From Quality to Utility: Adaptive Service Selection Framework

Chung-Wei Hang and Munindar P. Singh

Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA
{chang,singh}@ncsu.edu

Abstract. We consider an approach to service selection wherein service consumers choose services with desired nonfunctional properties to maximize their utility. A consumer's utility from using a service clearly depends upon the qualities offered by the service. Many existing service selection approaches support agents estimating trustworthiness of services based on their quality of service. However, existing approaches do not emphasize the relationship between a consumer's interests and the utility the consumer draws from a service. Further, they do not properly support consumers being able to compose services with desired quality (and utility) profiles.

We propose an adaptive service selection framework that offers three major benefits. First, our approach enables consumers to select services based on their individual utility functions, which reflect their preferences, and learn the providers' quality distributions. Second, our approach guides consumers to construct service compositions that satisfy their quality requirements. Third, an extension of our approach with contracts approximates Pareto optimality without the use of a market mechanism.

1 Introduction

We consider the problem of service selection. In service-oriented environments [16], consumers consume services for direct interaction or composition. How to select the "right" services to consume is one of the main challenges in service-oriented computing. The "right" services are determined based on two kinds of properties: functional and nonfunctional. *Service description matching* provides design-time automation for consumers to discover services with desired functionalities. By contrast, our emphasis is on *service selection*, which offers run-time automation for consumers to select services with the desired nonfunctional properties—quality of service (QoS), to meet consumers' preferences.

Three main challenges arise in service selection. First, how do consumers collect information about the QoS offered by a particular service? In open settings, quality information can be collected based on either consumers' direct experience or third-party referrals. However, learning from direct experience with a dynamic service (which delivers differing quality from time to time) is not trivial. Also, how consumers accommodate false referrals provided by malicious referrers is challenging. Second, how do consumers define their preferences for QoS? Potentially, each consumer may have a different requirements. For example, one may prefer high throughput, another low throughput, and yet another intermediate throughput. Another difficulty is accommodating the consumers' goals and context. For example, a latency of five seconds may be acceptable for a galaxy simulation but unacceptable for a web search. Third, how do consumers make selection decisions based on quality information and consumers' preferences? Selecting the apparently best services may not always lead to long-term success, because consumers have limited or incomplete knowledge, and services may change their behavior.

Trust models provide a promising solution to the first and third challenges. Trust models enable consumers to estimate the trustworthiness of dynamic services in terms of QoS based on both direct and indirect evidence. We particularly consider trust models, e.g., [10, 19, 21], that provide certainty measurement as an indicator of how confident the estimated trustworthiness is.

Many trust-based service selection approaches have been proposed [6, 9, 11–13, 20]. However, many of them fail to deal with the second challenge practically. Maximilien and Singh [13] describe consumers' preferences of QoS in terms of utility theory. They only consider that consumer preferences follow a monotonic increasing function of QoS (i.e., the higher the quality the better). In practical cases, the utility function of QoS may not always be increasing (say, for price), or even monotonic. For example, a consumer may prefer medium over high or low capacity. Low capacity may lead to long response time because many consumers are waiting. In contrast, high capacity may lead to long latency because the service may not have sufficient buffer space to handle many requests at the same time.

Many service selection approaches, e.g., [9, 12, 13], guide consumers to select services for direct interaction but not for composition. Selecting for composition is nontrivial because the composition changes the context and thus can affect a consumer's preferences. For example, a consumer may prefer low latency for interaction but may prefer two subservices composed in parallel to have approximately equal latency so as to reduce the need for buffering. But in a sequential composition, the consumer may still prefer low latency for each service. Consequently, a service selection method should guide consumers to adjust their preferences based on the composition and quality being considered.

This paper proposes a service selection framework that enables consumers to select services based on the utility they expect to gain from the services. Our approach addresses all three of the above challenges. First, our approach incorporates any probabilistic trust model to collect quality information about the services. Second, our approach enables consumers to specify utility functions describing their preferences. Third, our approach provides a learning policy to explore and exploit desired services in order to maximize each consumer's utility based on its preferences and estimated quality of the services involved. We show experimentally that our approach helps achieve near Pareto optimality across the consumers. Besides, our approach can guide a consumer to adjust its preferences for desired compositions based on its quality requirements and composition types.

Section 2 surveys the relevant literature. Section 3 defines our problem and scope. Section 4 formalizes our approach. Section 5 evaluates our approach via simulations. Section 6 discusses our results and highlights future directions.

2 Related Work

Maximilien and Singh [13] study trust-based service selection involving multiple qualities. Their approach considers consumers' utility of each quality, and enables consumers to define preferences between qualities. However, Maximilien and Singh only consider two shapes of utility functions: *linear* and *sigmoid*. Our approach can take any utility function as input.

