Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

SEN

Software Engineering

Software ENgineering

Negotiating over Bundles and Prices Using Aggregate
Knowledge

D.J.A. Somefun, T.B. Klos, J.A. La Poutré

REPORT **SEN-E0405** JUNE **2004**

CWI is the National Research Institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organization for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

Negotiating over Bundles and Prices Using Aggregate Knowledge

ABSTRACT

Combining two or more items and selling them as one good, a practice called bundling, can be a very effective strategy for reducing the costs of producing, marketing, and selling goods. In this paper, we consider a form of multi-issue negotiation where a shop negotiates both the contents and the price of bundles of goods with his customers. We present some key insights about, as well as a technique for, locating mutually beneficial alternatives to the bundle currently under negotiation. The essence of our approach lies in combining historical sales data, condensed into aggregate knowledge, with current data about the ongoing negotiation process, to exploit these insights. In particular, when negotiating a given bundle of goods with a customer, the shop analyzes the sequence of the customer's offers to determine the progress in the negotiation process. In addition, it uses aggregate knowledge concerning customers' valuations of goods in general. We show how the shop can use these two sources of data to locate promising alternatives to the current bundle. When the current negotiation's progress slows down, the shop may suggest the most promising of those alternatives and, depending on the customer's response, continue negotiating about the alternative bundle, or propose another alternative. Extensive computer simulation experiments show that our approach increases the speed with which deals are reached, as well as the number and quality of the deals reached, as compared to a benchmark. In addition, we show that the performance of our system is robust to a variety of changes in the negotiation strategies employed by the customers.

Negotiating over Bundles and Prices
Using Aggregate Knowledge*

D.J.A. Somefun[1], T.B. Klos[1], and J.A. La Poutré[1,2]

[1] *CWI, Kruislaan 413, NL-1098 SJ Amsterdam, The Netherlands,*
{koye,tomas,hlp}@cwi.nl
[2] *Faculty of Technology Management, University of Eindhoven, The Netherlands*

Abstract

Combining two or more items and selling them as one good, a practice called bundling, can be a very effective strategy for reducing the costs of producing, marketing, and selling goods. In this paper, we consider a form of multi-issue negotiation where a shop negotiates both the contents and the price of bundles of goods with his customers. We present some key insights about, as well as a technique for, locating mutually beneficial alternatives to the bundle currently under negotiation. The essence of our approach lies in combining historical sales data, condensed into aggregate knowledge, with current data about the ongoing negotiation process, to exploit these insights.

In particular, when negotiating a given bundle of goods with a customer, the shop analyzes the sequence of the customer's offers to determine the progress in the negotiation process. In addition, it uses aggregate knowledge concerning customers' valuations of goods in general. We show how the shop can use these two sources of data to locate promising alternatives to the current bundle. When the current negotiation's progress slows down, the shop may suggest the most promising of those alternatives and, depending on the customer's response, continue negotiating about the alternative bundle, or propose another alternative.

Extensive computer simulation experiments show that our approach increases the speed with which deals are reached, as well as the number and quality of the deals reached, as compared to a benchmark. In addition, we show that the performance of our system is robust to a variety of changes in the negotiation strategies employed by the customers.

---

*This is an extended version of our paper with the same title that was accepted for the 5th *International Conference on Electronic Commerce and Web Technologies* (EC-Web 2004), August 30–September 3, 2004 in Zaragoza, Spain.

1 Introduction

Combining two or more items and selling them as one good, a practice called bundling, can be a very effective strategy for reducing the costs of producing, marketing, and selling products [1]. In addition, and maybe more importantly, bundling can stimulate demand for (other) goods or services [2, 3, 4]. To stimulate demand by offering bundles of goods, requires knowledge of customer preferences. Traditionally, firms first acquire such aggregated knowledge about customer preferences, for example through market research or sales data, and then use this knowledge to determine which bundle-price combinations they should offer. Especially for online shops, an appealing alternative approach would be to *negotiate* bundle-price combinations with customers:[1] in that case, aggregate knowledge can be used to facilitate an interactive search for the desired bundle and price. Due to the inherently interactive characteristics of negotiation, such an approach can very effectively adapt the configuration of a bundle to the preferences of a customer. A high degree of bundle customization can increase customer satisfaction, which may lead to an increase in the demand for future goods or services.

In this paper, we present an approach that allows a shop to make use of aggregate knowledge about customer preferences. Our procedure uses aggregate knowledge about *many* customers in bilateral negotiations of bundle-price combinations with *individual* customers. Negotiating concerns selecting a subset from a collection of goods or services, viz. the bundle, together with a price for that bundle. Thus, the bundle configuration—an array of bits, representing the presence or absence of each of the shop's goods and services in the bundle— together with a price for the bundle, form the negotiation issues. In theory, this is just an instance of multi-issue negotiation. Like the work of [7, 8, 9, 10], our approach tries to benefit from the so-called win-win opportunities offered by multi-issue negotiation, by finding mutually beneficial alternative bundles during negotiations. The novelty of the approach, however, lies in the use of aggregate knowledge of customer preferences. We show that a bundle with the highest 'gains from trade' Pareto-dominates all other bundles within a certain collection of bundles.[2,3] Based on this important insight, we develop an approach for combining aggregate knowledge of customer preferences with data about the ongoing negotiation process, to find alternative bundles that are likely to lead to the highest Pareto improvements. Computer simulations show how, for various types of customers—with distinct negotiation heuristics—our procedure increases the speed with which deals are reached, as well as the number

---

[1] See [5, 6] for other online bundling approaches.

