experiment are selected as follows. Service requests emerge according to Poisson process with intensity λ=100. 10 providers of the service register to both the broker and QoS broker. The time that the providers complete a request follows negative exponential distribution with randomly selected parameters μ_i = 19, 12, 9, 7, 2, 16, 8, 14, 8 and 5 min respectively. The time gap T_p is set to 0.2 min.

Since DII does not change along time, multiple DIIs can be merged statically by the method of weighted mean. So only one DII needs to be considered in the experiment, which is called synthetic cost, or cost for short. The cost to offer a service to a user varies with different providers, and is set randomly to 0.35, 0.6, 0.5, 0.45, 0.3, 0.6, 0.7, 0.5, 0.3, and 0.65 respectively. The weight α in the object function is set to (–0.4, –0.6).

A comparison experiment is conducted by Selection Method Using PROMETHEE (SMUP) proposed in [13], with same parameters as configured in MCMS.

During the experiment, 10,000 requests are assigned to the providers using MCMS approach, and their object function values are recorded. The mean f values that the 1,000th–9,000th requests gain do not have any clear trend, as shown in Fig. 5, so the 8,000 records are taken as the samples for analysis.

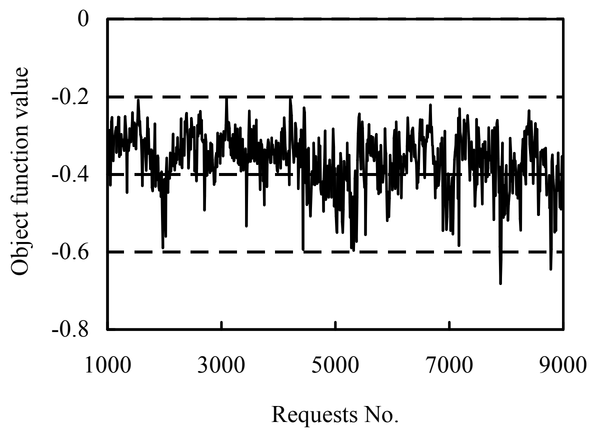

Figure 5. Mean value of the object function in the experiment.

The total time and queue length of the 8,000 samples above are tracked. The performance of MCMS and SMUP is compared in Table 1.

TABLE I.
COMPARISON OF MCMS AND SMUP

Item	SMUP	MCMS	MCMS/ SMUP
average object function value	–0.79602	–0.36498	45.85%
maximum object function value	–1.93109	–1.75969	91.12%
average total time (min)	0.99511	0.28162	28.30%
maximum total time (min)	2.91849	2.73282	93.64%
average queue length	13.5395	2.60013	19.20%
maximum queue length	51	16	31.37%

It is obvious from Table 1 that MCMS outperforms traditional pure strategies like SMUP since MCMS averagely improves the object function more than 50%, shortens 70% of the average total time and queue length. Therefore service users can obtain much better service level when the capability of providers is not changed at all.

Besides, the queue length sequence of a typical service provider, M_3, during the time interval [50, 65] is shown in Fig. 6 to compare the performance of MCMS and SMUP.

Fig. 6 indicates that cooperative mixed strategy MCMS can not only reduce the peaks of queue length from 10–20 to 2–6, but also weaken the periodic fluctuation significantly.

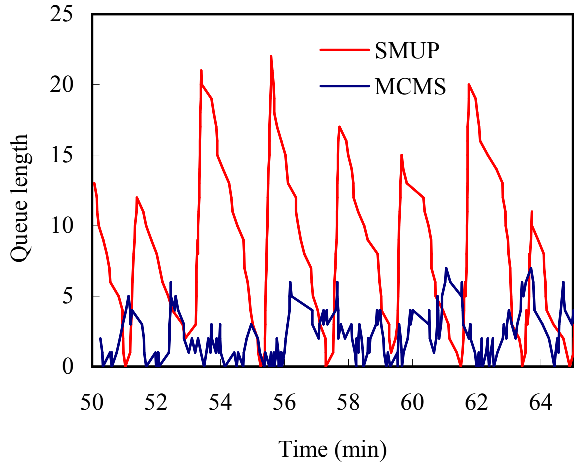

Figure 6. Queue length sequence of M_3 using MCMS and SMUP.

B. Stability of Stochastic Process

During the experiment described above, we can easily track the random counting process $N_i(t)$ that represents the total number of requests assigned to the provider M_i in a time interval between 0 and t. $N_3(t)$ is shown in Fig. 7 as an example.

