# Coalition Deal Negotiation for Services

Jiangbo Dang
*University of South Carolina*
*Columbia, SC 29208 USA*
*dangj@engr.sc.edu*

Michael N. Huhns
*University of South Carolina*
*Columbia, SC 29208 USA*
*huhns@sc.edu*

**Abstract**

*This paper investigates an optimal strategy for multiple-issue negotiation for services between interacting agents (service requestors and service providers). We assume the agents are self-interested. They negotiate over both functionality and QoS (quality of service) issues to reach a service agreement, while maximizing their own utilities. We assume each agent only knows its own negotiation parameters. By combining the merits of issue-by-issue negotiation and package deal negotiation, we propose a coalition deal negotiation to optimize the agents' utilities with minimized computational cost. We prove that the coalition deal makes a better tradeoff between issue-by-issue negotiation and package deal negotiation by providing both approximately optimal utility and efficient computation. In addition, we discuss optimal strategies for service-oriented negotiation in a competitive environment.*

## 1: Introduction

In supply chains, e-commerce, and Web services, the participants negotiate contracts and enter into binding agreements with each other by agreeing on functional and quality metrics of the services they request and provide. The functionality of a service is the most important factor, especially for discovering services. Once discovered, however, services are engaged, composed, and executed by the participants' negotiating over QoS metrics to maximize their profits.

Negotiation is a process by which agents communicate and compromise to reach agreement on matters of mutual interest, while maximizing their individual utilities. Negotiation for QoS-aware services is currently limited to primitive QoS verification methods or sorting and matching algorithms. We extend current techniques by presenting an optimal negotiation procedure that considers the cost to reach an agreement for QoS-aware service engagement and contracting.

### 1.1: Research Issues

In general, negotiation is a technique for reaching mutually beneficial agreement among parties via communication. Negotiation in QoS-aware services involves a sequence of information exchanges between the parties to establish a formal agreement among them, whereby one or more parties will provide services to one or more other parties. The agreement typically involves QoS issues [1]. By QoS, we refer to the non-functional properties of services, such as performance, cost, reliability, and security. To meet the requirements of service requestors, multiple issues, including both functional and non-functional, need to be taken into account during service advertisement, discovery, composition, and delivery. Preist [10] has discussed how negotiation plays an important role in reaching a service agreement for a semantic Web service. In this paper, we focus on the optimal strategy of efficiently negotiating multiple issues to reach an agreement that gives both a requestor and a provider their maximum utilities.

Many researchers have investigated multiple-issue negotiation [4, 7, 3] . Fatima et al. [3] presented an optimal agenda and procedure for two-issue negotiation by introducing two negotiation procedures: *issue-by-issue negotiation* and *package deal*. For $n$-issue negotiation where $n > 2$, which is common in service-oriented negotiation, the computational cost to reach a package deal might exceed the benefits obtained by optimizing the participants' utilities. By considering both utility optimization and computational efficiency, we propose the *coalition deal* that is suitable for multiple-issue negotiation, especially in the case of multi-issue negotiation for services [2].

### 1.2: Service Negotiation Scenario

In order to illustrate the coalition deal for $n$-issue negotiation over the QoS metrics of a service, we present a motivating scenario. Consider how one site, a requestor, might arrange to get a stock quote from a service provider. In this scenario, a service requestor $a$ locates a *GetStockQuote* service provided by $b$ that meets its functionality requirements. The *GetStockQuote* service takes the requestor's inquiring stock number as an input and a currency symbol as an argument, and provides a stock quote.

During the service selection procedure, QoS becomes an important factor to both $a$ and $b$. Before agreeing to a service contract, they need to negotiate over (1) **payment method**, which indicates the way a user pays for inquiries (e.g., pay per inquiry or pay for a bundle); (2) **inquiry cost**, which indicates the cost per inquiry; (3) **update interval**, which represents how often the stock quote information is updated; (4) **response time**, which is the round-trip time between sending an inquiry and receiving the response; (5) **availability**, which represents the probability that the service is available and ready for immediate use; (6) **service plan cost**, which is the plan cost for a service with agreed-upon quality.

Agents $a$ and $b$ could negotiate each issue individually using issue-by-issue negotiation, but some issues are related to each other and isolating them will degrade utility and increase the risk of a conflict deal. A package deal allows both $a$ and $b$ to make trade-offs among all six issues, but the computation is intractable with exponential cost. By using a coalition deal, we can divide the six issues into two partitions where strongly related issues are in the same partition. For example, payment method, inquiry cost, and update interval belong to partition one, while response time, availability, and service plan cost belong to partition two. $a$ and $b$ can negotiate both partitions in parallel, where each partition is settled as a package deal and independently of the other partition. By pursuing a coalition deal, agents can reach a service agreement while optimizing their utilities with efficient computation. In the context of service-oriented negotiation, the coalition deal is explored in the remainder of this paper.

## 2: Background and Related Work

A typical real world service-oriented environment is dynamic, competitive, and partially observable. Semantic Web services, as envisioned by Berners-Lee, are intended to be applied not statically by developers, but dynamically by the services themselves through automatic and autonomous selection, composition, and execution. Dynamic selection and composition first require service requestors to discover service providers that satisfy the requestors' functional requirements. Second, the requestors and providers negotiate non-functional requirements (QoS), including cost and qualities such as response time, accuracy, and availability.

As one step toward real-world service-oriented computing, many efforts have been made to automate service negotiation in a Web service environment. Current standards for Web services do not support automated QoS negotiations. As a result, several researchers have attempted to merge negotiation from the MAS domain into QoS-aware Web services. Ran [13] proposes to enrich current UDDI registries by extending the SOAP message format and the UDDI data structures to describe QoS information. Petrone [9] proposed a conversation model to enrich the communication and coordination capabilities of Web services by adapting agent-based concepts to the communications among Web services and users. Maximilien and Singh [6] propose a Web service agent framework (WSAF) with a QoS ontology. When a service consumer needs to use a service, WSAF will create a service agent that can capture a consumer's QoS preferences and select the most suitable service.

Service negotiation involves both functional and non-functional issues. We cannot apply existing multiple-issue negotiation models to service negotiation and contracting directly, because existing models often make the limiting assumption that agents know the private information of their opponents, and their theoretic models do not take computational cost into consideration. Therefore, these models do not fit typical competitive environments, where self-interested agents engage in on-line QoS negotiation for services.