[2] The gains from trade for a bundle are equal to the customer's 'valuation' of the bundle minus the seller's valuation of the bundle, which is his (minimum) price. The term refers to the gains obtained from trading the bundle for the price: both sides benefit from trading whenever the customer's valuation is higher than the seller's [11].

[3] An offer constitutes a Pareto improvement compared to another offer whenever it makes one bargainer better off without making the other worse off. Thus a bundle b' Pareto-dominates another bundle b whenever switching from bundle b to that bundle b' results in a Pareto improvement [11].

and the Pareto efficiency of the deals reached.

In the context of bundling, the distinction between complementary and non-complementary goods is important. In the case of complementary goods, the valuation of a bundle is higher than the sum of the valuations of the individual goods. Bundling complementary goods clearly results in higher gains from trade and is therefore mutually beneficial. Firms usually know beforehand which goods do and which do not complement one another (e.g., bicycle and bicycle tier, copier and toner, etc.), and for complementary goods they will make straightforward bundling decisions accordingly. For an important subclass of non-complementary goods—so-called additively separable goods—the bundle valuation is obtained by just adding up the individual valuations. In that case, the way in which bundling may be advantageous is less clear: it depends on the shop's and the customer's valuations. The shop may enjoy economies of scale or scope in the production or distribution of goods, while the customer's valuations for different goods may be correlated (see [12] and the references cited therein). It is this setting of additively separable goods that we will focus on in the current paper. A cable provider with TV, phone, internet, and pay TV services offers an example of additively separable goods. Another example is the common practice of mobile phone operators in Europe to offer prepaid subscriptions for fixed amounts of SMS, long-distance minutes, international calls, and other services. A final example is an online news provider selling various news items in relatively independent categories such as sports, finance, culture, and science.

For numerous real world applications—like the ones mentioned above—the number of individual goods to be bundled, n, is relatively small. In this paper we will also consider only small values of n (say $n \leq 10$), for which aggregate knowledge still greatly facilitates the process of finding attractive alternative bundles during a negotiation process. For example, with $n = 10$, there are $2^n - 1 = 1023$ possible bundle configurations, so facilitating the search process among all those bundles is highly valuable. On the other hand, obtaining the desired aggregate knowledge is still manageable, since with additively separable goods this only requires information about customers' valuations for individual goods, and not for all possible bundles.

The next section provides a high-level overview of the interaction model. In Section 3 we introduce relatively mild conditions on the seller's and his customers' preferences. Based on these conditions, Section 4 develops a procedure for finding the most promising alternative bundles. In order to test the performance of our system, we used it in interactions with simulated customers. Section 5 presents our computer experiments and discusses the results. Conclusions follow in Section 6.

2 Overview

This section gives an overview of the interaction between the shop and the customer, as they try to negotiate an agreement about the price and the com-

position of a bundle of goods. The shop sells a total of n goods, each of which may be either absent or present in a bundle, so that there are $2^n - 1$ distinct bundle-configurations containing at least 1 good. In the current paper, we use $n = 10$. A negotiation concerns a bundle (configuration), together with a price for that bundle, and it is conducted in an alternating exchange of offers and counter offers [13], typically initiated by the customer. An example of such a practice may involve the sales of bundles of news items in categories like politics, finance, economics, sports, arts, etc.

We develop a procedure that a seller can use to find mutually beneficial alternative bundles during the negotiation about a given bundle, so that alternative bundles may be recommended whenever the negotiation about the given bundle stalls. Specifically, the procedure finds *Pareto improvements* by changing the bundle content.[3] It uses information specific to the current negotiation process as well as aggregate knowledge.[4] The ongoing negotiation is analyzed to determine *when* an alternative bundle is needed, and both the ongoing negotiation process and the aggregate knowledge are used to assess *which* bundle to recommend.

A customer can explicitly reject a suggested bundle by specifying a counter offer with a different bundle content (e.g., the previous one), and she can implicitly reject a suggested bundle by offering a low price for it. In the current paper, only implicit rejection is allowed: customers only specify the bundle content for the opening offer, and thereafter only the seller can change the bundle content of an offer. This is to ease the description of our model and solutions. The possibility for customers to explicitly reject or change the bundle content can be easily incorporated in our model and solutions, however.

