COORDINATING ONE-TO-MANY CONCURRENT NEGOTIATION FOR SERVICE PROVISION

Completed Research Paper

Jing Zhao
Department of Information Systems
City University of Hong Kong
Hong Kong
Jingzhao-c@my.cityu.edu.hk

Sherry X. Sun
Department of Information Systems
City University of Hong Kong
Hong Kong
Sherry.sun@cityu.edu.hk

Sumit Sarkar
Naveen Jindal School of Management
University of Texas at Dallas
Richardson, TX, United States
sumit@utdallas.edu

Abstract

In Service-Oriented Architecture (SOA), a consumer may require a service which can be offered by several providers with the same functionality but different Quality of Services (QoS), such as price, response time, reliability, and data quality. The consumer can negotiate with all of these providers simultaneously to reach an agreement on the QoS with one provider – this is referred to as one-to-many concurrent negotiation model. Most existing studies on one-to-many concurrent negotiation adopt the concession function from one-to-one bilateral negotiation, which determines the utility level the consumer is going to concede to for each provider individually. This makes it impossible to adjust the concessions based on the negotiation with other opponents in the context of one-to-many negotiation. In our work, we propose a novel concession strategy that coordinates the multiple concurrent negotiation threads. Simulation experiments are conducted to demonstrate the effectiveness of our proposed approach.

Keywords: Negotiation, service oriented architecture, coordination, service provision


Introduction

Service-Oriented Architecture (SOA) has recently become an important framework for deploying and developing new business applications (Zhao et al. 2007). In a SOA environment, software components can be packaged as self-contained and self-describing Web services (Zhao et al. 2008). Once a Web service is deployed, other applications can search for, discover, and bind that service through the Web services standards, such as SOAP, WSDL, and UDDI (Curbera et al. 2002). In SOA, a consumer may require a service which can be offered by several providers with the same functionality but different Quality of Services (QoS), such as price, response time, reliability, data quality, etc (Menasce 2002). Different consumers may have diverse preferences on QoS; for example, when requesting multi-media content from a service provider, a price-sensitive consumer may care more about the cost of service while another consumer may care more about the delay and jitter of the delivery of multi-media. Therefore, service providers can benefit by configuring their services appropriately to fulfill the needs of different consumers with the customized QoS specified in a service contract. Such contracts can be described using Web services standards, such as Web Service Level Agreement (WSLA) (Ludwig et al. 2003).

In an open environment, consumers and providers can enter and leave the service market freely, resulting in constantly changing supply and demand. In such a dynamic environment, it may not be desirable to fix all the QoS parameters; instead negotiation has become a natural and promising approach for the consumers and providers to dynamically agree on the QoS at the time of a request (Jennings et al. 2001; Elfatatry and Layzell 2004). When there are many providers available, the consumer can negotiate simultaneously with multiple providers — this is referred to as a one-to-many concurrent negotiation model (Nguyen and Jennings 2003). This model can be considered as several one-to-one individual negotiations carried out concurrently with each individual negotiation thread involving one consumer and one provider (Rahwan et al. 2002; Nguyen and Jennings 2003).

A significant issue in this model is how to coordinate all the negotiation threads to generate offers by making concessions. In existing studies on one-to-many negotiation, the coordination is limited. For example, in the works presented by Nguyen and Jennings (2005) and Sim and Shi (2010), the consumer adopts time-dependent functions to make concessions for each thread independently and there is no coordination until an offer is accepted. If an offer is accepted by one provider, the consumer updates the reservation value in other threads and continues to negotiate with other providers to seek better offers. If a better offer comes, the consumer can breach the agreed contract and pay a penalty. Given that the consumer may adopt different concession strategies in response to different providers’ concession strategies; it may result in that the consumer concedes too much in one thread to seek an agreement while there is still a good chance to obtain a higher utility in another thread. Therefore, it would be better if the consumer can avoid such intermediate contracts and potentially costly breaches as much as possible by considering the information from all negotiation threads. In this paper, we study how to coordinate all the negotiation threads to avoid extra concessions, which can result in a higher utility for the consumer.

In one-to-many concurrent negotiations, a consumer generates an offer for each negotiation thread. In our method, an offer is evaluated by considering not only the utility of this offer to the consumer but also how favorable this offer may be to the provider, which is accomplished using a concept of preferability between this offer and the provider’s last offer. We use a composite measure, satisfaction degree, which incorporates both the utility and preferability associated with an offer and enables a trade-off across them. Given that the negotiation is fundamentally time-dependent (Faratin et al. 1998; Sim and Choi 2003), the satisfaction degree is also specified to be time-dependent. Based on the concept of satisfaction degree, we have formally defined the efficiency of offers and used it to check whether efficient concessions are made in one-to-many concurrent negotiation.

In our work, for each negotiation thread, the consumer generates an offer which maximizes the consumer’s satisfaction degree. Thus, we model the concession making process in one-to-many concurrent negotiation as a multi-objective optimization problem, with the consumer's satisfaction degree on each offer being the objective. Further, by specifying an efficiency requirement constraint, we make sure that no unnecessary concession can be made in one thread while a strictly superior offer (with both higher utility and satisfaction degree) is feasible in another thread. We show the concavity of the objective functions, which enables us to obtain the optimal values in an elegant manner. The proposed approach is
easily extended to ensure all the offers are efficient.

To validate our method, we have conducted simulation experiments that compare our approach with two benchmark methods, both making concessions individually. In the first benchmark, the consumer opts out of other negotiation threads once an offer is agreed upon with one provider. In the second one, if an agreement is reached, the consumer continues to negotiate with ongoing providers to seek better offers and is allowed to breach a contract without paying any penalties. The experiments show that our coordination approach achieves a considerably better utility than the first benchmark, with only a slight increase in negotiation time. Our approach achieves nearly the same utility as the second benchmark with less negotiation time even though that the second benchmark is biased in favor of the consumer, given that the consumer can breach a contract without paying any penalty.

The novel contribution of this paper is a new concession method which can coordinate all the negotiation threads to avoid unnecessary concession when a better offer is available in one-to-many concurrent negotiation. The contribution includes the following: 1) we have formally defined efficient concessions and derived the equivalent constraints to guarantee efficient concessions; 2) we have formulated the problem of making efficient concession as a multi-objective optimization problem; 3) we have proposed an efficient algorithm to solve this problem.

Literature Review

The research work on bilateral negotiations has been investigated extensively in game theory and artificial intelligence literature (Raiffa 1982; Rubinstein 1982; Faratin et al. 2002). In game theory based negotiation models, researchers consider negotiation as a bilateral bargaining game and analyze the equilibrium solutions under complete and incomplete information (Chatterjee and Samuelson 1983; Rubinstein 1985; Fatima et al. 2004; Gatti et al. 2008). However, it is difficult to identify equilibrium solutions in more general negotiation contexts, where such information (e.g. negotiation deadline, reservation value) is not available using traditional game-theoretic models. Thus, computational solutions to negotiation problems have received substantial attention in the area of artificial intelligence and multi-agent systems, which focus on how to make automated negotiation technologically possible (Faratin et al. 1998; Jennings et al. 2001; Faratin et al. 2002).

In a bilateral negotiation model, various communication protocols and mechanisms have been designed to automate the negotiation process; for instance, time-dependent, resource-dependent, and behavior-dependent tactics have been developed for making concessions in the negotiation process (Faratin et al. 1998; Jennings et al. 2001). To understand the opponent’s negotiation behavior and improve negotiation outcomes based on such understanding, several prediction techniques, including Bayesian learning, fuzzy rules inference, and other heuristics, have been applied to estimate the opponent’s preference, reservation values, and concession tactics (Zeng and Sycara 1998; Mok and Sundarraj 2005; Cheng et al. 2006; Chari and Agrawal 2007; Buffett and Spencer 2007; Lau et al. 2008; Sim et al. 2009).

When there are multiple providers available, a consumer can choose reverse auction or one-to-many negotiation. In reverse auctions, the consumer lacks the ability to perform two-way communication of offers and counter offers, which is especially important in multi-attribute contracts because it allows the consumer to indicate the areas of the search space where a possible agreement may lie (Rahwan et al. 2002; Nguyen and Jennings 2005). Moreover, in auctions, no agreement can be reached before the deadline while in one-to-many negotiation, the consumer can reach an agreement before the deadline and benefit from the agreed deal more quickly (Nguyen and Jennings 2005).