In [4], agents know the incomplete preference information about their opponents and exploit this information to improve negotiation efficiency. This work is thus limited to cooperative negotiation, where agents care about not only their own utilities, but also equity and social welfare, which is not the common case in Web service environments. [5] show the formation of a coalition is a dominant strategy under a proportionally fair divisible auction, by which agents bid for services and computational resources. A negotiation model is defined in [17] to allow agents to generate and evaluate proposals by employing a set of strategies and tactics within the model to reach an agreement for service provision. Sandholm and Lesser [16] discuss the issues in automated negotiation among bounded rational self-interested agents. In the context of task allocation negotiation, they present a negotiation protocol to support leveled commitment by introducing counter-proposals into the Contract Network Protocol.

The outcome of multiple-issue negotiation depends on not only strategies, but also the procedure by which issues will be negotiated. Different procedures yield different outcomes. Based on an incomplete information assumption, Fatima et al. [3] discussed two procedures for multiple-issue negotiation: *issue-by-issue* and *package deal*. For two-issue negotiation, they determined the equilibrium strategy for these procedures and analyzed the optimal agenda and procedure. They concluded that the package deal is the procedure that provides agents with optimal utilities for two-issue negotiation. They did not address the computational cost with increasing issue size. However, the computational cost becomes crucial when more issues are involved.

In this paper, we hypothesize that a coalition deal negotiation can overcome these limitations. We prove that it is the optimal strategy for negotiation over multiple issues for service when computation cost is taken into consideration. The coalition deal mitigates the computational cost problem by making a trade-off between optimal utility and computational efficiency. Therefore, it is the optimal strategy for agents with bounded rationality. This paper makes four contributions to the advancement of QoS-aware service negotiation and contracting. First, it describes the coalition deal negotiation for reaching utility optimization and computational efficiency. Second, it generalizes the analysis of an optimal negotiation procedure to multiple-issue negotiation over more than two issues. Third, it tailors negotiation components to fit QoS-aware service negotiation. Fourth, it focuses on agents' own information; no agent has any information, such as reserve price, about its opponent.

## 3: The Negotiation Model

Negotiation for services has four components: (1) a negotiation set, which represents the possible proposal space for both functionality and QoS metrics of a service; (2) a protocol, which defines the legal proposals that an agent can make, as defined in a service description and constrained by negotiation history; (3) a strategy, which determines what proposals the agents will make, decided by an agent's private preference and affected by the service discovery result; and (4) a rule enforced by a mediator to determine when a deal has been struck and what the agreement is. By envisioning a competitive environment with the self-interested agents, we focus on the procedure of multiple-issue negotiation, which adopts Rubinstein's alternating offers protocol [8] in this paper.

As described in our motivating scenario, let $a$ denote the service requestor and $b$ the service provider. We assume that each agent only has complete information about its own negotiation parameters. For some private information, such as the opponent's deadline, we can use the negotiation protocol in [15] to make truth telling about a negotiation deadline the dominant strategy. We use $S_a(S_b)$ to denote the set of negotiation parameters for agent $a(b)$ and start with the single-issue negotiation model described similarly to that in [3].

### 3.1: Single-Issue Negotiation

Consider $a$ and $b$ negotiating over an issue set $I$, where $I = \{A\}$ and $A$ is one single issue, say, the inquiry price. The agents' parameter sets are defined as

$$\begin{aligned} S_a &= \left\langle P_a^A, U_a^A, T_a^A, \delta_a^A \right\rangle \\ S_b &= \left\langle P_b^A, U_a^A, T_b^A, \delta_b^A \right\rangle \end{aligned} \tag{1}$$

Where $P_a^A$, $U_a^A$, $T_a^A$, and $\delta_a^A$ denote agent $a$'s reserve price over issue $A$, agent $a$'s utility function over issue $A$, agent $a$'s bargaining deadline, and agent $a$'s time discounting factor, respectively. Agent $b$'s negotiation parameters are defined analogously. Agents' utilities at price $p$ and at time $t$ are defined as in [3]:

$$U_a^A(p,t) = \begin{cases} (P_a^A - p)(\delta_a^A)^t & \text{if } t \leq T_a \\ 0 & \text{if } t > T_a \end{cases}$$

$$U_b^A(p,t) = \begin{cases} (P_b^A - p)(\delta_b^A)^t & \text{if } t \le T_b \\ 0 & \text{if } t > T_b \end{cases} \tag{2}$$

When agent $a$ is patient and will gain utility with time, $\delta_a^A > 1$. When agent $a$ is impatient and will lose utility with time, $\delta_a^A < 1$. When agent $a$'s utility is independent of time $t$, $\delta_a^A = 1$. Similar relationships hold for $b$. For service negotiation purposes, we only consider the case where $\delta_a^A \le 1$ and $\delta_b^A \le 1$.

In a single-issue negotiation scenario, the preferences of the agents are symmetric, in that a deal that is more preferred from one agent's point of view is guaranteed to be less preferred from the other's point of view, and vice versa. At the beginning of the negotiation, an agent makes an offer that gives it the highest utility and then incrementally concedes by offering its opponent a proposal that gives it lower utility as the negotiation progresses. Because of the symmetric preferences of agents, agents have to concede to offer deals that are more likely to be accepted by their opponents if they prefer reaching an agreement instead of the conflict deal. An outcome is *individual rational* if it gives an agent a utility that is no less than its utility from the conflict outcome. The maximum possible utility that agent $a(b)$ can get from an outcome over issue $A$ is denoted $U_{max,a}^A(U_{max,b}^A)$, which is individual rational to both agents.