ServiceTrust [9] calculates reputations of services from all consumers. It introduces *transactional trust* to detect QoS abuse, where malicious services gain reputation from small transactions and cheat at large ones. However, ServiceTrust models transactions as binary events (success or failure), and combines reports from all consumers without taking their preferences into account.

Malik et al. [12] propose a reputation assessment approach based on Hidden Markov Models (HMMs). They maps qualities to each consumer's personal evaluations that incorporate its preferences by weighing preferred qualities higher. Next, they calculate an aggregated reputation based both on personal evaluations and ratings by others. Based on the aggregated reputation, Malik et al. apply HMM to predict future behavior. Instead of using weights, our approach reflects consumers preferences via utility functions. Our approach can also incorporate other consumers' quality ratings by choosing probabilistic trust models that support referrals to model our quality distribution. For example, our approach can adopt Wang and Singh's trust model [19], which provides trust propagation [8] and trust updates to deal with referrals from other consumers [7].

Li et al. [11] estimate the trustworthiness of composite services using Bayesian inference. They formalize a variety of service invocations and convert compositions into a service invocation graph. Then consumers can construct a desired composite service by finding the optimal service execution flow. Li et al. express trust as a normal distribution. and apply Bayesian inference to maintain trust. Our approach models a service's QoS as a probability distribution, which is also learned and updated via Bayesian inference. However, our approach deals with composition differently. Consumers select subservices for composition based on utility. More specifically, a consumer adjusts its utility function based on the already selected subservices to achieve better orchestration.

ServiceRank [20] ranks services from two aspects: quality of service and social information. ServiceRank considers two qualities: response time and availability. Social information includes (1) how many consumers a service has and (2) how frequently a service is invoked. Similar to ServiceTrust [9], ServiceRank models transactions as binary events. Our approach does not limit to any particular qualities and selects services with the highest expected utility of multiple qualities. In our approach, an analog of social information is characterized via the certainty of the quality distributions. Certainty measurements can be found in many probabilistic trust models [10, 19, 21]. Certainty reflects how much evidence a quality distribution has. If a service is seldom invoked or has few customers, then the certainty of its quality distribution will be low.

Hang and Singh [6] present a trust model for service selection. Their model focuses on how to estimate the trustworthiness of subservices based on the observed quality of composite services. Hang and Singh introduce *composition operators* that define how a quality is composed from subservices to compositions. They show their model can accurately estimate trustworthiness of subservices under various composition operators. Hang and Singh fail to specify how consumers interpret a quality as trust. Our approach builds trust based on the reputations of services (i.e., their quality distributions) and consumers' preferences (i.e., utility functions). Following Hang and Singh's formalization, our approach can be used to select services for composition by adjusting utility functions. Depending on the composition operator and existing subservices in the composition, the utility function is adjusted differently.

Gerding et al. [5] design mechanisms for service procurement. They consider the case where services may fail. Their mechanism guides consumers to procure services to form a workflow for completing a task with constraints, for example, within a certain time. The consumers aim to (1) maximize the probability of success given the constraints and (2) balance success probability with costs. Their work focuses on designing a mechanism to incentivise service providers to reveal their quality information, whereas, in our approach, consumers collect such information by themselves. Consumers apply probabilistic trust models to collect quality information of providers from direct interaction, referrals, and compositions. We show that our approach guides service procurement and leads to Pareto optimality, i.e., resources are allocated efficiently.

3 Problem Description and Scope

Environment A service-oriented environment includes a set of providers $\mathcal{P} = \{ P_1, \ldots, P_m \}$ and a set of consumers $\mathcal{C} = \{ C_1, \ldots, C_n \}$. The providers provide services of the same underlying functionality, but each provider potentially offers different levels of the l qualities in $\mathcal{Q} = \{Q_1, \ldots, Q_l\}$. A service composition is defined as $\chi = \langle \mathcal{S}, \mathcal{T} \rangle$, where $\mathcal{S} \subseteq \mathcal{P}$ are services, and the composition type $\mathcal{T}$ is one of *sequence*, *flow*, and *case*, as in typical orchestrations [2].

Assumption We assume the value of quality Q_k offered by provider P_j is governed by a probability distribution that is independent of the consumer and the service composition in consideration.

Objective Each consumer defines a utility function for each quality to describe its preferences and selects services to maximize its utility.

Scenario 1: Select Services for Direct Interaction A consumer C_i would like to interact with a service from $\mathcal{P}$ to maximize its utility regarding $\mathcal{Q}$.

Scenario 2: Select Services for Composition We are given a partial composition $\chi' = \langle \mathcal{S}', \mathcal{T} \rangle$ with existing subservices $S' \subseteq \mathcal{P}$, and a consumer C_i's utility function regarding $\mathcal{Q}$. Now, C_i would like to create a "supercomposition" of χ' as $\chi = \langle \mathcal{S}' \cup \mathcal{S}'', \mathcal{T} \rangle$ that builds on χ' by adding services $\mathcal{S}''$ to the composition.