In one-to-many negotiation model, a consumer can negotiate with the multiple providers sequentially or concurrently (Nguyen and Jennings 2003). In the sequential negotiation model, the consumer negotiates with one provider at a time and uses the negotiation outcome to dictate behaviors in subsequent negotiations; thus, different negotiation sequences may result in different negotiation outcomes. It is a challenge to determine a negotiation sequence given that the consumer may not know the information about the providers before negotiating with them. Although de Fontenay and Gans (2004) have shown that the negotiation sequence does not affect the outcome in specific situations, our model fundamentally differs from their work. In (de Fontenay and Gans 2004), bilateral negotiation is modeled as a game with the assumption the distribution of certain parameters is public information. In our method, we assume parameters such as reservation values, negotiation deadlines, the concession functions, and their
Due to the limitations of the sequential model, several researchers study the one-to-many concurrent negotiation model. Rahwan et al. (2002) propose a framework for one-to-many negotiations through conducting a number of concurrent one-to-one negotiations; however, no experiments are conducted to evaluate the proposed approach. Nguyen and Jennings (2004) use a heuristic to classify an opponent’s concession strategy, and then select the best concession strategy for each opponent accordingly. Empirical evaluation is conducted to validate the performance of the concurrent negotiation model. However, their model is biased in favor of the buyer because the intermediate deals are assumed to be binding on sellers, but not on the buyer.

In order to make the one-to-many negotiation model applicable in realistic situations, a few approaches have incorporated de-commitment mechanism into the model, where an agent can renege on an intermediate commitment by paying a penalty to the opponent (Nguyen and Jennings 2005; Sim and Shi 2010). When it comes to a buyer’s concession strategy for the negotiation threads, Nguyen and Jennings (2005) select a strategy (conceder or non-conceder) for each thread which generates the highest expected payoff for the buyer when applying this strategy. Sim and Shi (2010) apply linear, conciliatory, and conservative time-dependent concession strategies for the consumer to make concessions. However, in both these approaches, the concession strategy for each negotiation thread is applied independently, without considering the information from other threads. As a result, this concession mechanism is inefficient, given that the consumer may concede too much in one thread to reach an agreement while there is a good chance to obtain a better offer in another thread. Even if the buyer can breach an intermediate contract when a better offer is available, it would be better if the buyer does not agree on a less favorable intermediate contract in the first place since a penalty will be paid to the opponent, resulting in a decrease in the consumer's utility.

Dang and Huhns (2006) have proposed an extension to existing negotiation protocols which lets both service requestors and service providers manage several negotiation processes in parallel. Siebenhaar et al. (2012) have proposed an architecture for concurrent negotiations in cloud-based systems and developed a two-phase negotiation protocol to facilitate the negotiation. However, no novel concession strategies are proposed for coordinating all the negotiation threads in these two approaches. An et al. (2006) design a mechanism for concurrent negotiation where the concession is made based on four factors: time, opponents’ behavior, other negotiation threads, and the competition. Although they consider other threads’ influence on an agent’s negotiation strategies, the coordination is very simple that the consumer should not concede more than the best offer available from among all the opponents. This is quite straightforward since there is no meaning of conceding more than what the consumer can achieve by accepting the best available offer from providers.

It is worth noting that, in one-to-many negotiation, it is difficult to coordinate across all the multiple threads by adopting the existing concession functions customized for one-to-one bilateral negotiation. In a bilateral negotiation, an offer can be generated by following a decision making function, for example, time-dependent, resource-dependent, or behavior-dependent concession functions (Faratin et al. 1998). However, these functions define how an agent should concede to one provider. This makes it impossible to adjust the concession based on the negotiation with other opponents in the context of one-to-many negotiation. In the method we propose, we incorporate the information of the other threads by comparing a new offer with the offers generated concurrently in other threads. In order for an offer to be proposed to a provider, the new offer has to have a higher satisfaction degree or higher (or at least equal) utility than the offers to other providers.

Efficient Concessions in One-to-many Concurrent Negotiations

We first introduce the basic concepts of the one-to-many concurrent negotiation model, then define efficient concessions, and finally formulate the problem of efficient concessions in one-to-many negotiation.

Basic Concepts of the One-to-many Negotiation Model

In a market, a service can be offered by several providers and we assume the number of providers is n. A
consumer can negotiate with these providers concurrently, and a one-to-many concurrent negotiation model can be considered as several one-to-one bilateral negotiations carried out concurrently. Each one-to-one bilateral negotiation is referred to as a negotiation thread. In each thread, the consumer and provider negotiate on the QoS attributes of the service, such as price, availability, response time, and data quality; we assume there are \( m \) attributes to be negotiated over. A QoS attribute can be classified as either a positive or a negative attribute and an attribute is positive/negative if a higher/lower value indicates a higher quality. For example, price is a negative attribute for a consumer and a positive attribute for a provider while quality is a positive attribute for a consumer and a negative attribute for a provider.

An offer is a vector of values, with one value for each QoS attribute. In order for a consumer (or provider) to evaluate a given offer, a utility function is used to map all the QoS attributes into a single value (Keeney and Raiffa 1976). In the automated negotiation literature, a linear utility function has been widely adopted (Faratin et al. 2002; Nguyen and Jennings 2005; Lau et al. 2008). An offer \( o^A = (q^A_1, ..., q^A_k, ..., q^A_m) \) for an agent \( A \) (A can be a consumer or a provider) is then evaluated by a utility function

\[
U^A(o^A) = \sum_{k=1}^{m} V^A_k(q^A_k) \cdot w^A_k,
\]

where \( w^A_k \) represents the importance of the QoS attribute \( k \) to agent \( A \) satisfying \( \sum_{k=1}^{m} w^A_k = 1 \), and \( V^A_k(q^A_k) \) is the scaling function for attribute \( k \) that maps \( q^A_k \) into \([0, 1]\) ensuring the utility function is not biased by the relative magnitude of any attribute. The scaling function \( V^A_k(q^A_k) \) is defined as

\[
V^A_k(q^A_k) = \begin{cases} 
\frac{q^A_k - q^A_{k,\min}}{q^A_{k,max} - q^A_{k,\min}} & \text{for negative attribute } k, \\
\frac{q^A_{k,\max} - q^A_k}{q^A_{k,max} - q^A_{k,\min}} & \text{for positive attribute } k,
\end{cases}
\]

where \( q^A_{k,\min} \) and \( q^A_{k,max} \) are the minimal and maximal values of attribute \( k \) available in the market. The range for the scaling function and the utility function is \([0, 1]\).

The consumer has a deadline \( t^C_{i,\max} \) by when he/she must terminate the negotiation. Similarly, each provider \( i \) (\( i = 1, ..., n \)) has a deadline \( t^P_{i,\max} \). In a negotiation thread, the consumer and provider exchange offers and counter-offers in each negotiation round until an offer is accepted by the opponent or one player opts out of the negotiation (typical if its deadline is reached). An offer (or counter offer) is generated by adopting different concession strategies, for example, time-dependent, resource-dependent, and behavior-dependent concessions. An offer is considered acceptable to the consumer (or provider) if the utility of this offer is higher than the utility of the offer the consumer (or provider) is planning to propose in the next round. The consumer and each provider has a reservation utility, i.e., \( U^C_{\text{reserved}} \) and \( U^P_{\text{reserved}} \), which is the least utility level they are willing to concede to. In this work, we assume the negotiation deadlines, the functional forms of the concession rates, and the reservation utilities adopted by the agents are kept private to themselves. Consequently, it is not possible to compute a best offer agreeable to both the consumer and the provider using one-step optimization approaches (Fatima et al. 2004).

In the one-to-many concurrent negotiation, once the consumer reaches an agreement with one provider, the consumer opts out of the negotiation with other providers. If the consumer has not reached an agreement by the deadline, then the negotiation fails.

**Efficient Offers**

Before we discuss our coordinated concessions, we first define some basic concepts that will be used in our proposed approach. In the one-to-many concurrent negotiations, a consumer generates an offer for each provider and each offer is evaluated by the utility of the offer for the consumer and the favorability of this offer to the corresponding provider. In this paper, to measure how favorable this offer is to the provider, we use a notion of *preferability* between the offer to be proposed and a provider's recent offer, which is analogous to the concept of similarity proposed by Faratin et al. (2002). We first define the *preferability* between two attribute values and then define the *preferability* between two offers.