Agent $a$'s strategy (denoted $\sigma_a$) is a mapping from the previous negotiation proposals $p_{a,t'<t}$ and $S_a$ to action $Ac_{a,t}$ that it takes at time period $t$:
$\sigma_a : p_{a,t'<t} \times S_a \to Ac_{a,t}$ is defined as:

$$Ac_{a,t} = \begin{cases} \text{Quit} & \text{if } t \ge T_a \\ \text{Accept} & \text{if } U_a^A(p_{b,t}^A, t) \ge U_a^A(p_{a,t+1}^A, t+1) \\ \text{Offer } p_{a,t+1}^A \text{ at t+1,} & \text{otherwise.} \end{cases} \tag{3}$$

where $p_{b,t}^A$ is the offer made by agent $b$ over issue $A$ at time $t$. $p_{a,t+1}^A$ is defined analogously. Let $P_{a,t}^A$ denotes the offer that agent $a$ makes at time $t$ in equilibrium, drawn from agent $a$'s equilibrium strategy. $P_{a,t}^A$ is determined by:

$$P_{a,t}^A = (U^{-1})_a^A((1 - y_{a,t}^A) \times U_{max,A}^A) \tag{4}$$

where $y_{a,t}^A$ is agent $a$'s yield-factor [3] at time $t$.

### 3.2: Multiple-Issue Negotiation

Consider multiple-issue negotiation over issue set $I$ of $k$ issues, where $I = \{I_1, I_2, \ldots, I_k\}$. The agents' parameter sets are defined as follows:

$$\begin{aligned} S_a &= \left\langle P_a^I, U_a^I, T_a, \delta_a \right\rangle \\ S_b &= \left\langle P_b^I, U_b^I, T_b, \delta_b \right\rangle \end{aligned} \tag{5}$$

where $P_a^I = \{P_a^i \mid i \in I\}$ denotes agent $a$'s reserve prices over $I$ and $P_a^i$ denotes $a$'s reserve price over issue $i$, $U_a^I = \{U_a^i \mid i \in I\}$ denotes agent $a$'s utility functions over $I$ and $U_a^i$ denotes $a$'s utility functions over issue $i$, and $T_a$ and $\delta_a$ denote agent $a$'s bargain deadline and discount factor. Agent $b$'s negotiation parameters are defined similarly. We assume that an agent's utility from the issue set $I$ is the sum of its utilities from all issues.

Multiple-issue negotiation is usually described as cooperative negotiation, since it is a non-zero-sum game where, as the values along multiple dimensions shift in different directions, one dimension for each issue, it is possible for all parties to be better off [14]. Two procedures for multiple-issue negotiation have been discussed recently: *package deal* and *issue-by-issue negotiation*. For a package deal, an offer includes a value for each issue under negotiation. Thus, for $k$ issues, an offer is a package of $k$ values. This allows trade-offs to be made between issues. Agents are allowed to either accept a complete offer or reject a complete offer. For issue-by-issue negotiation, each issue is settled separately and an agreement can take place either on a subset of issues or on all of them.

We first describe the procedure for a package deal. Assume that the agents use the same protocol as for single issue negotiation, but instead of making an offer on a single issue, an agent offers a set of offers (an offer consists of a set of values for issues from $I$), all of which give it equal utility. This is because when there is more than one issue, an agent can make trade-offs across issues even if agents' preferences are symmetric over each individual issue.

As an example, Figure 2(a) illustrates the utility for *4*-issue negotiation with two package deals of two issues each. We focus here on the utility frontiers for the issue set $I = \{A, B\}$. In all figures, agents' utilities are measured along two axes; the origin represents the conflict outcome. The segment $AA'$ is the utility frontier for issue $A$ and $BB'$ that for issue $B$. The utility frontier for $I$ is $A''B''C''D''$ (i.e., the sum of all possible utilities from issue $A$ and issue $B$). The points along $LL'$ are pairs of values for issue $A$ and issue $B$ that give equal utility to agent $a$ but different utilities to agent $b$. $L$ is Pareto-optimal since it is the only one, from all possible pairs along $LL'$, that lies on the segment $A''B''C''D''$. Because an agent does not know its opponent's utility function, it does not know which of the possible pairs along $LL'$ is Pareto-optimal. Therefore, agent $a$ makes trade-offs across $A$ and $B$, and then offers a set of pairs that correspond to points along $LL'$.

An agent's preferences over issues can be represented as the slopes of the segments for issues. For example, the slopes of segments $AA'$ and $BB'$ represent how the agents value the issues $A$ and $B$. Agent $a$ is said to value issue $A$ more (less) than $b$ if the increase in $a$'s utility for a unit change for issue $A$ is higher (lower) than the increase in $b$'s utility for a unit change for issue $A$. Therefore, the slope of the segment represents the agents' utility preference for an issue, and is named comparative interest in [3].

At time $t$, Agent $b$ generates a set of offers that give itself equal utility. We define $\bar{P}^I_{b,t} = \left\langle \bar{P}^{I_1}_{b,t}, \bar{P}^{I_2}_{b,t}, \ldots, \bar{P}^{I_k}_{b,t} \right\rangle$ as agent $b$'s current optimal utility offer for agent $a$ if it gives agent $a$ the maximum utility. Therefore, agent $a$'s action $Ac_{a,t}$ for the package deal procedure is defined as follows:

$$Ac_{a,t} = \begin{cases} \text{Quit} & \text{if } t \geq T_a \\ \text{Accept} & \text{if } U^I_a(\bar{P}^I_{b,t}, t) \geq {U^I_{a,t+1}}' \\ \text{Offer } P_{a,t+1}({U^I_{a,t+1}}') \text{ at t+1,} & \text{otherwise.} \end{cases} \tag{6}$$

where ${U^I_{a,t+1}}'$ is the utility value for agent $a$ to generate its counter-offer at time $t+1$. $P_{a,t+1}({U^I_{a,t+1}}')$ is a set of offers, all of which give $a$ equal utility ${U^I_{a,t+1}}'$, from $a$. Agent $a$ is playing its equilibrium strategy if ${U^I_{a,t+1}}' = (1 - y^I_{a,t+1})U^I_{max,a}$, where $U^I_{max,a}$ is the maximum possible utility agent $a$ can get from issue set $I$ [3]. The equilibrium strategy for agent $b$ is defined analogously.

We now turn to the issue-by-issue procedure. Agent $a$ negotiates each issue separately. For each issue, $a$'s action $Ac_{a,t}$ is similar to that in single-issue negotiation.