4 Approach

We propose an adaptive service selection framework that enables consumers to select providers to maximize their utility based both on their preferences of qualities and what they learn about the providers' quality distributions.

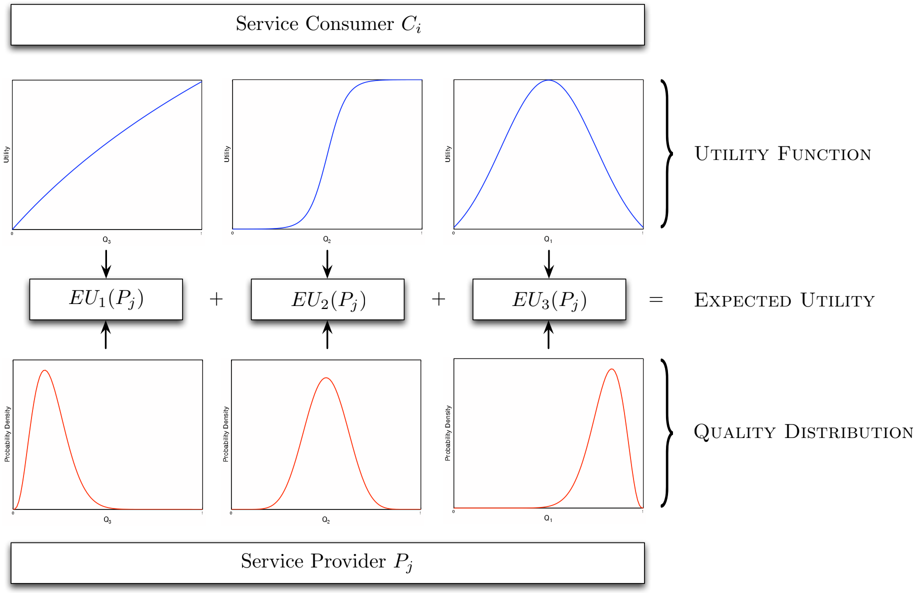


Fig. 1. Illustration of our approach. Consumer C_i describes its preferences of qualities via utility functions, collects information about and learns the distributions of all qualities of each provider, and calculates the expected utility of each provider.

Figure 1 illustrates the service selection process. For a consumer C_i, its *utility functions* with respect to each quality $Q_k \in \mathcal{Q}$ are given. Next C_i applies the following steps.

1. Collect quality information and learn the *quality distributions* of each provider $P_j \in \mathcal{P}$ with respect to each quality $Q_k \in \mathcal{Q}$.

2. Calculate *expected utility* of each provider $P_j \in \mathcal{P}$ with respect to all qualities $Q_k \in \mathcal{Q}$.
3. Select a provider $P_j \in \mathcal{P}$ to interact with.
4. Repeat Step 1.

In the remainder of this section, we (1) define utility functions and quality distributions; (2) introduce calculating expected utility involving multiple qualities; (3) relate the service selection strategy to the learning policy used in reinforcement learning; and (4) explain how our service selection can be used for selecting subservices in service compositions.

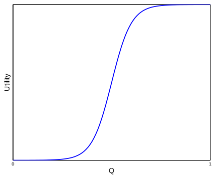
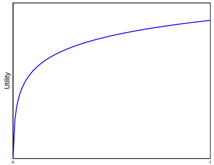
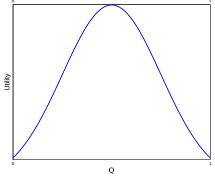
4.1 Describe Service Consumers: Utility Functions

We describe a consumer's preference of a particular quality by a utility function.

Definition 1. Utility function $U(x)$. *Let C_j be a consumer and random variable x be the volume of some quality of service Q_k. C_j's utility function $U(x)$ is defined as a function that maps Q_k's value x to C_j's utility.*

Table 1 shows some examples of utility functions.

Table 1. Example utility functions: Describing consumer preferences

Function	**Plot**	**Description**
Logistic [13] $\frac{1}{1+e^{-(x-u)/s}}$		Rapid increase close to desired value, as for availability
Logarithm $\log x$		Diminishing returns, as for throughput
Gaussian $\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$		Tradeoff between being idle or buffering heavily, as for throughput of subservice

4.2 Describe Service Providers: Quality Distributions

We capture the quality of service provided by a service provider P_j by introducing a quality distribution.

Definition 2. Quality distribution $Q_k(x)$. *Let P_j be a service and random variable x_j be the quality of service Q_k provided by P_j along some suitable dimension. Then P_j's quality function $Q_k(x_j)$ with respect to quality Q_k is defined as the probability density function of the probability distribution that governs x_j.*

The quality distributions can be learned by probabilistic trust models based on evidence regarding quality, which can be obtained through 1. direct experience [19], 2. referrals [8], or 3. composition [6, 14].