**Definition 1. Preferability between two values of an attribute:** Given two attribute values \( q^C_{i,k} \) and
We use an example to illustrate how a consumer evaluates a set of offers based on the utilities and preferabilities as shown in Table 1. The preferability between two offers is defined as a weighted summation of the preferability of all the attributes.

\[ \text{preferability of } q_{x,k} \text{ over } q_{y,k} \text{ for provider } i \]  
\[ \text{is defined as} \]
\[ \text{Pref}(q_{x,k}, q_{y,k}) = \sum_{k=1}^{m} w_{q_{i,k}}^p \times \text{Pref}(q_{x,k}, q_{y,k}) \]  
\[ (1) \]  
\[ \text{where } w_{q_{i,k}}^p \text{ represents the level of importance that the consumer believes provider } i \text{ places on attribute } k \text{ satisfying } \sum_{k=1}^{m} w_{q_{i,k}}^p = 1. \]  
If the consumer has no such information, he/she may assign equal weights to all the attributes or learn the provider’s preferences in the negotiation process (Nielsen and Jensen 2004; Jonker et al. 2007; Hindriks and Tykhonov 2008; Buffett and Spencer 2007). Based on equations (4) and (5), we have
\[ \text{Pref}(o_{i,x}, o_{i,y}) = 1 + \sum_{k=1}^{m} w_{q_{i,k}}^p \times (V_{q_{i,k}}^x(q_{x,k}) - V_{q_{i,k}}^y(q_{y,k})) \]  
\[ (2) \]  
\[ \text{Pref}(o_{i,x}, o_{i,y}) \]  
\[ \text{is used to measure the preferability between two offers } o_{i,x} \text{ and } o_{i,y} \text{ to provider } i \text{ by considering the provider’s preferences on all the attributes. If } \text{Pref}(o_{i,x}, o_{i,y}) > 1, U^p(o_{i,x}) > U^p(o_{i,y}), o_{i,x} \text{ is more preferable than } o_{i,y} \text{ to provider } i; \text{ if } \text{Pref}(o_{i,x}, o_{i,y}) = 1, U^p(o_{i,x}) = U^p(o_{i,y}), o_{i,x} \text{ is equally preferable as } o_{i,y} \text{ to provider } i; \text{ if } \text{Pref}(o_{i,x}, o_{i,y}) < 1, U^p(o_{i,x}) < U^p(o_{i,y}), o_{i,x} \text{ is less preferable as compared to } o_{i,y} \text{ to provider } i. \]

The utility indicates how valuable an offer is to the consumer while the preferability shows how valuable the offer is to the provider. Although the consumer would prefer an offer with a higher utility, the consumer also needs to consider the likelihood this offer will be accepted by the provider. Normally, in one negotiation thread, if an offer has a higher utility than another to the consumer, then the preferability of this offer could be lower than the other to the provider, indicating that this offer is less likely to be accepted by the provider compared to the other one. Therefore, the consumer should balance the utility (to herself) and the preferability (to the provider) when generating an offer.

We use an example to illustrate how a consumer evaluates a set of offers based on the utilities and preferabilities.

**Example 1.** Suppose the consumer is going to propose four offers \( o_{1}^c, o_{2}^c, o_{3}^c, \) and \( o_{4}^c \), to four providers, with utilities and preferabilities as shown in Table 1.

<table>
<thead>
<tr>
<th>Offer</th>
<th>Utility</th>
<th>Preferability</th>
</tr>
</thead>
<tbody>
<tr>
<td>( o_{1}^c )</td>
<td>0.6</td>
<td>0.8</td>
</tr>
<tr>
<td>( o_{2}^c )</td>
<td>0.8</td>
<td>0.6</td>
</tr>
<tr>
<td>( o_{3}^c )</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>( o_{4}^c )</td>
<td>0.5</td>
<td>0.6</td>
</tr>
</tbody>
</table>

**Table 1.** The utilities and preferabilities for the offers

---

Although the consumer may not have information about a provider’s preference, the consumer can learn such information either in the negotiation process or from the provider’s negotiation records (Nielsen and Jensen 2004; Jonker et al. 2007; Hindriks and Tykhonov 2008; Buffett and Spencer 2007). When the consumer negotiates with the help of a service broker (Elfataty and Layzell 2004), the provider’s preference is more likely to become available because the broker can learn the provider’s preference from previous experience.
As shown in Table 1, both offer $o^C_i$ and $o^E_i$ are better than offer $o^E_j$ for the consumer since the utility and preferability of both $o^C_i$ and $o^E_i$ are higher than that of $o^E_j$. Further, offer $o^C_i$ is also better than offer $o^E_j$ because although the preferability of $o^C_i$ is equal to that of $o^E_i$, the utility of $o^E_i$ is higher than that of $o^C_i$. Therefore, $o^C_i$ is the worst among the four offers.

None of the three offers $o^C_i$, $o^E_i$, and $o^F_i$ are clearly preferable relative to each other, to compare any two such offers $o^C_i$ and $o^E_i$, a Weighted Product Model (WPM) is used (Bridgman 1922; Miller and Starr 1969; Triantaphyllou et al. 1998) as follows

$$P \left( \frac{o^C_i}{o^F_i} \right) = \frac{\mu^C(o^C_i)}{\mu^E(o^E_i)} \alpha \cdot \frac{\text{Pref}(o^C_i)}{\text{Pref}(o^F_i)} \beta$$

(7)

where $\alpha$ and $\beta$ are the weights for the utility and preferability, respectively, and they satisfy $\alpha+\beta=1$. If $P \left( \frac{o^C_i}{o^F_i} \right)>1$, offer $o^C_i$ is more preferable than $o^C_i$ for the consumer. Table 2 lists three different cases where the offers are evaluated given different $\alpha$ and $\beta$. For example, in case 1, we get $P \left( \frac{o^C_i}{o^E_i} \right)<1$ and $P \left( \frac{o^E_i}{o^C_i} \right)>1$, indicating that $o^C_i$ is less preferable than $o^C_i$ and $o^E_i$ is more preferable than $o^E_i$. That is to say, the ranking of these three offers is as follows: $R(o^C_i)<R(o^E_i)<R(o^C_i)$, where $R(o^C_i)$ represents the ranking of $o^C_i$ to the consumer.

<table>
<thead>
<tr>
<th>Different cases</th>
<th>$P \left( \frac{o^C_i}{o^E_i} \right)$</th>
<th>$P \left( \frac{o^E_i}{o^C_i} \right)$</th>
<th>$P \left( \frac{o^E_i}{o^F_i} \right)$</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1: $\alpha=0.7$ and $\beta=0.3$</td>
<td>0.891</td>
<td>0.934</td>
<td>1.048</td>
<td>$R(o^C_i)&lt;R(o^E_i)&lt;R(o^F_i)$</td>
</tr>
<tr>
<td>Case 2: $\alpha=0.5$ and $\beta=0.5$</td>
<td>1</td>
<td>0.990</td>
<td>0.990</td>
<td>$R(o^C_i)=R(o^E_i)&lt;R(o^F_i)$</td>
</tr>
<tr>
<td>Case 3: $\alpha=0.3$ and $\beta=0.7$</td>
<td>1.122</td>
<td>1.048</td>
<td>0.934</td>
<td>$R(o^E_i)&lt;R(o^C_i)&lt;R(o^F_i)$</td>
</tr>
</tbody>
</table>

When there are multiple offers, instead of doing a pairwise comparison between two offers, we can calculate the product without ratios (Triantaphyllou et al. 1998) and we referred to the product as the satisfaction degree in this work.

**Definition 3. Satisfaction Degree:** Given an offer $o^P_{t-1}=[q^P_{(t-1)}]$ ($k=1,\ldots,m$ and $t>1$) proposed by provider $i$ at time ($t-1$), the consumer satisfaction degree on the offer $o^C_t=[q^C_t]$ to be proposed at time $t$ is defined as

$$f(o^C_t) = U^C(o^C_t)\alpha(t) \cdot \text{Pref}(o^P_{t-1}, o^C_t)\beta(t)$$

(8)

where $U^C(o^C_t)$ is the utility of the offer $o^C_t$ for the consumer, and $\text{Pref}(o^P_{t-1}, o^C_t)$ is the provider $i$’s preferability of this offer over his/her recent offer. Further, $\alpha(t)$ and $\beta(t)$ are the weights for utility and preferability satisfying $\beta(t) = \left\{ \begin{array}{ll} 1 - \alpha(t), & \text{if } \text{Pref}(o^P_{t-1}, o^C_t) < 1; \\ 0, & \text{otherwise} \end{array} \right.$ The function $f(o^C_t)$ represents how satisfied the consumer is with the offer $o^C_t$, with a higher value indicating a higher satisfaction level. When $\text{Pref}(o^P_{t-1}, o^C_t) \geq 1$, the offer to be proposed $o^C_t$ is more preferable to the provider than the recent offer $o^P_{t-1}$ proposed by the provider, and therefore this offer is considered acceptable to this provider. Under such conditions, the consumer satisfaction degree on this offer is only dependent upon the utility of this offer, thus, we set $\beta(t)=0$, resulting in $f(o^C_t) = U^C(o^C_t)\alpha(t)$.