## 4: Coalition Deal Negotiation

As discussed in [16], a self-interested agent's rationality is bounded by computational complexity in automated negotiation. In most practical situations, such as real time service binding and contracting, a contracting agent is bounded rational because its computation resource is costly, the environment is dynamic, and there is time pressure for missing a negotiation deadline and a better offer. With all these limitations, if too much time is spent in computing a new proposal, another agent may win the contract before the proposal is sent. If too little time is spent, the agent may make a suboptimal proposal.

The outcome of negotiation depends on different negotiation strategies and procedures. Issue-by-issue negotiation is a sequence of single issue negotiations, in which agents' preferences are symmetric and no trade-off is allowed to be better off. For issue-by-issue negotiation in our *GetStockQuote* example, agents agree on the issue of payment method with pay for bundle and they also reach agreement that $p$ is the inquiry cost. Since agents negotiate these issues independently, it is possible that $p$ is too high to $a$ if $a$ chooses to pay for the bundle as its payment method. That means issue-by-issue negotiation may degrade the agents' utilities. In package deal negotiation, agents may combine different payment methods with different inquiry costs to reach mutually beneficial agreement when they cross over the six issues. However, the package deal also leads to an exponential growth in the computation cost to generate the offer sets. Most services, of course, are more complex than our example, and when they are composed this computation problem is significant. To make negotiation for services both optimum and efficient, we introduce the coalition deal.

### 4.1: Definition and Negotiation Model

We define coalition deal negotiation, which makes a better trade-off between issue-by-issue negotiation and the package deal procedure, to provide an agent the flexibility to balance its time and utility. Moreover, coalition deal provides agents approximately optimized utilities with minimized computation costs.

**Definition 1.** *For a coalition deal, all negotiation issues are partitioned into disjoint partitions and each partition is negotiated independently of other partitions. Like the package deal, issues inside the same partition are negotiated as a whole package and an offer includes a value for each issue in this partition.*

From this definition, we can see that issue-by-issue negotiation is a specific case of a coalition deal with one issue per partition. The package deal is also a coalition deal, where there is only one partition for all issues. Coalition deal negotiation provides (a) better utility than issue-by-issue negotiation, (b) less computational cost than package deal negotiation, (c) more flexible negotiation (details below), and (d) better management for service negotiation.

Consider multiple-issue negotiation with issue set $I$ of $k$ issues, where $I = \{I_1, I_2, \ldots, I_k\}$. From the definition, we know that there exists a partition $IP$ of size $s$ over $I$, where $IP = \{IP_j \mid 1 \le j \le s\}$. $IP$ satisfies the constraint: $\forall 1 \le m \le s, 1 \le n \le s, m \ne n$, we have $IP_m \cap IP_n = \emptyset$ and $\cup_{j \in IP} \cup_{i \in j} = I$. Similarly, agents' parameter sets can be defined as follows:

$$S_a = \left\langle P_a^{IP}, U_a^{IP}, T_a, \delta_a \right\rangle$$

$$S_b = \left\langle P_b^{IP}, U_b^{IP}, T_b, \delta_b \right\rangle \quad (7)$$

where $P_a^{IP} = \{p_a^i \mid i \in j, j \in IP\}$ denotes agent $a$'s reserve prices set over partitions of issue set $I$ and $p_a^i$ denotes $a$'s reserve price over issue $i$, which belongs to partition $j$, $U_a^{IP} = \{U_a^i \mid i \in IP\}$ denotes agent $a$'s utility functions over partition $IP$ where $U_a^i$ denotes agent $a$'s utility function over one partition $i$ from $IP$, and $T_a$ and $\delta_a$ denote agent $a$'s bargaining deadline and discount factor. Agent $b$'s negotiation parameters are defined analogously. An agent's utility from partition $IP$ of the issue set $I$ is the sum of its utilities from all partitions. For a coalition deal, each partition is negotiated separately and independently of other partitions. An agreement can take place either on some of the partitions or all of them. For each partition, an offer includes a value for each issue inside this partition that would be the same as the package deal for this partition. An agreement has to take place either on all the issues inside the partition or none of them. For each partition, we assume that the agents use the same protocol as for the package deal, but instead of making a set of offers over issue set $I$, an agent offers a set of offers over issues from this partition. An agent can make trade-offs only across issues in the same partition, resulting in a set of offer sets, all of which give it equal utility.

Figure 2(a) illustrates the utility frontiers for issue set $I$ where $I = \{A, B, C, D\}$. There exists a partition $IP$ for $I$ where $IP = \{\{A, B\}, \{C, D\}\}$. Let $IP_1 = \{A, B\}$, and $IP_2 = \{C, D\}$. The utility frontier for $IP_1$ is $A''B''C''D''$ and the utility frontier for $IP_2$ is $S''T''V''U''$. For $IP_1$, the points along $LL'$ are pairs of values for $IP_1$ that give equal utilities to agent $a$ but different utilities to agent $b$. The points along $RR''$ are pairs of values for $IP_1$ that give equal utilities to agent $b$ but different utilities to agent $a$. The utility for $IP$ is the sum of the utilities from $IP_1$ and $IP_2$ after these partitions are negotiated independently. If we only consider the optimal outcome from both negotiations over $IP_1$ and $IP_2$, All optimal outcomes for $IP_1$ lie on the segment $MB''K$, and all optimal outcomes for $IP_2$ lie on the segment $XT''Y$ as we described for the package deal. Therefore, the possible utility frontier for $IP$ is represented by region $OM''P''QQ'P$ in Figure 2(b). At time $t$, agent $b$ generates a set of offers over a partition $IP_i$ of $k_i$ issues that give itself the equal utility. We define $\bar{P}_{b,t}^{IP_i} = \left\langle \bar{P}_{b,t}^{IP_i(1)}, \ldots, \bar{P}_{b,t}^{IP_i(k)} \right\rangle$ as agent $b$'s current optimal utility offer for agent $a$ if it gives $a$ the maximum utility. Therefore, agent $a$'s action $Ac_{a,t}$ for the coalition deal procedure is:

$$Ac_{a,t} = \begin{cases} \text{Quit} & \text{if } t \geq T_a \\ \text{Accept package deal for } IP_i & \text{if } U_a^{IP_i}(\bar{P}_{b,t}^{IP_i}) \geq {U_{a,t+1}^{IP_i}}' \\ \text{Offer } P_{a,t+1}({U_{a,t+1}^{IP_i}}') \text{ for } IP_i \text{ at t+1,} & \text{otherwise.} \end{cases} \quad (8)$$

where ${U_{a,t+1}^{IP_i}}'$ is the utility value for agent $a$ to generate its counter-offer at time $t+1$ over partition $IP_i$. $P_{a,t+1}(U_{a,t+1}^{IP_i})'$ is a set of offers, each of which give $a$ the utility ${U_{a,t+1}^{IP_i}}'$, from $a$. Similarly, we define agents are playing their equilibrium strategy for the package deal over each partition.