4.3 Calculate Expected Utility of Providers

The consumers' decision making is based on the utility they expect to obtain from a provider.

Definition 3. Expected utility with respect to a quality. *Let P_j's quality distribution be $Q_k(x_j)$, and C_i be a consumer with utility function $U_k(x)$. The expected utility of quality Q_k that C_i can obtain from P_j is defined as*

$$EU_k^i(P_j) = \int_{x_j} U_k(x_j)Q_k(x_j)dx_j. \tag{1}$$

The expected utility of provider P_j based on all its qualities is as follows.

Definition 4. Expected utility with respect to all qualities. *Suppose $Q_1(x)$, ..., $Q_l(x)$ are quality distributions of provider P_j and $U_1(x)$, ..., $U_l(x)$ are utility functions of consumer C_i. Then C_i's expected utility of P_j is defined as*

$$EU^i(P_j) = \sum_{k=1,\ldots,l} EU_k^i(P_j) = \sum_{k=1,\ldots,l} \int_{x_j} U_k(x_j)Q_k(x_j)dx_j \tag{2}$$

4.4 Select Services: Exploration vs. Exploitation

Based on the expected utility of a provider with respect to all qualities, a consumer C_i can decide on which providers to interact with to maximize its utility as follows:

$$P_j = \arg\max_{P_j \in \mathcal{P}} EU^i(P_j) \tag{3}$$

However, given the assumption that the consumer can only learn the quality distributions from direct experience, selecting the provider that yields the most utility may not lead to long-term success. This is because the consumer lacks evidence (direct experience) to learn accurate quality distributions of the providers.

Here, we address this challenge by modeling the service selection problem as the multiarmed bandit problem [15, 1]. An agent (gambler) seeks to maximize its reward by taking a series of actions (pulling levers from a multiarmed slot machine). At each instant, the reward is based on a probability distribution associated with each action. Notice that a myopic decision (choosing an action with the highest known reward) may not yield the most long-term reward. A *policy* is a function that predicts the (long-term) rewards from each action. Research in reinforcement learning [18] studies how to learn an optimal policy, by which the agent can select its actions. Doing so involves addressing the tradeoff between *exploration* (trying new alternatives that might lead to higher payoffs) and *exploitation* (making decision based on current knowledge).

We model the service-oriented environment as a multiarmed bandit, where the providers are the options and each consumer selects providers to maximize its long-term reward. Here, the reward is defined as the utility derived by the consumer and consumers learn a policy that maps actions (selecting a $P_j \in \mathcal{P}$) to their expected utility. To balance the above tradeoff between exploration and exploitation, we adopt *Boltzmann Exploration* [3], a widely used learning policy in reinforcement learning, written in our setting as:

$$Pr(P_j|EU^i) = \frac{e^{T*EU^i(P_j)}}{\sum_{P_i \in \mathcal{P}} e^{T*EU^i(P_i)}}, \tag{4}$$

where $P_j \in \mathcal{P}$ is a service, $EU^i(P_j)$ is C_i's expected utility of P_j, and T is a *temperature parameter*. The consumer C_i chooses service P_j with probability $Pr(P_j|EU^i)$. The idea of Boltzmann Exploration is for consumers to choose services based on their expected utility. The services with low expected utility (i.e., exploration) are chosen less frequently than those with high expected utility (i.e., exploitation). The actual percentage of exploration and exploitation can be adjusted by T. There are two ways of choosing T. For stationary environments, T increases over time to reduce the probability of exploration after the policy is learned. In contrast, for nonstationary environments, consumers may use a fixed T to ensure continual exploration. Satinder Singh et al. [17] show a convergent result for choosing T. Applying their result to our case, we can define T as $\ln t/C_t$, where t is the timestep, $C_t = \max_{P_j \in \mathcal{P}} |EU^i(P_{max}) - EU^i(P_j)|$ and $P_{max} = \arg\max_{P_j \in \mathcal{P}} EU^i(P_j)$.

4.5 Selecting Services for Composition

Now we describe how our approach guides consumers to select services for composition. Section 3 mentions a scenario where a consumer C_i would like to add services $\mathcal{S}''$ from $\mathcal{P}$ to a current composition $\chi' = \langle \mathcal{S}', \mathcal{T} \rangle$ to maximize the utility of the resulting composition $\chi = \langle \mathcal{S}' \cup \mathcal{S}'', \mathcal{T} \rangle$ where $\mathcal{S}' \subseteq P$ is the set of constituent services of χ' and $\mathcal{T}$ is the composition type.