$\alpha(t)$ is used to balance the trade-off between the utility and preferability when evaluating an offer. In general, a large value of this factor implies that the consumer weighs the utility more important than the preferability whereas it is opposite when $\alpha(t)$ is small. Because a negotiation is fundamentally time-dependent, consistent with the literature (Fatatin et al. 1998, Sim et al. 2003), we define factor $\alpha(t)$ as

$$\alpha(t) = 1 - \left( \frac{\min(t, t_{\text{max}})}{t_{\text{max}}} \right)^{\frac{1}{\beta\text{max}}},$$

where $t_{\text{max}}$ is the deadline for the consumer. The function satisfies $0 \leq \alpha(t) \leq 1$, $\alpha(0)=1$, and $\alpha(t_{\text{max}})=0$. When $t=0$, $\alpha(t)=1$, $f(o^C_t)=U^C(o^C_t)$ and the satisfaction degree is only dependent upon the utility; when $t=t_{\text{max}}$, $\alpha(t)=0$, $f(o^C_t)=\text{Pref}(o^P_{t-1}, o^C_t)$ and the satisfaction degree is only
dependent upon the preferability. \( v_i \) is a coefficient that determines the degree of convexity/concavity of the function \( a(t) \). If \( v_i = 1 \), \( a(t) \) is linear and the consumer weighs utility more in the first half of the negotiation period and weighs preferability more in the second half of the negotiation period. If \( v_i > 1 \), \( a(t) \) is concave and the consumer is eager to reach an agreement and beginning to place more importance on preferability than utility even in the first half of the negotiation period. If \( v_i < 1 \), \( a(t) \) is concave and the consumer is patient and weighs preferability over utility until the second half of the negotiation period. Since the consumer has the same deadline for all the negotiation threads, the time effect on each thread is the same. We set \( v_i \) equal for all the threads and as a result, \( a(t) \) is also equal for all the threads.

**Definition 4. Dominated by:** Given two offers \( o^c_{i,t} \) and \( o^c_{h,t} \) the consumer is planning to propose in thread \( t \) and \( h \), if \( U^c(o^c_{i,t}) < U^c(o^c_{h,t}) \) and \( f(o^c_{i,t}) \leq f(o^c_{h,t}) \), where \( f(o^c_{i,t}) \) and \( f(o^c_{h,t}) \) are the satisfaction degrees for offer \( o^c_{i,t} \) and \( o^c_{h,t} \) respectively, then offer \( o^c_{i,t} \) is dominated by \((<)\) offer \( o^c_{h,t} \), and is denoted as \( o^c_{i,t} \prec o^c_{h,t} \).

\( f(o^c_{h,t}) \leq f(o^c_{i,t}) \) indicates that the offer \( o^c_{h,t} \) in thread \( h \) is at least as preferable as the offer \( o^c_{i,t} \) in thread \( t \) to the consumer. Thus, the consumer should avoid conceding more in thread \( t \) than in thread \( h \). Otherwise, the offer in thread \( t \) will be dominated by the offer in thread \( h \) and the consumer should avoid proposing such an offer. We define the concept of domination based on satisfaction degree rather than preferability considering that the consumer’s preferences on offers change over time.

As shown in Table 1, \( U^c(o^c_{1}) \prec U^c(o^c_{2}) \prec U^c(o^c_{3}) \prec U^c(o^c_{4}) \), then, we derive the domination relationships between the offers under three different cases as shown in Table 3.

<table>
<thead>
<tr>
<th>Case</th>
<th>( \alpha ) and ( \beta )</th>
<th>Satisfaction Degree</th>
<th>Dominations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1:</td>
<td>( \alpha = 0.7 ) and ( \beta = 0.3 )</td>
<td>( f(a^c_{i}) &lt; f(a^c_{h}) \prec f(a^c_{j}) \prec f(a^c_{k}) )</td>
<td>( o^c_{i} \prec o^c_{h} \prec o^c_{j} \prec o^c_{k} )</td>
</tr>
<tr>
<td>Case 2:</td>
<td>( \alpha = 0.5 ) and ( \beta = 0.5 )</td>
<td>( f(a^c_{i}) &lt; f(a^c_{h}) \prec f(a^c_{j}) \prec f(a^c_{k}) )</td>
<td>( o^c_{i} \prec o^c_{j} \prec o^c_{h} \prec o^c_{k} )</td>
</tr>
<tr>
<td>Case 3:</td>
<td>( \alpha = 0.3 ) and ( \beta = 0.7 )</td>
<td>( f(a^c_{i}) &lt; f(a^c_{h}) \prec f(a^c_{j}) \prec f(a^c_{k}) )</td>
<td>( o^c_{i} \prec o^c_{j} \prec o^c_{h} \prec o^c_{k} )</td>
</tr>
</tbody>
</table>

**Definition 5. Efficient offer:** Given a set of offers \( o^c_i = \{o^c_{i,t}\} \) \((i=1, ..., n)\) the consumer is going to propose to the providers, an offer \( o^c_{i} \) is efficient if and only if this offer is not dominated by any other offer.

In Example 1, offers \( o^c_i, o^c_k \), and \( o^c_j \) are efficient in case 1; offers \( o^c_i \) and \( o^c_j \) are efficient in case 2; and only offer \( o^c_i \) is efficient in case 3.

**Definition 6. Efficient offer set:** Given a set of offers \( o^c_i = \{o^c_{i,t}\} \) \((i=1, ..., n)\) the consumer is going to propose to the providers, this set of offers is efficient if every offer is an efficient offer.

**Problem Formulation for Efficient Concession**

We propose a coordinated concession approach that can generate a set of efficient offers in one-to-many concurrent negotiation. Based on the definition of efficient offer, we first derive the efficiency constraints in Proposition 1. As long as these constraints are satisfied, the efficiency of the offers can be guaranteed.

**Proposition 1. Efficiency Constraints:** Suppose \( o^c_{i,t} \) is the offer the consumer is going to propose to provider \( i \) \((i=1, ..., n)\), offer set \( o^c = \{o^c_{i,t}\} \) is efficient if and only if \( \forall i, h \in \{1, ..., n\}, U^c(o^c_{i,t}) \geq U^c(o^c_{h,t}) \) whenever \( f(o^c_{i,t}) \leq f(o^c_{h,t}) \).

\(^2\) The efficiency for a generated offer as defined in our paper is different from Pareto efficiency for negotiation outcomes as commonly discussed in the negotiation literature. In the former, the goal is to make sure an offer proposed by a consumer to a provider is not inferior to (or dominated by) the offers proposed by the same consumer to other providers. The latter refers to the notion that no one can be made better off without making someone else worse off by making any changes of the offers (Rubinstein 1982).
Proposition 1 can be proven by using the definition of efficiency. We provide the proofs for all the propositions in the appendix. Proposition 1 provides a way to transform the concept of efficiency into a set of constraints, which can be used to verify whether a set of offers is efficient.

Given that the satisfaction degree is a trade-off between the utility and the preferability, it is to be maximized when a consumer generates offers in each thread. Moreover, the efficiency of the offers can be guaranteed by satisfying the efficiency constraints. Therefore, the problem of generating efficient offers in the one-to-many concurrent negotiation is formulated as

**Problem P1**

Maximize \( f(\pi^c_i) \), for \( i = 1, ..., n \) \hspace{1cm} (9)

Subject to \( U^c(\pi^c_i) \geq U^c_{\text{reserved}}, \) for \( i = 1, ..., n \) \hspace{1cm} (10)

\[ U^c(\pi^c_i) \geq U^c(\pi^c_h), \] whenever \( f(\pi^c_i) < f(\pi^c_h), \forall i, h \in \{1, ..., n\} \) \hspace{1cm} (11)

\[ q^\text{min}_{i,k} \leq q^c_{i,k} \leq q^\text{max}_{i,k}, \] for \( i = 1, ..., n \) and \( k = 1, ..., m \) \hspace{1cm} (12)

The objective functions maximize the satisfaction degree of each offer to be proposed. Constraint (10) ensures that the utility of an offer to be proposed is not lower than the consumer’s reservation utility \( U^c_{\text{reserved}} \), the least utility level of an offer acceptable to the agent. Constraint (11) is used to guarantee the efficiency of the offers and Constraint (12) ensures the value of the generated offer lies within its valid domain.

**Problem Solution of Efficient Concessions in One-to-many Concurrent Negotiations**

In this section, we propose an approach to solve the problem of determining efficient offers in one-to-many concurrent negotiations (Problem P1). First, we maximize the satisfaction degrees independently in each thread without considering the efficiency constraints (Equation 11). In doing so, some of the offers may turn out to be efficient while others may not. Then, we adjust each inefficient offer in an iterative manner until all the offers are efficient.

**Maximizing Satisfaction Degree Independently**

The problem of maximizing the satisfaction degree in a thread \( i (i = 1, ..., n) \) is as follows.

**Problem P2**

Maximize \( f(\pi^c_i) \)

Subject to \( U^c(\pi^c_i) \geq U^c_{\text{reserved}} \)

\[ q^\text{min}_{i,k} \leq q^c_{i,k} \leq q^\text{max}_{i,k}, \] for \( k = 1, ..., m \)

Problem P2 differs from Problem P1 in that the efficiency constraints (11) are not incorporated. To solve problem P2, we first show that this problem is a concave optimization problem and then formulate its Lagrangian. Since there are a large number of constraints, it is still not easy to identify the optimal solution. Then, we discuss the properties of the optimal solution and propose an efficient algorithm to search for the optimal solution.

**Lagrangian Problem Formulation**

**Proposition 2. Concavity:** \( f(\pi^c_i) \) is a concave function.

We prove Proposition 2 by first showing that the two compounds of \( f(.) \) are both concave and then show the product is also concave.