### 4.2: Coalition Deal Utility

Previous sections describe three different negotiation procedures: issue-by-issue, package deal, and coalition deal. These three procedures can generate different outcomes, and consequently give different utilities to the agents. To decide the optimal procedure that gives

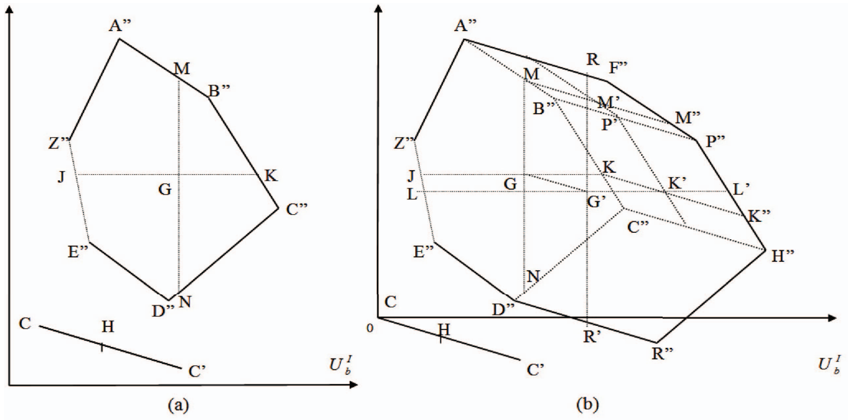


**Figure 1. Agents' utilities for n-issue negotiation**

the agents their highest utilities, we need to compare their utilities from these procedures for n-issue negotiation. An agent's utility frontier lies in quadrant *Q1* if it has a zone of agreement where both agents prefer agreement over no deal [3]. We next discuss the scenario in which both agents are individually rational (i.e., all issues have a zone of agreement ensured by the service description and the discovery procedure).

**Lemma 1.** *Each agent's utility from the package deal is no worse than its utility from issue-by-issue negotiation for two-issue negotiation.*

Lemma 1 has been proven in [3]. We give a brief description here. In Figure 2(a), the two segments $AA'$ and $BB'$ denote agent $a$'s and agent $b$'s utilities from issues $A$ and $B$ respectively. The agents' combined utilities from the two issues lie in the region $A''B''C''D''$. Points $E$ and $F$ denote the equilibrium outcomes if each issue is negotiated using the single issue protocol. Then the point $G$ shows each agent's cumulative utility from both issues in issue-by-issue negotiation. In a package deal, agent $a$ makes offers at time $t$ that give it the same utility. An agreement for a package deal can occur anywhere along segment $MB''K$. Since all points on $MB''K$ dominate $G$ for both agents, the package deal gives each agent a utility that is no worse than its utility from issue-by-issue negotiation. In the case where the agents have the same comparative interests over issues, the agents' combined utilities from the two issues form a segment, instead of a region. Each agent always offers a single pair along the segment; the package deal therefore gives each agent a utility that is the same as it receives from issue-by-issue negotiation.

In a service-oriented environment, there are many issues concerning functionality and quality that need to be negotiated during service engagement. Can we generalize Lemma 1 to cover more than two? Here, we compare agents' utilities from a package deal and issue-by-issue negotiation for $n$-issue negotiation.

**Theorem 1.** *Each agent's utility from the package deal is no worse than its utility from issue-by-issue negotiation for n-issue negotiation, where* $n > 2$.

*Proof.* We can prove this by induction.

(1) Base case: For $n$-issue negotiation where $n \leq 2$, it is proved by Lemma 1.

(2) We assume Theorem 1 holds for $k$-issue negotiation where $k > 3$, then consider $(k+1)$-issue negotiation. In Figure 1, we assume region $A''B''C''D''E''Z''$ represents the combined utility from a package deal of $k$ issues. $CC'$ denotes the utility for agents $a$ and $b$ over the $(k+1)$-th issue and $H$ denotes the equilibrium outcome if this issue is negotiated using the

single-issue protocol. The point $G$ shows each agent's cumulative utility from $k$ issues for issue-by-issue negotiation. All points on $MB''K$ represent Pareto-optimal utilities over $G$ from package deal of $k$ issues. For issue-by-issue negotiation, point $G'$ shows each agent's cumulative utility from $k+1$ issues. Since points on $MB''K$ are the optimum utility from a package deal of $k$ issues, segment $M'P'K'$ is the cumulative utility from $k$-by-1 negotiation. Similarly, the agents' combined utilities from a package deal of $k+1$ issues lie in the region $A''F''P''H''R''D''E''Z''D''$. Since Theorem 1 holds for $k$-issue negotiation, all points on segment $MB''K$ dominate $G$. We know that all points on Segment $RF''P''L'$ dominate those on segment $M'P'K'$ and all points on segment $M'P'K'$ dominates $G'$ as long as all points on segment $MB''K$ dominate $G$. Then, a package deal of $k+1$ issues gives agents utilities no worse than their utilities from k-by-1 negotiation. k-by-1 negotiation gives agents utilities no worse than those from issue-by-issue negotiation. Therefore, a package deal of $k+1$ issues gives agents utilities no worse than their utilities from issue-by-issue negotiation. Theorem 1 thus is proved by induction.

From Theorem 1, we know that a package deal gives agents utilities better than issue-by-issue negotiation does. As stated in the previous section, a coalition deal provides approximately optimized utilities to agents. Here, we prove that a coalition deal give agents utilities better than issue-by-issue negotiation does.