Following our previous work [6], we introduce the *composition operators* SWITCH, SUM, MAX, MIN, and PRODUCT. These operators determine the quality values of a composite service based on the quality values of the constituent services. They depend upon the composition type (*sequence*, *flow*, or *case*), and on the nature of the quality being considered. Table 2 shows examples of composition operators given the types of compositions and qualities.

Based on the expected quality of the existing subservices $\mathcal{S}'$ in a composition χ, consumer C_i can adjust the utility function $U_k(x)$ of the composition with respect to Q_k. This adjustment depends on (1) the composition operator f, (2)

Table 2. Mapping composition types and representative qualities to operators [6].

Quality	Sequence	Flow	Case
Latency	SUM	MAX	SWITCH
Throughput	MIN	SUM	SWITCH
Failure	PRODUCT	PRODUCT	SWITCH

composition χ's preferences of the quality Q_k, and (3) the expected quality $x_{\mathcal{S}'}$ of existing subservices $\mathcal{S}'$. For example, let f be MAX, the maximum of expected quality Q_k of $\mathcal{S}'$ be $E[x_{P_{max}}]$, and χ's original utility function be $U_k(x)$, Then the adjusted utility function U'_k should be

$$U'_k(x) = \begin{cases} U_k(E[x_{P_{max}}]) & \text{if } x < E[x_{P_{max}}] \\ U_k(x) & \text{otherwise.} \end{cases} \tag{5}$$

The idea behind this adjustment is that if the quality x provided by the new subservices is lower than $E[x_{P_{max}}]$, then the composite quality x_χ is dominated by P_{max} because $x_\chi = \max(x, E[x_{P_{max}}]) = E[x_{P_{max}}]$. Thus, the composite utility will be $U_k(E[x_{P_{max}}])$. Conversely, if $x > E[x_{P_{max}}]$, then the composite quality x_χ is dominated by x. The composite utility will be $U_k(x)$.

Table 3 shows utility function examples for various composition operators. It shows how monotonically increasing and decreasing utility functions are adjusted based on existing constituent services and the composition types. Note that our approach is not limited to these composition operators or utility functions.

5 Experiments

We conduct three experiments to evaluate the effectiveness of our approach. The first experiment examines how accurately the expected utility can predict the actual gained utility. The second experiment verifies the selection strategy described in Section 4.4 by checking if the consumers can quickly learn the quality distributions of all providers (exploration), and select the provider with the highest expected utility afterwards (exploitation). The third experiment adopts a resource allocation perspective and examines how closely our approach can lead to *Pareto optimal* allocations [4].

5.1 Two Consumers, Three Providers, One Quality, No Selection

We consider three providers: *Low*, *Medium*, and *High*, whose quality Q is governed by beta distributions with low, fair, and high means, respectively. We create two consumers, respectively with utility functions *Logistic* (prefers high over low Q) and *Gaussian* (prefers mid range Q over high or low values). At each timestep, each consumer interacts with all providers once, and learns their quality distributions. Figure 2 plots the expected and gained utility of the *Logistic* (left) and *Gaussian* (right) consumers from provider *Medium* at each timestep. *Medium* provides average Q_1 values, yielding utility around 0.50 for *Logistic*, whose gained utility has a high variance, because *Logistic* is highly sensitive to Q_1. *Medium* brings a fair amount of utility to *Gaussian*, who is less picky. The expected utility calculated by both consumers accurately predicts the actual gained utility. The experiment has total 100 timesteps. *Low* and *High* providers yield similar results. The result verifies that the expected utility is an effective basis to maximize consumers' utility.

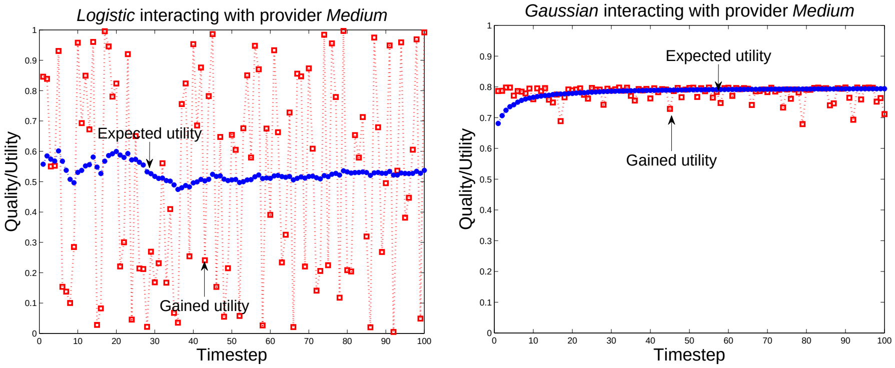


Fig. 2. The expected and gained utility of *Logistic* (left) and *Gaussian* (right) against *Medium*. Both consumers receive the same QoS from *Medium* but gain different utility based on their utility functions. *Logistic* is sensitive to quality, and thus has high variance in its gained utility. This experiment shows the expected utility correctly predicts the actual gained utility. The other providers yield similar results.