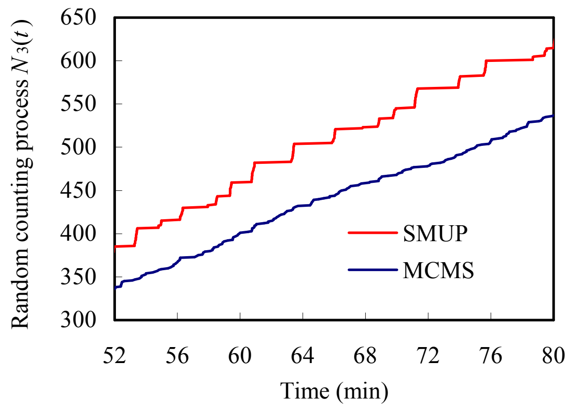

Figure 7. Random counting process $N_3(t)$ under SMUP and MCMS.

It can be observed from Fig. 7 that for any τ, the increment $N_3(t+\tau)–N_3(t)$ is time-independent and proportional to τ under MCMS, whereas the increment is time-dependent and is not determined by τ under SMUP. Hence $N_i(t)$ is much more likely to be an independent increment process under MCMS than under SMUP

Moreover, since $N_i(0)=0$, $N_i(t)$ can be treated as a stochastic Poisson increment process that is *covariance stationary*. On the contrary, output of pure strategy like SMUP is hard to be forecasted or controlled due to its lack of stability.

C. Non-cooperative Case and Equilibrium

The global optimized solution calculated in Section IV maybe not the equilibrium of the n-person non-cooperative game, according to game theory. In fact, the optimal solution gained from (2) maybe not in line with the parties' own best interests. In this case, it is necessary to discuss the non-cooperative equilibrium of the dynamic game.

Assuming that q proportion of service users are disobedient, they hold uncooperative attitude and adopt pure strategy. After obtaining other user's strategies by (2), the disobedient users select the "best" provider with probability 1. Obviously, when q=0, the non-cooperation case degrades to cooperation case.

We execute a series of experiments to simulate the system performance when q changes. The parameters of the experiments are configured same as that in the previous section. Fig. 8 shows average total time recorded in experiments corresponding with various q. The data from Table 1 which illustrates the average performance of pure strategy SMUP are also drawn as a constant function (the horizontal line located around –0.8) in Fig. 8 for comparison.

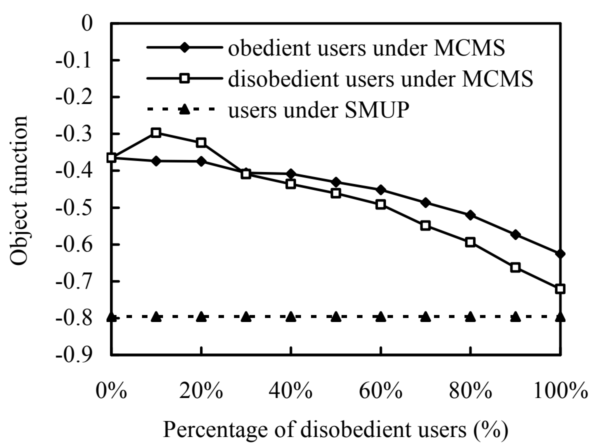

Figure 8. Decreasing trend of the object function value when the proportion of the disobedient, q, increases.

It can be seen from Fig. 8 that

1) the average object function value under mixed strategy MCMS is always much better than that under pure strategy;

2) because of the stationary of MCMS, when $q \leq 30\%$, their uncooperative attitude does not result in any big descending in average performance of the system; and

3) when $40\% \leq q \leq 100\%$, the average object function has a decreasing trend when the number of the disobedient increases. But during this process, the payoff function of the users adopting uncooperative strategy is, contrary to what might be expected, worse than the others. The phenomenon is caused by strategy conflict among the disobedient.

Therefore, existence of a few disobedient users around the optimal solution of MCMS does not influence the SOA performance significantly. When the disobedient increases, their uncooperative attitude does not give them extra payoff, but reduces it. This statistical result encourages rational users to adapt MCMS instead of any pure strategy to obtain optimal service. This kind of negative feedback forces the system back to the optimal solution of MCMS, so the MCMS-based situation becomes the equilibrium point of the game.