**Theorem 2.** *Each agent's utility from a coalition deal is no worse than its utility from issue-by-issue negotiation for n-issue negotiation, where $n > 2$.*

*Proof.* This can be proved by Theorem 1 and our assumption of additive utilities.
Given an issue set $I = \{I_1, I_2, \ldots, I_k\}$ and a partition $IP = \{IP_1, IP_2, \ldots, IP_s\}$ over $I$, Agent $a$'s utility from a coalition deal is denoted as $U_a^{IP}$, where $U_a^{IP} = \sum_{1 \leq j \leq s} U_a^{IP_j}$ since each agent's utility from a coalition deal is the sum of the utility the agent earns from each partition, Agent $a$'s utility from issue-by-issue negotiation is $U_a^I$, where $U_a^I = \sum_{i \in I} U_a^i$ since each agent's utility from issue-by-issue negotiation is the sum of the utility the agent earns from each issue. For issue-by-issue negotiation, we can sort all issues by partitions and denote $U_a^I = \sum_{IP_j \in IP} \sum_{i \in IP_j} U_a^i$. We know that issues in the same partition are negotiated as a whole package and agents' utilities from package deal are no worse than their utilities from issue-by-issue negotiation. Since there is at least one partition with multiple issues, which gives the agent a utility no worse than the utility from issue-by-issue negotiation over issues inside this partition. It means $U_a^{IP_j} \geq \sum_{i \in IP_j} U_a^i$ over the issues in partition $IP_j$. We have $U_a^{IP_j} \geq \sum_{i \in IP_j} U_a^i$ for every $IP_j \in IP$, then $U_a^{IP} \geq U_a^I$ . Therefore, the utilities agents earn from a coalition deal are no worse than those from issue-by-issue negotiation.

Both the package deal and coalition deal give agents utilities better than issue-by-issue negotiation does. Which procedure, package deal or coalition deal, gives agents better utilities? To answer this question, we first prove that the package deal gives agents utilities better than a coalition deal of two partitions.

**Lemma 2.** *Each agent's utility from the package deal is no worse than its utility from i-by-j negotiation for n-issue negotiation, where $i \geq 1, j \geq 1, n > 2$, and $i + j = n$.*

*Proof.* The $i$-by-$j$ negotiation is a coalition deal of two partitions. We proved that the agents' utilities from a package deal of $n$-issue negotiation are better than their utilities from $(n-1)$-by-1 negotiation in Theorem 1. In the case where $i \neq 1$ and $j \neq 1$, we can prove Lemma 2 by induction.

(1) Base case: For $n \leq 3$, it has been proved by Theorem 1.

(2) We assume Lemma 2 holds for $k$-issue negotiation where $k > 3$ and then consider

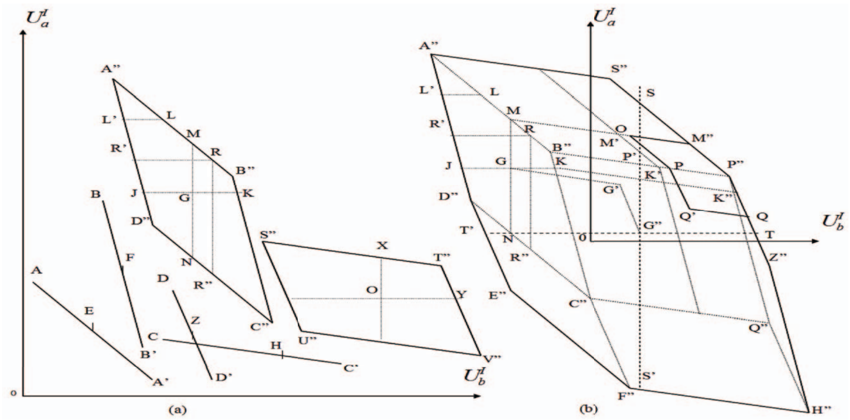


**Figure 2. Agents' utilities for 4-issue negotiation**

$(k+1)$-issue negotiation. Let the region $R_k$ represent the utility from a package deal over $k$ issues and the region $R_{i,j}$ represent the utility from $i$-by-$j$ negotiation that is the sum of the utilities from a package deal over $i$ issues (denoted $R_i$) and a package deal over $j$ issues (denoted $R_j$). Since Lemma 2 holds for $k$-issue negotiation, there exists a utility frontier $R_{k,-i,-j}$ that represents points in the region $R_k$ that dominates all points in the region $R_{i,j}$. For $(k+1)$-issue negotiation, let segment $CC'$ denote utility for agents $a$ and $b$ over the $(k+1)$-th issue $C$, let point $H$ denote the equilibrium outcome if this issue is negotiated using single-issue negotiation. Without losing generality, we randomly combine issue $C$ with the set with $j$ issues to form a package deal over $j+1$ issues. Therefore, we let region $R_{j+1}$ represent agents' utilities from the package deal of $j+1$ issues, the sum of the utility from region $R_j$ and segment $CC'$. Region $R_{i,j+1}$ represents the sum of the utility from region $R_i$ and region $R_{j+1}$. We define a region $R_{i,j+1}{}'$ to represent the sum of the utility from region $R_{i,j}$ and segment $CC'$. Based on our assumption of additive utilities, $R_{i,j+1}$ and $R_{i,j+1}{}'$ are equal because real numbers are associative under addition. Therefore, $R_{i,j+1}$ represents the sum of the utility from region $R_{i,j}$ and segment $CC'$. Let region $R_{k+1}$ represent the utility from the package deal over $k+1$ issues. We know that $R_{k+1}$ also represents the sum of the utility from region $R_k$ and segment $CC'$. From the above, we can say that there is a utility frontier $R_{k+1,-i,-(j+1)}$ representing some points in region $R_{k+1}$ that dominate all points in the region $R_{i,j+1}$, since real numbers are also transitive under addition. In another words, agents' utilities from package deal of $k+1$ issues are no worse than their utilities from $i$-by-$(j+1)$ negotiation.

It can be illustrated by examples. Figure 2 shows the scenario where each agent's utility from a package deal of four issues is better than its utility from a 2-by-2 negotiation. We have $R_i = MB''K$ and $R_j = XT''Y$ in Figure 2(a) and $R_{i,j} = OM''P''QQ'P$, $R_k = SP''T$ and $R_{k,-i,-j} = M''P''Q$ in Figure 2(b).

We have proven that the package deal gives agents utilities better than a coalition deal of two partitions. For $n$-issue negotiation, we need to extend Lemma 2 to the coalition deal with more than two partitions.