Since the objective in Problem P2 is concave and the constraints are all linear, Problem P2 is a concave optimization problem. For such a problem, the Karush-Kuhn-Tucker (KKT) conditions are both necessary and sufficient for primal-dual optimality (Boyd and Vandenberghe 2003).
For the sake of simplicity, let \( x_k = q^c_{ik,t} \) and \( X = (x_1, x_2, \ldots, x_m) = o^c_t = (q^c_{1t,1}, q^c_{2t,2}, \ldots, q^c_{mt,m}) \). The Lagrangian for Problem P2 is

\[
L(X) = f(X) + \sum_{k=1}^m \mu_k \cdot (-x_k + q^\text{max}_{ik}) + \sum_{k=1}^m v_k \cdot (x_k - q^\text{min}_{ik}) + \lambda \left( \sum_{k=1}^m w^c_{ik} \cdot \frac{q^\text{max}_{ik} - x_k}{q^\text{max}_{ik} - q^\text{min}_{ik}} - U_{\text{reserved}} \right)
\]

where \( \mu_k, v_k \ (k \in \{1, \ldots, m\}) \), and \( \lambda \) are the Lagrangian multipliers. Given P2 is a concave optimization problem, an offer satisfying the KKT conditions as follows is an optimal solution.

\[
\begin{align*}
(1) & \quad \frac{\partial L(X)}{\partial x_j} = \frac{\partial f(X)}{\partial x_j} - \mu_j + v_j - \frac{\lambda w^c_{ij}}{q^\text{max}_{ik} - q^\text{min}_{ik}} = 0; \\
(2) & \quad -x_k + q^\text{max}_{ik} \geq 0; \quad x_k - q^\text{min}_{ik} \geq 0; \quad \sum_{k=1}^m w^c_{ik} \cdot \frac{q^\text{max}_{ik} - x_k}{q^\text{max}_{ik} - q^\text{min}_{ik}} - U_{\text{reserved}} \geq 0 \\
(3) & \quad \mu_j \geq 0; \quad v_k \geq 0; \quad \lambda \geq 0; \\
(4) & \quad \mu_j \cdot (-x_k + q^\text{max}_{ik}) = 0; \quad v_k \cdot (x_k - q^\text{min}_{ik}) = 0; \quad \lambda \left( \sum_{k=1}^m w^c_{ik} \cdot \frac{q^\text{max}_{ik} - x_k}{q^\text{max}_{ik} - q^\text{min}_{ik}} - U_{\text{reserved}} \right) = 0
\end{align*}
\]

where \( \frac{\partial f(X)}{\partial x_j} = d_j \cdot K_j(X), \quad d_j = \frac{1}{q^\text{max}_{ik} - q^\text{min}_{ik}} \cdot U^c(o^c_{ik,t} q(t)^{(t-1)}) - \text{Pref}(o^c_{ik,t-l}, o^c_{ik,t}) \cdot \beta(t)^{(t-1)} \), and \( K_j(X) = \alpha_j(t) \cdot w^c_{ij} \cdot \left( -1 + \sum_{k=1}^m \left( \frac{w^c_{ikj} - q^\text{min}_{ik}}{q^\text{max}_{ik} - q^\text{min}_{ik}} \right) \right) + \beta(t) \cdot w^c_{ij} \cdot \left( \sum_{k=1}^m \left( \frac{w^c_{ikk} - q^\text{max}_{ik}}{q^\text{max}_{ik} - q^\text{min}_{ik}} \right) \right) - \sum_{k=1}^m x_k \cdot \left( \frac{\alpha_j(t) \cdot w^c_{ikj} + \beta(t) \cdot w^c_{ij} \cdot q^\text{max}_{ik}}{q^\text{max}_{ik} - q^\text{min}_{ik}} \right). \]

**An Efficient Algorithm for Solving the Problem**

In this section, we first discuss the properties of optimal solutions and then propose an efficient algorithm for searching for a solution satisfying the KKT conditions.

For simplicity but without loss of generality, we use the attributes negative to the consumer to elaborate the proposed algorithm.

**Proposition 3.** Let \( o^c_t = (x_1, \ldots, x_m) \) be an optimal solution for Problem P2. If \( \frac{w^c_{ij} - w^c_{ik}}{w^c_{ij}} \) \((k, j \in \{1, \ldots, m\})\), then the consumer should never start the concession in attribute \( k \) before the concession in attribute \( j \) is completely done, i.e., if \( x_k > q^\text{max}_{ik} \), then \( x_j = q^\text{max}_{ij} \), where \( q^\text{min}_{ik} \) is the most preferred value on attribute \( k \) for the consumer and \( q^\text{max}_{ij} \) is the least acceptable value on attribute \( j \) for the consumer.

The intuition behind this proposition is that if the consumer obtains the same amount of utility by conceding in attribute \( k \) or \( j \), then the provider would prefer the consumer’s concession in attribute \( j \) than in attribute \( k \) because that would result in a higher preferable for the provider.

Proposition 3 provides an important insight regarding how the consumer should generate an offer. That is, the consumer should first rank each attribute based on the ratio of a provider’s weight to the consumer’s weight on the attribute. Then, the consumer should concede in the attributes in the order of their rankings. Only when the concession on a higher ranked attribute is completely exhausted should the concession on a lower ranking attribute be started.

Based on Proposition 3, we develop an efficient algorithm to search for an optimal solution. Let \( YA \) denote the set of attributes where the concessions are exhausted completely and \( NA \) denote the set of attributes where the concessions are not started yet. Therefore, for any \( x \in YA \), \( q^c_{i} = q^c_{i}^{\text{max}} \) and for any \( r \in NA \), \( q^c_{i} = q^c_{i}^{\text{min}} \). Initially, \( YA = \emptyset \) and \( NA = \{1, \ldots, m\} \). The searching process is as follows. We first identify attribute \( j \) with the highest rank\(^3\) in \( NA \) and update \( NA \) by removing \( j \) from it. The value of \( q^c_{i} \) could be one of the three cases: \( q^c_{i} = q^c_{i}^{\text{min}} \cdot q^c_{i}^{\text{min}} < q^c_{i}^{\text{max}} \), and \( q^c_{i} = q^c_{i}^{\text{max}} \). Based on the KKT conditions, we can derive the requirements for each case, which are summarized in Table 4. As long as the requirements are satisfied, the optimal solution is found. Otherwise, the concession in this attribute is completely exhausted; we add \( j \) to \( YA \) and

---

\(^3\) When there are multiple attributes with the same rank, there may be more than one optimal solution. Anyhow, we can still find one optimal solution by randomly ordering those attributes.
set $q_{i,j}=q_{i,j}^{\text{max}}$. This process is repeated until the optimal solution is found. If by the end of the iteration, no requirements shown in Table 4 are met, the offer with the reservation utility is the optimal solution.

Table 4. Requirements derived based on the KKT conditions

<table>
<thead>
<tr>
<th>Cases</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i) $U^c(o_{i,j}^c) &gt; U_{\text{reserved}}$ and $x_j=q_{i,j}^{\text{min}}$</td>
<td>$K_j(X)\leq 0$, $K_r(X)\leq 0$, and $K_c(X)\geq 0$</td>
</tr>
<tr>
<td>(ii) $U^c(o_{i,j}^c) &gt; U_{\text{reserved}}$ and $q_{i,j}^{\text{min}} &lt; x_j &lt; q_{i,j}^{\text{max}}$</td>
<td>$K_j(X) = 0$, $K_r(X) \leq 0$, and $K_c(X) \geq 0$</td>
</tr>
<tr>
<td>(iii) $U^c(o_{i,j}^c) &gt; U_{\text{reserved}}$ and $x_j=q_{i,j}^{\text{max}}$</td>
<td>$K_j(X) \geq 0$, $K_r(X) \leq 0$, and $K_c(X) \geq 0$</td>
</tr>
</tbody>
</table>

How do we derive the requirements in Table 4 is explained as follows. If $x_j=q_{i,j}^{\text{min}}$ (case i), then based on KKT constraint 4 $\mu_k \ast (x_k - q_{i,k}^{\text{max}}) = 0$ and $v_k \ast (x_k - q_{i,k}^{\text{min}}) = 0$ ($k=1,..,m$), we get $\mu_i = 0$, $v_i = 0$, and $\mu_r = 0$. When $U^c(o_{i,j}^c) > U_{\text{reserved}}$, based on the KKT constraint 4 $\lambda \{ \sum_{k=1}^{m} w_{ik}^p * \frac{q_{i,k}^{\text{min}} - x_k}{q_{i,k}^{\text{max}} - q_{i,k}^{\text{min}}} - U_{\text{reserved}} \} = 0$, we get $\lambda = 0$.