5.2 Two Consumers, Three Providers, One Quality, Selection

This experiment helps us evaluate the effectiveness of the strategy described in Section 4.4. Its setting follows Section 5.1, except that the consumers choose only one provider instead of all to interact with. Each consumer seeks to maximize its long-term utility. Figure 3 shows *Logistic*'s (left) and *Gaussian*'s (right) expected utilities of all providers and the actual utility gained from the sole provider each chooses to interact with at each timestep. *Logistic* explores all providers at the beginning, gaining low utility at the beginning. After it determines *High* is its desired provider, *Logistic* selects *High* much more than the other providers. *Gaussian* has a similar strategy as *Logistic* at the beginning. However, since *Gaussian* is not as picky as *Logistic*, *Gaussian* chooses all providers almost equally often, but because *Medium* offers a slightly better expected utility, *Gaussian* selects *Medium* slightly more often than the other two. Both consumers gain utility close to optimal.

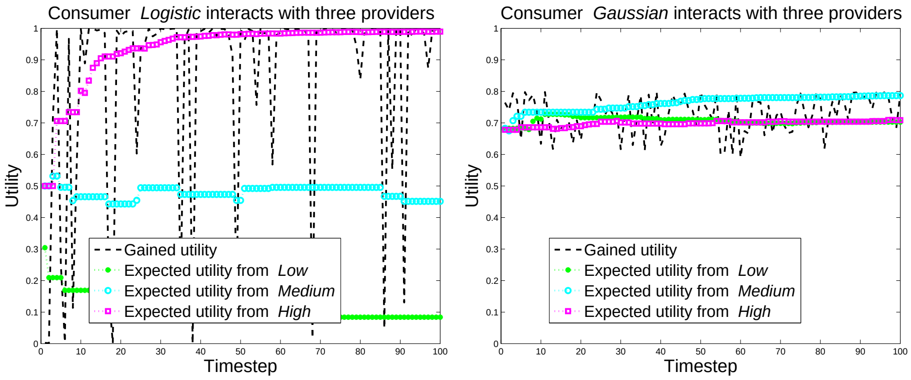


Fig. 3. The expected and gained utility of *Logistic* (left) and *Gaussian* (right) consumers against three providers. In each timestep, consumers select one of the providers to interact with. This shows each consumer learns the expected utility of all providers (exploration), and subsequently interacts with the desired provider to maximize its utility (exploitation).

Figure 4 plots the comparison of optimal and actual gained utility of two consumers. Our approach learns the quality distribution through greater exploration at the beginning followed by greater exploitation subsequently. Because the gained utility is smaller than the optimal when the consumers choose to explore (or when the quality distribution is not learned accurately), the utility increases over time.

This result shows our service selection approach not only efficiently learns the quality distributions of the providers, but also exploits the desired provider to maximize each consumer's utility.

5.3 Contracts as a Means to Approximate Pareto Optimality

A resource allocation is *Pareto optimal* if and only if it is impossible to alter the allocation that would offer increased utility to one agent without offering reduced utility to another. In other words, no two agents can trade (their allocations) where they would both gain. Pareto optimality is a widely used indicator of economic efficiency. The intuition behind it is that we should allocate each resource to the party who values the resource the most. To ensure Pareto optimality typically requires some market mechanism such as an auction that takes bids from all consumers and allocates resources in a way to guarantee efficiency. For example, allocating each resource to the highest bidder means that the bidder who values the resource the most gets to use it.

We show how a natural extension of our approach with contracts leads to approximating Pareto optimality despite independent decision making by the consumers and without any central mechanism such as an auction clearinghouse. In an important departure from the previous experiments, here a consumer may

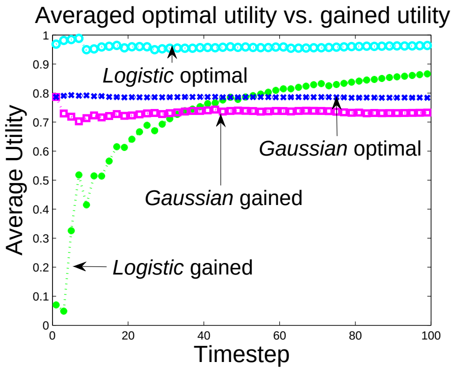


Fig. 4. Comparison of optimal and gained utilities of *Logistic* and *Gaussian*. Consumers choose nonoptimal services for exploration. This shows that when expected utilities of providers are close, the consumers are more willing to explore.