**Theorem 3.** *Each agent's utility from a coalition deal is no better than its utility from the package deal for n-issue negotiation, where $n > 2$.*

*Proof.* Lemma 2 tells us that the package deal gives agents better utilities than a coalition deal of two partitions does. Given an issue set $I = \{I_1, I_2, \ldots, I_k\}$ and a partition

$IP = \{IP_1, IP_2, \ldots, IP_s\}$ over I, we divide $I$ into two partitions: $IP_1$ and $IP - IP_1$, let $U_a^I$ denote agent $a$'s utility from package deal and $U_a^{IP}$ denote $a$'s utility from two-partition coalition deal. From Lemma 2, we know that $U_a^I \geq U_a^{IP}$ where $I' = I - IP_1$ and $IP' = \{IP_2, \ldots, IP_s\}$, we denote $a$'s utility from the package deal as $U_a^{I-IP_1}$ and denote $a$'s utility from a two-partition coalition deal as $U_a^{IP-IP_1} = U_a^{IP_2} + U_a^{IP-IP_1-IP_2}$. We know that $U_a^{I-IP_1} \geq U_a^{IP_2} + U_a^{IP-IP_1-IP_2}$. Keeping the partitioning, we have $U_a^I \geq \sum_{i=1}^{r<s}(U_a^{IP_i} + U_a^{I-(IP_1+\ldots+IP_r)}) \geq \sum_{i=1}^{s} U_a^{IP_i}$. Since the transitive property of addition holds during partitioning, each agent's utility from a coalition deal is no better than that from the package deal for $n$-issue negotiation.

### 4.3: Coalition Deal Efficiency

From Theorem 1, 2, and 3, we know that each agent's utility from the package deal is better than its utility from a coalition deal and issue-by-issue negotiation. Therefore, we should choose the package deal to maximize agents' utilities. However, we need to consider the computational costs, which can be the primary factor in the competitive circumstance of QoS-aware service negotiation.

Given an issue set $I = \{I_1, \ldots, I_n\}$ and a partition $IP = \{IP_1, \ldots, IP_k\}$ over $I$, we define the unit computational cost for generating a price for one issue as a constant. We assume that every issue in issue-by-issue negotiation can be negotiated in parallel and every partition in a coalition deal can also be negotiated in parallel. To compare the computational efficiency, we need to compare the computational cost of generating an offer in each round of three different procedures.

An $n$-issue negotiation can be viewed as a distributed search through an $n$-dimensional space, where each issue has a separate dimension associated with it. In issue-by-issue negotiation, each issue is negotiated separately. Based on the above equilibrium strategy, agents will compute a value for each issue. Therefore, the computational cost in one round is $O(n)$, where $n$ is the size of the issue set. In the package deal, an offer is a set including a value for each issue under negotiation. In each round, an agent can make trade-offs across all n issues to offer a set of offers that give it the same utilities. In the worst case, the computational cost in one round is $O(m^n)$, where we assume each issue may have $m$ possible values.

The computation problem of generating an offer set is equivalent to searching in an $n$-dimensional space for all combinations of possible distributions of given utility value among all $n$ issues with a utility constraint. This problem is intractable in general and takes $O(m^n)$ time in the worst case. Even worse, we have to solve this problem every round during the package deal negotiation procedure. It means that it will be infeasible for an agent to consider every possible offer given a utility constraint. In coalition deal negotiation, issues are partitioned into $k$ disjoint partitions and each partition is settled independently of the other partitions. Therefore, the computation problem is reduced to the sum of $k$ searches where the $i$-th search is in an $n_i$-dimension space, where $n_i << n$ and $\sum_{i=1}^{k} n_i = n$. This problem takes $O(km^{n_s})$ time in the worst case, where $n_s = \textbf{argmax}\ n_i$. Moreover, we can limit the maximum size of a partition to a constant $C$. Therefore, the computational cost of a coalition deal reduces to $O(nm^C)$. The time complexity will be $O(m^C)$ if we have several agents, one for each partition, work together to generate a coalition deal.

In our *GetStockQuote* service scenario, we divide six issues into two partitions. The computational cost is *6* in each round for issue-by-issue negotiation. In package deal, agents

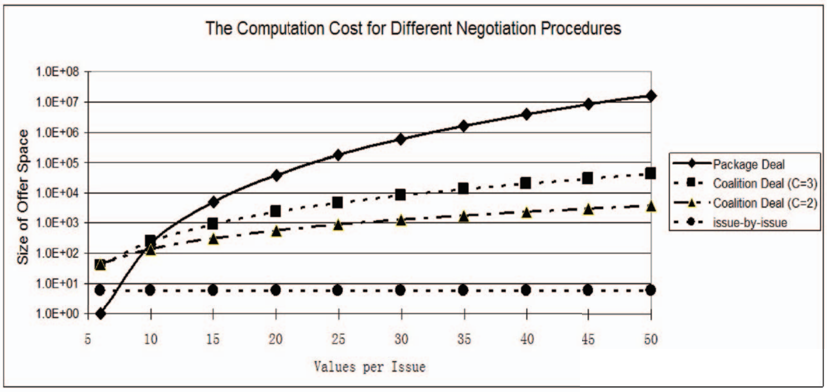


**Figure 3. The worst case computation costs for different procedures**

need to search through all possible offers in a 6-dimensional space to meet the given utility constraint. The computational cost is $O(a^6)$ in the worst case, where $a$ is the size of possible value per issue. In a coalition deal, the computational cost is $O(a^3)$ in the worst case. With our assumption of *individual rationality*, the upper bound is reduced to $O(\binom{m}{n})$, where $\binom{m}{n} = \frac{m!}{n!(m-n)!}$ and $m! = \sqrt{2\pi m}\left(\frac{m}{e}\right)^m\left(1+\theta\left(\frac{1}{m}\right)\right)$. For different negotiation procedures, the worst case computation costs in our example with the increase of value per issue are compared in Figure 3.