For $x_j=q_{i,j}^{\text{max}}$ (case iii), we get $K_j(X) \geq 0$, $K_r(X) \geq 0$, and $K_c(X) \leq 0$. Further, if $q_{i,j}^{\text{min}} > x_j < q_{i,j}^{\text{max}}$ (case ii), we get $K_j(X) = 0$, $K_r(X) \geq 0$, and $K_c(X) \leq 0$. If $K_j(X) = 0$, we can compute $x_j$ (i.e. $q_{i,j}^c$) as follows

$$x_j = \sum_{k=1}^{m} a_k (t) \ast w_{ik}^p \ast \left(1 + \sum_{k=1}^{m} \left( w_{ik}^p \ast \frac{p_{k,c}-\min}{q_{i,k}^{\text{max}}-q_{i,k}^{\text{min}}} \right) \right) + \beta_j (t) \ast w_{ij}^p \ast \left( \sum_{k=1}^{m} w_{ik}^p \ast \frac{q_{i,k}^{\text{max}}-q_{i,k}^{\min}}{q_{i,k}^{\text{max}}-q_{i,k}^{\text{min}}} \right)$$

Example 2. In this example, we set the parameters as follows: $m=3$, $\alpha=0.3$, $\beta=0.7$, $q_{i,1}^{\text{min}}=q_{i,2}^{\text{min}}=q_{i,3}^{\text{min}}=10$, $q_{i,1}^{\text{max}}=q_{i,2}^{\text{max}}=q_{i,3}^{\text{max}}=20$, $q_{i,1}=17$, $q_{i,2}=16$, $q_{i,3}=17$, $w_{i,1}^p=0.3$, $w_{i,2}^p=0.5$, $w_{i,3}^p=0.2$, $w_{i,1}^p=0.6$, $w_{i,2}^p=0.3$, and $U_{\text{reserved}}=0.5$.

Since $\frac{w_{i,1}^p}{w_{i,2}^p}=0.6$, and $\frac{w_{i,2}^p}{w_{i,3}^p}=0.5$, we get $\frac{w_{i,1}^p}{w_{i,2}^p}=0.3$. Thus, the consumer starts concessions in the first attribute, followed by the second attribute, and finally in the third attribute. The process of searching for a solution is as shown in Table 5.

Table 5. Example process of searching for a solution

<table>
<thead>
<tr>
<th>Offer</th>
<th>Requirements</th>
<th>Verification</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>(10, 10, 10)</td>
<td>$K_j(X)\leq 0$, $K_r(X)\leq 0$, $K_c(X)\leq 0$</td>
<td>$K_j(X)=0.3903$; $K_r(X)=0.1605$; $K_c(X)=0.0502$</td>
<td>Feasible</td>
</tr>
<tr>
<td>(x_1, 10, 10)</td>
<td>$K_j(X)=0$, $K_r(X)\leq 0$, $K_c(X)\geq 0$</td>
<td>$x_1=31.683&gt;q_{i,1}^{\text{max}}=20$</td>
<td>Infeasible</td>
</tr>
<tr>
<td>(20, 10, 10)</td>
<td>$K_j(X)\geq 0$, $K_r(X)\leq 0$, $K_c(X)\leq 0$</td>
<td>$K_j(X)=0.2103$; $K_r(X)=0.0075$; $K_c(X)=-0.0068$</td>
<td>Infeasible</td>
</tr>
<tr>
<td>(20, x_2, 10)</td>
<td>$K_j(X)\geq 0$, $K_r(X)=0$, $K_c(X)\leq 0$</td>
<td>$x_2=10.5$; $K_j(X)=0.1985$; $K_c(X)=-0.0095$</td>
<td>Feasible</td>
</tr>
</tbody>
</table>

In this example, the consumer starts the concession with attribute one and examines the solutions $x_1=q_{i,1}^{\text{min}}=10$, $q_{i,1}^{\text{min}}<x_1<q_{i,1}^{\text{max}}$, and $x_0=q_{i,1}^{\text{max}}=20$. Since the corresponding requirements fail to be satisfied, the consumer then starts conceding on the second attribute $x_2$, fixing $x_1=q_{i,1}^{\text{max}}=20$. The requirements are met for $x_2=10.4996$ and the optimal solution is identified as $(x_1, x_2, x_3)=(20, 10.5, 10)$. To find an optimal solution for Problem P2, we need to check each case shown in Table 4 until one of the requirements is satisfied. The total time complexity of checking is $O(m)$. For each check, the verification for $K_j(X)$ ($j=1,..,m$) is required, therefore, the overall time complexity is $O(m^2)$. 

---

Thirty Fourth International Conference on Information Systems, Milan 2013 11
Maximizing Satisfaction Degrees Concurrently with Efficiency Constraints

We have discussed how to maximize satisfaction degrees independently without considering the efficiency constraints. Next, we show how to make adjustments on inefficient offers to achieve efficiency.

According to Proposition 1, a set of offers are efficient if ∀l, h ∈ {1, ..., n}, \( U^C(o_{lt}) \geq U^C(o_{ht}) \) whenever \( f(o_{lt}) \leq f(o_{ht}) \). As a result, the offers generated for each negotiation thread independently by solving Problem P2 may not be efficient and there may exist l, h ∈ {1, ..., n}, satisfying both \( U^C(o_{lt}) < U^C(o_{ht}) \) and \( f(o_{lt}) \leq f(o_{ht}) \). In such a situation, \( o_{lt} \prec o_{ht} \) and we need to adjust offer \( o_{lt} \) in thread \( l \) to make sure offer \( o_{lt} \) is efficient.

The offers generated independently may have different satisfaction degrees and the consumer prefers an offer with a higher satisfaction degree than others. Assuming the satisfaction degree of offer \( o_{lt} \) is higher than that of others, i.e., \( f(o_{lt}) > f(o_{ht}) \) (∀i, i≠h), then \( o_{lt} \) is most preferable to the consumer and it is the first priority to preserve its optimal value by fixing this offer. After this offer is fixed, any other offer \( o_{lt} \) satisfying \( U^C(o_{lt}) < U^C(o_{ht}) \) should be adjusted because such an offer is an inefficient offer. This is accomplished by solving Problem P3

**Problem P3**

Maximize \( f(o_{lt}) \)

Subject to \( U^C(o_{lt}) \geq U^C(o_{ht}) \)

\[ q_{ik}^{min} \leq q_{ik}^{l} \leq q_{ik}^{max} \text{, for } k=1, ..., m \]

This problem is similar to P2 except that the constraint \( U^C(o_{lt}) \geq U^C(o_{ht}) \) is now updated as \( U^C(o_{lt}) \geq U^C(o_{ht}) \) and we can solve this problem similarly.

After the adjustment, the consumer will choose an offer with the highest satisfaction degree among the remaining offers, fix this offer and adjust other inefficient offers. This procedure is repeated until all the offers are efficient.

After an offer is adjusted, the satisfaction degree for this offer cannot be increased since the satisfaction degree is maximized in a reduced searching space than before the adjustment. An offer fixed earlier has a higher satisfaction degree than that of a later fixed offer, thus, the earlier fixed offer cannot be dominated by any offer fixed later and it will remain to be efficient until the iteration is done. Thus, all the offers are guaranteed to be efficient.

Let \( SY \) denote the set of numbers indexing the offers which are already fixed and \( SN \) denote the set of numbers indexing the offers to be fixed (note that \( SY \cup SN = \{1, ..., n\} \)). The procedure for maximizing all the objectives and achieving efficiency is summarized in Figure 1. First, \( SY \) and \( SN \) are initialized as \( \emptyset \) and \( \{1, ..., n\} \) respectively (step 1). Then we select the offer with the highest satisfaction degree among all the offers to be fixed (step 3). If there are several offers whose satisfaction degrees are all equal to the maximal value, we choose the one with the highest utility among them because this offer is not dominated by any other offer (step 4-8). Next, we update \( SY \) and \( SN \) (step 9) and adjust the offers violating the efficiency (step 10-14). The step 3-14 is repeated until \( SN = \emptyset \).

**Procedure 1. Maximizing the satisfaction degrees concurrently while preserving efficiency**

**Input:** a set of offers generated by maximizing the satisfaction degree independently (the output of solving Problem P2 for \( i=1, ..., n \))

**Output:** a set of efficient offers

1. **Initialization:** \( SY \leftarrow \emptyset \) and \( SN \leftarrow \{1, ..., n\} \)
2. **while** \( SN \neq \emptyset \) **do**
3. \( h \leftarrow \arg \max_{i \in SN} f(o_{lt}) \)
4. **for** each \( i \in SN \) **do**
5. **if** \( f(o_{lt}) = f(o_{lt}) \) and \( U(o_{lt}) > U(o_{lt}) \) **then**

12 Thirty Fourth International Conference on Information Systems, Milan 2013
Experimental Evaluations

**Benchmark Methods and Evaluation Metrics**

In order to study the performance of our proposed approach, we have created two benchmarks. In the first benchmark, the consumer makes concessions independently without considering other negotiation threads. We refer to this as Independent_Concession_Bench. In the second benchmark, the consumer also makes concessions independently and if an agreement is reached, instead of opting out of other negotiations immediately, the consumer continues to negotiate with the other providers to seek better offers. The utility of the agreed offer is updated as the reservation utility for the remaining negotiations and when a better offer is available the consumer breaches the agreed contract without paying any penalty. If the consumer has already conceded to a utility level lower than the adjusted reservation utility in one negotiation thread, this thread is terminated. The negotiations end when all the negotiation threads are terminated. This benchmark is actually biased in favor of the consumer and it is referred to as Independent_Concession_Privileged_Bench. Our proposed approach is referred to as Coordinated_Concession. To evaluate the negotiation outcomes, we use the following metrics that are widely adopted in negotiation literature (Faratin et al. 2002; Nguyen and Jennings 2005; Sim et al. 2010): (i) negotiation success rate; (ii) the expected utility to the consumer; (iii) normalized negotiation time for the consumer when negotiation is successful.