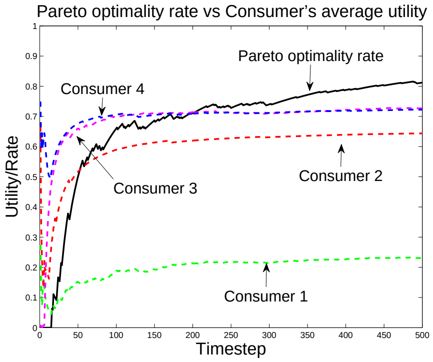


Fig. 5. The rates of Pareto optimality and consumers' average utility with four consumers and three providers, each of capacity two. This shows that even though consumers find desired services and maximize utility independently, together they achieve a Pareto optimal allocation 80% of the time.

sign a long-term contract with a provider. A contract is a way for a consumer to exploit (i.e., select the provider with most expected utility). We consider an exploitation strategy where the contract length is proportional to (ten times as) the expected utility. When a consumer chooses to explore, it interacts with the selected provider for one timestep. The intuitive motivation is that the consumers who value a provider the most will sign the longest contracts with it and thus gain the greatest utility from it for the longest time.

We create four consumers with different preferences of two qualities: throughput and price. Intuitively, the consumers vary from rich (prefers the highest throughput at any price) to poor (prefers to save money and accepts a little throughput). The consumers have *logistic* utility functions (monotonically increasing) for throughput with parameters $(\mu, s) = (0.8, 0.05), (0.6, 0.05), (0.4, 0.05)$, $(0.2, 0.05)$, respectively. For price, each consumer has a negative exponentially decreasing utility function, with σ of 5, 4, 3, and 2, respectively:

$$U_{price}(x) = -\frac{x^{\sigma}}{100^{\sigma}} \qquad (6)$$

Three providers provide throughput governed by beta distributions with means 0.75, 0.5, and 0.25, and asking \$75, \$50, and \$25 for their service, respectively.

The experiment runs for 500 timesteps. At each timestep, the consumers select one provider to interact with. A provider can serve two consumers at a time. We check if the resource allocation at each timestep is Pareto optimal.

For each timestep, Figure 5 shows the consumers' average utility and the cumulative percentage of instantaneous allocations that are Pareto optimal. We see that the rate of Pareto optimality increases as the average individual utility increases. This indicates that when the consumers maximize their utility, Pareto optimality is likely to be achieved. Pareto optimality is not achieved in every instance of exploitation because a consumer may not be able to sign a contract with its best provider who might be under another contract or because of the effects of randomness in quality. In general, though, this experiment shows that our approach can approximate Pareto optimality by selecting services in an economically efficient way.

6 Conclusions and Directions

This paper proposes an adaptive service selection approach, which enables service consumers to 1. collect quality information of service providers efficiently, 2. describe their preferences of quality by defining utility functions, and 3. select "right" service providers to maximize their long-term utility. Besides, our approach can guide consumers to select services for composition by defining utility functions for service compositions. Importantly, our approach yields a natural way to approximate Pareto optimality of service selection.

Important directions for future study include the generalization of our models of quality and utility. In particular, we hypothesize that if we allow the providers to change their quality profiles dynamically, that will yield protection against greedy consumers signing arbitrarily long contracts with the best current service and thus blocking other consumers.

Acknowledgment

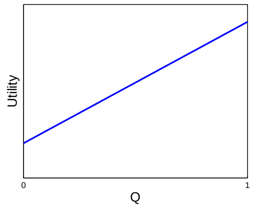
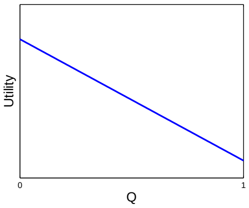
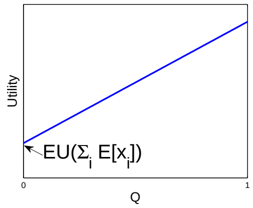
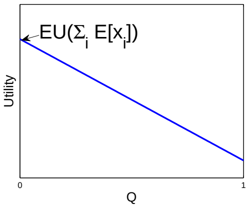
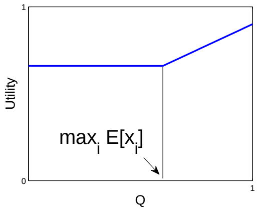
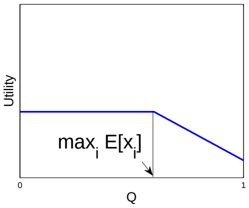
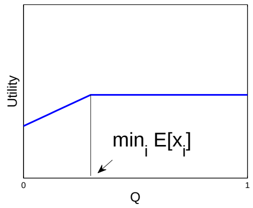
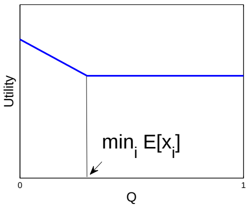
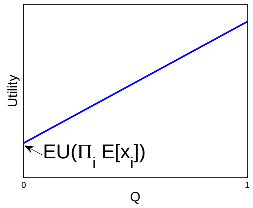
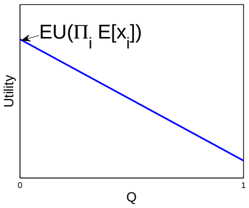
This work is supported by the U.S. Army Research Office (ARO) under grant W911NF-08-1-0105 managed by NCSU Secure Open Systems Initiative (SOSI).