How is a partition over the negotiation issues found? There is a concern that the computation efficiency will degrade if finding a partition is computationally expensive. To compute an optimum partition is unnecessary since it may be computationally expensive and not always optimal for all the opponents. However, there are still some guidelines for partitioning the issue set:

- If agent's utility from issues is additive, the issue set could be partitioned arbitrarily and independently, and an agent can choose different partition sizes according to its current computation capacity and time pressure.
- In the situation where some issues could be dependent and their utilities are in the form of one multi-variable function with one variable for each issue, we can put these issues into the same partitions and assume a more general form of additive utility in which each addend is a multi-variable function.
- From Figure 2, we know that a better utility frontier can be formed by putting issues with distinct comparative interests into the same partition. To get a better position in real-world negotiation, people always bind their most interested issues with their opponents' most interested issues. If the opponent's preference is unknown, people tend to put their most interested issues together with the issues that they are most likely to compromise.

### 4.4: Coalition Deal Negotiation for Services

The approach of coalition deal negotiation can be used in different negotiations in general. It is more suitable for service-oriented negotiation in a competitive environment with self-interested agents, especially in the situation when a complex service is involved. Moreover, it is easier for the service agents to partition the issue set with the guidelines from service profiles or descriptions.

Furthermore, the coalition deal is more flexible in most automated negotiations, since most negotiations have time constraints and the coalition deal allows agents to balance their computation costs and utility gains:

- An agent can adjust its partition size based on its current time pressure and computation resource.
- Most distributed automated negotiation systems encounter a message congestion problem and the coalition deal can mitigate it. A message congestion problem occurs when an agent cannot process its received messages as fast as they arrive.
- If new issues are involved, by adding new partitions for new issues, the coalition deal provides better scalability without affecting the existing issues.

Another advantage of the coalition deal is that it is natural to partition issues into different categories and deal with each category separately. For example, in bilateral negotiation of a labor dispute, it would be easier if money issues such as salary and bonus are negotiated in a partition separately from issues such as working condition and healthcare. Of course, it is possible that both sides would benefit if they could deal with all issues as a package, but the negotiation might become infeasible.

In QoS-aware service contracting, self-interested service agents negotiate with each other over multiple issues to reach an agreement while maximizing their utilities. With the coalition deal, agents can reveal their deadlines; honesty about their real deadline is enforced by the negotiation protocol. For example, the agent that has the latest deadline will receive better payoff at the time right before its deadline. After partitioning the issue set, agents choose either the Boulware or conceder discount functions [11, 12] based on their time concerns. Then they compute the expected cumulative utility and generate a set of offers by crossing over multiple issues inside one partition. Since all partitions can be negotiated in parallel and independently, a service agent can breed a negotiation thread for each partition. The thread agents collaborate to reach a service agreement.

## 5: Conclusions and Future Work

This paper investigates the optimal strategy for QoS-aware negotiation for services by proposing the coalition deal to optimize agents' utilities while reducing computational cost. Using equilibrium strategies for the package deal and issue-by-issue negotiation, we compare the agents' utilities and computational cost for the coalition deal with those from the package deal and issue-by-issue negotiation. Finally, we prove that the coalition deal makes better tradeoffs between utility optimization and computational efficiency.

There are several possible directions for future work. First, the partitioning problem can be investigated further and the coalition deal generalized to allow agents to make tradeoffs among different partitions. Second, the opponent's interest can be learned from the package deal and exploited. Third, concurrent coalition deal negotiation may be coordinated to speed up the negotiation and further improve agents' utilities.

## References


[1] Semantic web services architecture requirements. Semantic Web Services Initiative Architecture committee (SWSA). http://www.daml.org/services/swsa/swsa-requirements.html.

[2] Jiangbo Dang, Devendra Shrotri, and Michael N. Huhns. Distributed coordination of an agent society based on obligations and commitments to negotiated agreements. In Paul Scerri, editor, *Challenges in the Coordination of Large-Scale Multiagent Systems*. Springer Verlag, 2005.

[3] S. S. Fatima, M. Wooldridge, and N. Jennings. Optimal negotiation of multiple issues in incomplete information settings. In *Proc. Third International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS'04)*, pages 1080–1089, New York,USA, 2004. ACM.

[4] C. M. Jonker and V. Robu. Automated multi-attribute negotiation with efficient use of incomplete preference information. In *Proc. Third International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS'04)*, pages 1056–1063, New York,USA, 2004. ACM.

[5] Rajiv T. Maheswaran and Tamer Basar. Coalition formation in proportionally fair divisible auctions. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 25–32, New York, NY, USA, 2003. ACM Press.

[6] E. M. Maximilien and M. P. Singh. A framework and ontology for dynamic web services selection. *IEEE Internet Computing*, 8(5):84–93, 2004.

[7] T. D. Nguyen and N. Jennings. Coordinating multiple concurrent negotiations. In *Proc. Third International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS'04)*, pages 1064–1071, New York,USA, 2004. ACM.

[8] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. the MIT Press, 1994.

[9] G. Petrone. Managing flexible interaction with web services. In *Proc. Workshop on Web Services and Agent-based Engineering (WSABE 2003)*, pages 41–47, Melbourne, Australia, 2003.

[10] C. Preist. A conceptual architecture for semantic web services. In *Proceedings of the Third International Semantic Web Conference 2004 (ISWC2004)*, Hiroshima,Japan, 2004.

[11] D. G. Pruitt. *Negotiation Behaviour*. Academic Press, New York, USA, 1981.

[12] H. Raiffa. *The Art and Science of Negotiation*. Harvard University Press, Cambridge, MA, USA, 1982.

[13] S. Ran. A model for web services discovery with qos. *ACM SIGecom Exchanges*, 4(1):1–10, 2003.

[14] J. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. the MIT Press, 1994.

[15] T. Sandholm and N. Vulkan. Bargaining with deadlines. In *Proc. the National Conference on Artificial Intelligence (AAAI)*, pages 44–51, Orlando, FL, 1999.

[16] Tuomas Sandholm and Victor Lesser. Issues in automated negotiation and electronic commerce: Extending the contract net framework. In Victor Lesser, editor, *Proc. of the First International Conference on Multi-Agent Systems (ICMAS'95)*, pages 328–335, San Francisco, CA, USA, 1995. The MIT Press: Cambridge, MA, USA.

[17] Carles Sierra, Peyman Faratin, and Nicholas R. Jennings. A service-oriented negotiation model between autonomous agents. In *Proceedings of the 8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pages 17–35. Springer-Verlag, 1997.