**Experimental Procedures**

In this experiment, negotiation is conducted over four QoS attributes and the preferences for consumers and providers are generated randomly satisfying $\sum_{k=1}^{m} w_k \phi_k = 1$. The feasible negotiation range $[q_{k,\text{min}}, q_{k,\text{max}}]$ of attribute $k$ is randomly generated. The negotiation deadline for the consumer and any provider is randomly generated between $[20, 40]$ and a negotiation thread should end before the time exceeds the deadline of either participant. In the negotiation process, the consumer and each provider have a reservation utility and it is randomly set between 0.4 and 0.6 for the consumer and each provider. We assume each provider applies a time-dependent function (Faratin et al. 1998) to make concession. The coefficient $v_i$ that determines the concavity/convexity degree of the concession tactic is randomly generated within the range $[0.4, 2.5]$ for both the consumer and the providers. The consumer can have different levels of information on how the providers weigh different attributes, i.e., no information (no providers' preferences are known), partial information (half of the providers' preferences are known), and full information (all the providers' preferences are known). The qualitative nature of our findings holds for all the settings. In the following section, we report the experimental results under the partial information setting with respect to the number of providers ($n$), while other parameters are randomly generated.

**Results and Discussion**

Figure 3 shows the success rate, expected utility, and negotiation speed achieved by the three approaches with respect to different number of providers ($n$). As shown in Figure 3-a, the three approaches achieve the same success rate. Given that the intention of the coordination is to adjust inefficient offers, it is reasonable that the success rate under the proposed approach will in general not be reduced. Moreover, with more providers available, the success rate increases given that more providers indicate more opportunities for the consumer to reach an agreement.

As observed in Figure 3-b, the proposed approach achieves a higher expected utility than the first benchmark (Independent_Concession_Bench) in our test cases. The reason can be explained as follows. In the first benchmark, the consumer concedes independently without considering the information of other ongoing negotiation threads while in the proposed approach the consumer coordinates all the negotiation threads by adjusting inefficient offers. Therefore, after the coordination, an inferior offer is adjusted to avoid extra concession, which otherwise would result in an agreement with a lower utility. For the same reason, the proposed approach takes slightly more negotiation time than the first benchmark as shown in Figure 3-c. By considering all the negotiation threads, the proposed approach restricts the concessions for some negotiation threads, which otherwise may reach an earlier agreement, thus leading to slightly more negotiation time.

---

4 Given that the computation time of generating an offer is very fast (in milliseconds), we only consider the time used for negotiation here.
In the second benchmark (Independent_Concession_Privileged_Bench), even if one offer is agreed, the consumer continues to negotiate with ongoing providers to seek better offers until all the threads are terminated. Therefore, the second benchmark achieves a higher utility than the first benchmark at the cost of negotiation time. This is consistent with the observation in our experiment.

Moreover, as observed in our experiment, the proposed approach achieves nearly the same utility as the second benchmark with less negotiation time. We should point out that the second benchmark shows the best utility a consumer can achieve in an ideal situation for independent concessions. Given that we assume there is no penalty associated with breaches in the second benchmark, our approach should outperform the second benchmark when the penalty associated with each breach is non-trivial.

Further, when there are more providers available, a higher utility can be achieved with less negotiation time and this is the case for all the approaches discussed here.

Conclusions and Future Work

In a service market, a consumer may negotiate with several providers concurrently. This paper presents a novel method to help the consumer make efficient concessions in each negotiation thread by coordinating across all the threads. Our method differs from the existing automated negotiation methods in that through coordinating different negotiation threads, our method can avoid unnecessary concession when a better offer is available in one-to-many concurrent negotiation, and the existing methods cannot avoid such inefficiency. Our contributions are summarized as follows. First, our approach is the first effort to formally define efficient concessions in the one-to-many concurrent negotiation context. The efficient concession can be guaranteed by satisfying the following requirements: if the consumer cannot achieve a higher satisfaction degree in one thread than in another, the consumer should avoid conceding more in this thread than in the other. Then, we formulate the concession making as a multi-objective optimization problem, where the satisfaction degree of each offer is maximized with a guarantee for the efficiency of the offers. Further, we show how to solve the multi-objective optimization problem. We first maximize the satisfaction degree in each thread independently and then fix the offer with the highest satisfaction degree and adjust the inefficient offers. The procedure is repeated until all the offers are efficient. Our method guarantees the efficiency of the concessions.

To evaluate our proposed method, we have conducted simulation experiments by comparing our approach with two benchmark methods. Through the experiments, we demonstrate that by coordinating all the negotiation process, our proposed approach has achieved a higher utility than individual concession while using a little more negotiation time. Furthermore, our approach achieves a utility close to the best that can be achieved when there is no coordination, with less negotiation time. The method proposed in this research can improve the effectiveness of one-to-many concurrent negotiation and is particularly useful in a market where breaching contracts is costly.

The adoption of service-oriented architecture (SOA) is considered pivotal for the 21st century enterprises (SOA Consortium, 2010). SOA enables quick development of new business applications through integrating and combining existing applications that are developed independently and packaged as services (e.g., software service, grid service, cloud service). In highly dynamic environments where the supply and demand for such services are continuously changing, fixing the price or any aspects of the service may not be possible (Elfatatry and Layzell 2004). In such environments, automated negotiation has been widely recognized as an important mechanism for the service consumer and service provider to dynamically reach an agreement for a service at the time when it is needed. Given its importance in service management, automated negotiation has been incorporated and implemented as a critical module, often in the existing middleware (Ardagna et al. 2011 and Lee and Kang 2012). In such a context, the method we have proposed for automated negotiation will allow a service user to simultaneously negotiate with different providers with great efficiency to add more flexibilities to service provision and business process management.

In the future, we plan to extend our research in the following directions. We have considered a concurrent negotiation model in this paper given that it is more time efficient than a sequential model. In a sequential negotiation model, the sequence in which a consumer negotiates with different providers can impact the negotiation outcome. In an environment where sequential negotiation is applicable, determining the best sequence could be an interesting research problem. Moreover, we can incorporate learning into the
negotiation model to help improve the negotiation outcome. When the consumer requests a composite service constructed by several atomic services, each of which can be offered by several providers, how to negotiate with all the providers to obtain such a composite service is a problem (Mansour et al. 2012) that needs further investigation. In addition, we have not considered the competitions among consumers in this paper. Another extension would be to design negotiation tactics customized to such contexts.

Acknowledgments

The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 149511).

Appendix

A.1. Proof of Proposition 1

Proof: (1) We prove that if \( \forall l, h \in \{1, \ldots, n\}, U^c(o^c_{l,t}) \leq U^c(o^c_{h,t}) \) whenever \( f(o^c_{l,t}) \leq f(o^c_{h,t}) \), then the offer set \( o^c_{l,t} =\{o^c_{l,t}\} \) \((i=1, \ldots, n)\) is efficient. Based on the definition of domination, an offer \( o^c_{l,t} \) is dominated by another offer \( o^c_{h,t} \) if both \( U^c(o^c_{l,t}) < U^c(o^c_{h,t}) \) and \( f(o^c_{l,t}) \leq f(o^c_{h,t}) \) are met. However, once \( f(o^c_{l,t}) \leq f(o^c_{h,t}) \), \( U^c(o^c_{l,t}) \geq U^c(o^c_{h,t}) \) is guaranteed by assumption and thus, we get that no offer is dominated by another. That is to say, each offer is efficient and therefore the offer set \( o^c_{l,t} =\{o^c_{l,t}\} \) \((i=1, \ldots, n)\) is efficient.

(2) We prove that if the offer set \( o^c_{l,h} =\{o^c_{l,h}\} \) \((i=1, \ldots, n)\) is efficient, then \( \forall l, h \in \{1, \ldots, n\}, U^c(o^c_{l,t}) \geq U^c(o^c_{h,t}) \) whenever \( f(o^c_{l,t}) \leq f(o^c_{h,t}) \). Since the offer set is efficient, we get that \( \forall l \in \{1, \ldots, n\} \), offer \( o^c_{l,t} \) is efficient. By the definition of efficiency, we get there exists no offer \( o^c_{h,t} \) \((h \in \{1, \ldots, n\})\), satisfying that \( o^c_{l,t} < o^c_{h,t} \). Therefore, if \( f(o^c_{l,t}) \leq f(o^c_{h,t}) \), \( U^c(o^c_{l,t}) \geq U^c(o^c_{h,t}) \) is guaranteed. Otherwise, we get \( o^c_{l,t} < o^c_{h,t} \), which violates the efficiency of offer \( o^c_{l,t} \).