VI. Conclusions

The P2P architecture of SOA induced the regression of the software and services from disorder to order. Under the disorder state, decision conflict leads to a new orderly state, i.e. the burst jam and periodic fluctuation. Any deterministic approach is unable to resolve the problem in the P2P world.

The first contribution of this article is giving consideration to both the DDI and DII indexes of QoS, and develops the stochastic process model of services selection under the multi-index environment. This model reproduces the fluctuations of QoS indexes, and creates analysis condition to the service performance forecast and service levels optimization.

On this basis, theory of game, optimization, and queuing are integrated to form a cooperation mixed strategy MCMS which is effective in solving services selection problem. The simulation results show that this strategy provides optimal solutions with stability and equilibrium.

Most of the current literatures applied deterministic approaches. Even fuzzy logic applied in the service selection produces same solution to all the users. The second contribution of the paper is the establishment of a random approach for the problem. By taking full advantage of the individual uncertainty under the control of the general distribution, MCMS is able to balance the load of the providers, weaken the QoS fluctuation, and therefore improve the service level of SOA significantly.

This paper focuses on the service selection problem of a single kind of service in SOA. It needs further study on how to adapt the mixed strategy to composition situation where user takes a selection of various services composed together to form a complex business process.

Acknowledgment

The authors wish to thank Xiaofeng Di, Keyao Shen for their comments which led to improvements of this paper. This work was supported in part by the National Hi Technology Research and Development Program of China under Grant 2006AA04Z151 and 2006AA04Z166, and by the National Natural Science Foundation of China under Grant 60674080.

References

[1] M. N. Huhns and M. P. Singh, "Service-oriented computing: key concepts and principles," *IEEE Internet Computing*, vol. 9(1), pp. 75–81, 2005.

[2] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 279(5), pp. 35–43, 2001.

[3] S. M. Kim, M. C. Rosu, "A survey of public Web services," in *Proc. of the WWW 2004*, S. I. Feldman, M. Uretsky, M. Najork, and C. E. Wills, Eds. New York: ACM Press, 2004, pp. 312–313.

[4] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, et al., *Web Services Architecture*, W3C Working, 2004, http://www.w3.org/TR/ws-arch/
[5] F. Li, F. C. Yang, and S. Su, "On distributed service selection for QoS driven service composition," in *Proc. of the EC-Web 2006*, LNCS 4082, K. Bauknecht, B. Pröll, and H. Werthner, Eds. Heidelberg: Springer-Verlag, 2006, pp. 173–182.
[6] T. Yu and K. J. Lin, "Service selection algorithms for composing complex services with multiple QoS constraints," in *Proc. of the ICSOC 2005*, LNCS 3826, B. Benatallah, F. Casati, P. Traverso, Eds. Heidelberg: Springer-Verlag, 2005, pp. 130–143.
[7] H. C. Wang, C.S. Lee, and T.H. Ho, "Combining subjective and objective QoS factors for personalized web service selection," *Expert Systems with Applications*, vol. 32, pp. 571–584, 2007.
[8] S. L. Liu, Y. X. Liu, N. Jing, G. F. Tang, and Y. A. Tang, "Dynamic Web service selection strategy with QoS global optimization based on multi-objective genetic algorithm," in *Proc. of the GCC 2005*. LNCS 3795, H. Zhuge and G. C. Fox, Eds. Heidelberg: Springer-Verlag, 2005, pp. 84–89.
[9] M. Comuzzi and B. Pernici, "Negotiation support for Web service selection," in *Proc. of the TES 2004*. LNCS 3324, M. C. Shan, U. Dayal, and M. Hsu, Eds. Heidelberg: Springer-Verlag, 2004, pp. 29–38.
[10] Y. Zhang, S. S. Zhang, and S. Q. Han, "A new methodology of QoS evaluation and service selection for ubiquitous computing," in *Proc. of the WASA 2006*, LNCS 4138, X.Z. Cheng, W. Li, and T. Znati, Eds. Heidelberg: Springer-Verlag, 2006, pp. 69–80.
[11] M. Ouzzani and A. Bouguettaya, "Efficient access to web services," *IEEE Internet Computing*, vol. 8(2), pp. 34–44, 2004.
[12] Y. P. Chen, Z. Z. Li, Y. Z. Tang, and Z. S. Guo, "A method satisfying Markov process of Web service composition under incomplete constrains," *Chinese Journal of Computer*, vol. 29(7), pp. 1076-1083, 2006.
[13] Y. J. Seo, H. Y. Jeong, and Y. J. Song. "Best Web service selection based on the decision making between QoS criteria of service," in *Proc. of the ICESS 2005*. LNCS 3820, L. T. Yang, X. S. Zhou, W. Zhao, Z. H. Wu, Y. Zhu, et al., Eds. Heidelberg: Springer-Verlag, 2005, pp. 408-419.
[14] B. Cohen. *BitTorrent Plus! 2.130.2.110*. http://www.bitwonder.com
[15] Y. M. Shen and Y. S. Fan, "Cooperative mixed strategy for service selection in Service Oriented Architecture," in *Proc. of 2007 IEEE International Conference on Systems, Man, and Cybernetics*, M. El-Hawary, Eds. Montreal: Omnipress, 2007, pp. 1452–1457.
[16] X. G. Liu, Y. Jin. "Research on non-functional conditions-based Web services selection in Web services automation". *Computer Integrated Manufacturing Systems*, vol. 12(2), pp. 297-301, 2006 (in Chinese with English abstract).
[17] D. Gross and C. M. Harris, *Fundamentals of Queuing Theory*, New York: John Wiley & Sons, 1985, pp.79.