References

1. Berry, D.A., Fristedt, B.: Bandit Problems: Sequential Allocation of Experiments. Monographs on statistics and applied probability, Chapman and Hall, London, UK (1985)
2. BPEL: Web services business process execution language, version 2.0 (Jul 2007), http://docs.oasis-open.org/wsbpel/2.0/
3. Bridle, J.S.: Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. Advances in neural information processing systems 2 pp. 211–217 (1990)
4. Feldman, A.M.: Welfare Economics and Social Choice Theory. Kluwer Nijhoff Publishing (1980)
5. Gerding, E., Stein, S., Larson, K., Rogers, A., Jennings, N.R.: Scalable mechanism design for the procurement of services with uncertain durations. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). pp. 649–656. IFAAMAS, Columbia, SC (2010)

6. Hang, C.W., Singh, M.P.: Trustworthy service selection and composition. ACM Transactions on Autonomous and Adaptive Systems (TAAS) To appear (2010)
7. Hang, C.W., Wang, Y., Singh, M.P.: An adaptive probabilistic trust model and its evaluation. In: Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems. pp. 1485–1488. IFAAMAS, Columbia, SC (2008), short paper
8. Hang, C.W., Wang, Y., Singh, M.P.: Operators for propagating trust and their evaluation in social networks. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). pp. 1025–1032. IFAAMAS, Columbia, SC (2009)
9. He, Q., Yan, J., Jin, H., Yang, Y.: ServiceTrust: Supporting reputation-oriented service selection. In: Proceedings of the 7th International Conference on Service Oriented Computing. pp. 269–284 (2009)
10. Jøsang, A.: A subjective metric of authentication. In: Proceedings of the 5th European Symposium on Research in Computer Security (ESORICS) (1998)
11. Li, L., Wang, Y., Lim, E.P.: Trust-oriented composite service selection and discovery. In: Proceedings of the 7th International Conference on Service Oriented Computing. pp. 50–67 (2009)
12. Malik, Z., Akbar, I., Bouguettaya, A.: Web services reputation assessment using a hidden markov model. In: Proceedings of the 7th International Conference on Service Oriented Computing. pp. 576–591 (2009)
13. Maximilien, E.M., Singh, M.P.: Agent-based trust model involving multiple qualities. In: Proceedings of the 4th International Conference on Autonomous Agents and Multiagent Systems. pp. 519–526. IFAAMAS, Columbia, SC (2005)
14. Nepal, S., Malik, Z., Bouguettaya, A.: Reputation propagation in composite services. In: Proceedings of the 7th IEEE International Conference on Web Services (ICWS). pp. 295–302. IEEE Computer Society (July 2009)
15. Robbins, H.: Some aspects of the sequential design of experiments. Bulletin of the American Mathematical Society 55, 527–535 (1952)
16. Singh, M.P., Huhns, M.N.: Service-Oriented Computing: Semantics, Processes, Agents. John Wiley & Sons, Chichester, UK (2005)
17. Singh, S.P., Jaakkola, T., Littman, M.L., Szepesvári, C.: Convergence results for single-step on-policy reinforcement-learning algorithms. Machine Learning 38(3), 287–308 (2000)
18. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press (1998)
19. Wang, Y., Singh, M.P.: Formal trust model for multiagent systems. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI). pp. 1551–1556. IJCAI, Detroit, MI, USA (2007)
20. Wu, Q., Iyengar, A., Subramanian, R., Rouvellou, I., Silva-Lepe, I., Mikalsen, T.A.: Combining quality of service and social information for ranking services. In: Proceedings of the 7th International Conference on Service Oriented Computing. pp. 561–575 (2009)
21. Zacharia, G., Maes, P.: Trust management through reputation mechanisms. Applied Artificial Intelligence 14(9), 881–907 (2000)

Table 3. Utility function examples for composition operators based on two linear utility function: *prefer higher* and *prefer lower*. $E[x_i]$ is the expected quality from provide P_i.

Composition	Utility Function	
Operator	Prefer Higher	Prefer Lower
SWITCH		
SUM		
MAX		
MIN		
PRODUCT	 Utility EU(Πi E[xi]) 0 1 Q	 EU(Πi E[xi]) Utility 0 1 Q