A.2. Proof of Proposition 2

Proof: Let \( h_1(X) = h_3(h_2(X)) \), where \( h_2(X) = U^c(o^c_{l,h}) \), \( h_3(Y) = Y^{a_1(t)} \) and \( g_1(X) = g_3(g_2(X)) \), where \( g_2(X) = \text{Pref}(o^c_{l,t-1}, o^c_{l,t}) \), and \( g_3(Y) = Y^{b_0(t)} \). We need to prove \( f(X) = f(o^c_{l,t}) = h_1(X) \ast g_1(X) \) is concave.

(1) We prove \( h_1(X) \) is a concave function.

Since \( h_2(X) = \sum_{i=1}^m w^{l}_{ik} \ast \frac{q^{max}_{ik} - x_k}{r^{max}_{ik} - q^{max}_{ik}} \in [0, 1] \) is a concave function, we get \( h_2(\lambda X_1 + (1 - \lambda)X_2) \geq \lambda h_2(X_1) + (1 - \lambda)h_2(X_2) \). Further, \( h_3(Y) = Y^{a_1(t)} \) is a non-decreasing function if \( Y \geq 0 \) and \( a_1(t) \leq 1 \), so we get \( h_3(\lambda h_2(X_1) + (1 - \lambda)h_2(X_2)) \geq h_3(h_2(X_1)) = h_2(X_1) + (1 - \lambda)h_2(X_2) \). Given that \( h_3(Y) \) is also a concave function, we get \( h_1(\lambda h_2(X_1) + (1 - \lambda)h_2(X_2)) \geq \lambda h_1(h_2(X_1)) + (1 - \lambda)h_1(h_2(X_2)) \). Now we have \( h_3(h_2(X_1) + (1 - \lambda)X_2) = h_1(h_2(X_1)) + (1 - \lambda)h_1(h_2(X_2)) \). Since \( h_1(X) = h_3(h_2(X)) \), we get \( h_1(\lambda X_1 + (1 - \lambda)X_2) \geq \lambda h_1(X_1) + (1 - \lambda)h_1(X_2) \) and according to the definition, \( h_1(X) \) is a concave function.

(2) We prove \( h_1(X) \) is a non-increasing function.

Suppose \( X_2 \leq X_1 \). Since \( h_2(X_2) \) is a decreasing function, then we have \( h_2(X_2) \geq h_2(X_1) \). Further, \( h_3(Y) \) is a non-decreasing function, thus, we have \( h_3(h_2(X_2)) \geq h_3(h_2(X_1)) \). Then, we get \( h_1(h_2(X_1)) \geq h_1(h_2(X_2)) \). Thus, \( h_1(X) \) is a non-increasing function.

(3) Similar to (1), we can prove that \( g_1(X) \) is also a concave function.

(4) Similar to (2), we can prove that \( g_1(X) \) is a non-decreasing function.

(5) We prove \( f(X) = h_1(X) \ast g_1(X) \) is a concave function.

Let \( \Delta = f(\lambda X_1 + (1 - \lambda)X_2) - \lambda f(X_1) - (1 - \lambda)f(X_2) \). Once we prove \( \Delta \geq 0 \), \( f(X) \) is a concave function.

\[
\Delta = h_1(\lambda X_1 + (1 - \lambda)X_2) \ast g_1(\lambda X_1 + (1 - \lambda)X_2) - \lambda h_1(X_1) \ast g_1(X_1) - (1 - \lambda)h_1(X_2) \ast g_1(X_2).
\]

Given that \( h_1(X) \) and \( g_1(X) \) are both concave functions, we get
\[ \Delta \geq (\lambda h_1(X_2) + (1 - \lambda) h_1(X_1)) \times (\lambda g_1(X_1) + (1 - \lambda) g_1(X_2)) - \lambda h_1(X_1) \times g_1(X_2) - (1 - \lambda) h_1(X_2) \times g_1(X_2) \]

\[ = \lambda(1 - \lambda)(h_1(X_1) - h_1(X_2)) \times (g_1(X_2) - g_1(X_1)) \]

Given that \( h_1(X) \) is a non-increasing function and \( g_1(X) \) is a non-decreasing function, we get \( h_1(X_i) \geq h_1(X_2) \) and \( g_1(X_1) \geq g_1(X_2) \) if \( X_1 \leq X_2 \) and \( h_1(X_1) \leq h_1(X_2) \) and \( g_1(X_1) \leq g_1(X_2) \) if \( X_1 \geq X_2 \). Therefore, \( (h_1(X_1) - h_1(X_2)) \times (g_1(X_2) - g_1(X_1)) \geq 0 \) is guaranteed. Because \( 0 \leq \lambda \leq 1 \), we get \( \Delta \geq 0 \) and \( f(X) = h_1(X) \times g_1(X) \) is a concave function.

### A.3. Proof of Proposition 3

**Proof:** The proof is by contradiction. Suppose \( x_j' > x_k \), \( x_k' < x_k \), and \( x_r' = x_r \) for any \( r \in \{1, ..., m\}\) \( \setminus \{j, k\} \), and \( \text{Pref}(o_t^{x_j'}, o_t^{x_k'}) = \text{Pref}(o_t^{x_j}, o_t^{x_k}) \).

1. We prove \( U^C(o_t^{x_j}) < U^C(o_t^{x_j'}) \). Since \( \text{Pref}(o_t^{x_j}, o_t^{x_k'}) = \text{Pref}(o_t^{x_j}, o_t^{x_k}) \) and \( x_r' = x_r \) for any \( r \in \{1, ..., m\}\) \( \setminus \{j, k\} \), we get \( w_{q_{j,k}^{x_k' - x_k}}^p + w_{q_{j,k}^{x_j' - x_j}}^p = w_{q_{j,k}^{x_k' - x_k}}^p + w_{q_{j,k}^{x_j' - x_j}}^p \). Further, we get

\[ w_{q_{j,k}^{x_k' - x_k}}^p \times \frac{x_k' - x_k}{q_{j,k}^{x_k' - x_k}} + w_{q_{j,k}^{x_j' - x_j}}^p \times \frac{x_j' - x_j}{q_{j,k}^{x_j' - x_j}} = 0. \]

Then, we get \( \frac{x_k' - x_k}{q_{j,k}^{x_k' - x_k}} = \frac{x_j' - x_j}{q_{j,k}^{x_j' - x_j}} \). Let \( \Delta = U^C(o_t^{x_j}) - U^C(o_t^{x_j'}) \).

Since \( x_r' = x_r \) for any \( r \in \{1, ..., m\}\) \( \setminus \{j, k\} \), we get \( \Delta = w_{q_{j,k}^{x_k' - x_k}}^p \times \frac{x_k' - x_k}{q_{j,k}^{x_k' - x_k}} + w_{q_{j,k}^{x_j' - x_j}}^p \times \frac{x_j' - x_j}{q_{j,k}^{x_j' - x_j}} \). Because

\[ \frac{x_k' - x_k}{q_{j,k}^{x_k' - x_k}} = \frac{x_j' - x_j}{q_{j,k}^{x_j' - x_j}}. \]

we have \( \Delta = w_{q_{j,k}^{x_k' - x_k}}^p \times \frac{x_k' - x_k}{q_{j,k}^{x_k' - x_k}} + w_{q_{j,k}^{x_j' - x_j}}^p \times \frac{x_j' - x_j}{q_{j,k}^{x_j' - x_j}} \).

Because \( -\frac{x_k' - x_k}{q_{j,k}^{x_k' - x_k}} \) and \( -\frac{x_j' - x_j}{q_{j,k}^{x_j' - x_j}} \), we get \( \Delta > 0 \). Further, since \( x_j < x_j' \), we get \( \Delta > 0 \).

2. We prove \( f(o_t^{x_j}) > f(o_t^{x_j'}) \). Since \( f(o_t^{x_j}) = U^C(o_t^{x_j}) \times \text{Pref}(o_t^{x_j}, o_t^{x_k'}) \), \( f(o_t^{x_j'}) = U^C(o_t^{x_j'}) \times \text{Pref}(o_t^{x_j'}, o_t^{x_k}) \), \( U^C(o_t^{x_j}) > U^C(o_t^{x_j'}) \), and \( \text{Pref}(o_t^{x_j}, o_t^{x_k'}) = \text{Pref}(o_t^{x_j'}, o_t^{x_k}) \), we get \( f(o_t^{x_j}) > f(o_t^{x_j'}) \). Thus, \( o_t^{x_j} \) is not an optimal solution, contradicting with the assumption. □
Service Management and IS

References


Bridgman, P. W. 1922. Dimensional Analysis, Yale University Press, New Haven, CN.