Yimin Shen was born in Xianju, Zhejiang, China, 1968. He obtained his B.S. degree in applied mathematics from Zhengjiang University, Hangzhou, Zhejiang, China, in 1990; M.S. degree in computer application from Chengdu Institute of Computer Application (CICA), the Chinese Academy of Sciences, Chengdu, Sichuan, China, in 1993; and Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China, in 2008, respectively.

He worked in CICA as Associate Researcher, Vice Director of Computer Integrated Manufacturing System (CIMS) Laboratory, and Director of Software and Communication Department from 1993 to 2001. He is currently Professor of Computer Engineering Department, Director of Software Institute in Chengdu Electromechanical College, Chengdu, Sichuan, China. He has published about 30 journal papers and conference papers. His research interests include applied mathematics, business process reengineering, system integration, and software engineering.

Dr. Shen is a member of Sichuan Software Council, member of the Network and Information System Committee of Southwest Computer Society, and member of Sichuan Information Network of CIMS, China.

Jing Zhang was born in Zigong, Sichuan, China, 1969. She received her B.S. degree in computer science and application from Southwest Jiaotong University, Chengdu, China, in 1991, M.S. degree in software engineering from University of Electronic Science and Technology of China, Chengdu, China in 2006, respectively.

She worked as a software engineer in Chengdu Aero-Instrument Cooperation of Ministry of Aerospace Industry from 1991 to 1997. She joined the Department of Computer Engineering, Chengdu Electromechanical College in 1997 and she is now an associate professor. She has published about 20 papers in refereed journal and conferences. She is an author or co-author of 8 textbooks. She was the principal investigator and co-investigator of many projects. Her research interests include interdisciplinary applications in software engineering, computational intelligence, pervasive computing, and intelligent buildings.

Yushun Fan was born in Yangzhou, Jiangsu, China, 1962. He received the B.S. degree in automatic control from Beijing University of Aeronautics and Astronautics, Beijing, China, in 1984; and the M.S. and Ph.D. degrees in control theory and application from Tsinghua University, Beijing, China, in 1987 and 1990, respectively. From September 1993 to March 1994, he was a Visiting Scientist in University Bochum, German, supported by Federal Ministry for Research and Technology. From April 1994 to July 1995, he was a Visiting Scientist, supported by Alexander von Humboldt Stiftung, at Fraunhofer Institute for Production System and Design Technology (FhG/IPK), German.

He is currently Professor of Department of Automation, Vice Director of System Integration Institute, and Director of the Networking Manufacturing Laboratory in Tsinghua University. He is also guest Professor of Shanghai Jiaotong University, Southwest Jiaotong University, Shenyang Institute of Automation of the Chinese Academy of Sciences. He authored nine books and published more than 250 research papers in journals and conferences. His research interests include enterprise modeling methods, networking manufacturing, optimization analysis, and flexible software systems.

Prof. Fan is a member of the IFAC Advanced Manufacturing Technology Committee and vice director of China Standardization Committee for Industrial Automation System and Integration, vice director of Manufacturing Technology Committee of China Automation Association.