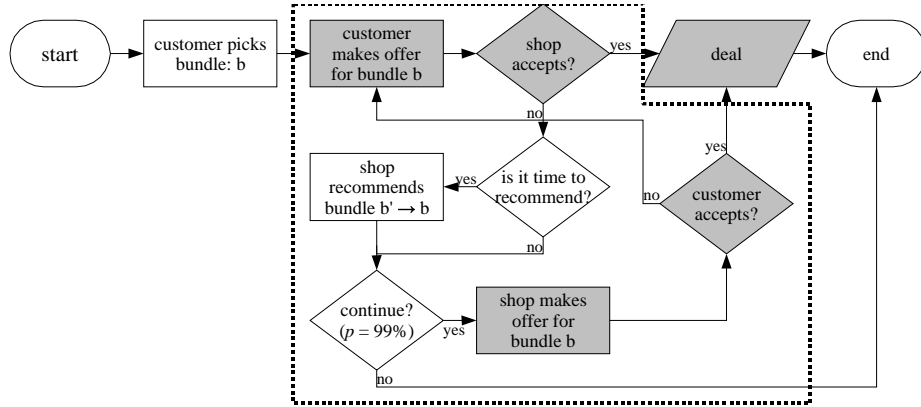Figure 1 provides a high-level overview of the interaction between a shop

Figure 1: A flowchart describing the integration of recommendation in a shop and a customer's alternating exchange of offers and counter offers.

---

[4]Aggregate knowledge may be obtained by analyzing past sales data or provided by marketing experts.

and a customer. The shaded elements are part of the actual negotiation—the exchange of offers. The process starts with the customer indicating her interests, by specifying the bundle they will initially negotiate about. After that, they enter into a loop (indicated by the dotted line) which ends only when a deal is made, or with a 1% exogenous probability. (We do not model the bargainers' impatience explicitly; therefore we need an exogenous stopping condition, which specifies the chance of bargaining breakdown.) In the loop, the customer makes an offer for the current bundle b, indicating the price she wants to pay for it. The shop responds to this offer either by accepting it, or by considering a recommendation. In any case, conditional upon the 99% continuation probability, the shop also makes an offer, either for the current bundle b or for a recommended bundle b' (which then becomes the current bundle).

In the model, the *valuations* of the customers and the seller are expressed as *monetary values*. The *utilities* of deals are expressed as strictly monotonic one-dimensional transformations of valuations. In the simplest form, this would be the difference between the valuation of the bundle and the negotiated price. The agents are interested in obtaining a deal with optimal utility ("net monetary value"). See Section 3 for details.

3 Preference Model

3.1 Informal Discussion

The essence of our model of valuations and preferences lies in the assumption that customers and seller order bundles based on their *net monetary value*; the bundle with the highest net monetary value is the most preferred bundle. A customer's net monetary value of a bundle is equal to the customer's valuation of the bundle (expressed in money) minus the bundle price; the seller's net monetary value is equal to the bundle price minus the seller's bundle valuation (also expressed in money).

Given the above assumption and the assumption that a customer wants to buy at most one bundle (within a given time period), Section 3.2 shows that any deal involving the bundle with the highest *gains from trade* is Pareto efficient. We can now specify which is the best bundle for the seller to advise: faced with the problem of recommending one bundle out of a collection of bundles, the "best" bundle to recommend is the bundle with the highest expected gains from trade; this bundle Pareto dominates all other bundles. (Section 3.2 can be skipped upon first reading.)

3.2 Formal Discussion

Before being able to more formally state the results, some notation is necessary. Let $N \subset \mathbb{N}$, with $n = |N|$, denote the collection of n individual goods and 2^N denotes the power set of N (i.e., the collection of all subsets of n), then

$B = 2^N \setminus \{\emptyset\}$ denotes the collection of all possible bundles. Furthermore, let $P = \mathbb{R}$ denote the collection of all possible bundle prices.[5] The customer and the seller attach the monetary values of $v_c(b)$ and $v_s(b)$, respectively, to a bundle $b \in B$ (with $v_c(b), v_s(b) \in P$). The function $x_j : B \times P \mapsto \mathbb{R}$ with $j \in \{c, s\}$ denotes the *net* monetary value for bundle b and bundle price p: $x_c(b, p) = v_c(b) - p$ and $x_s(b, p) = p - v_s(b)$ denote the customer's and the seller's net monetary values, respectively.

We assume that the customer's and the seller's utility for consuming bundle b for a price p, denoted by $u_j(b, p)$ with $j \in \{c, s\}$, can be expressed as the composition function $g_j \circ x_j(b, p)$ with $j \in \{c, s\}$ and $g_j : \mathbb{R} \mapsto \mathbb{R}$. For g_j we assume that $\frac{dg_j(x)}{dx} > 0$ for all $x \in \mathbb{R}$ and $j \in \{c, s\}$.

Given the customer's and seller's monetary values, we define a useful subset B^* of B as follows: $B^* \equiv \arg\max_{b \in B}(v_c(b) - v_s(b))$, that is, B^* represents the collection of bundles with the highest gains from trade. We are now ready to introduce the following proposition.

Proposition 1 *A deal (b, p) with $b \in B$ and $p \in P$ is Pareto efficient if and only if $b \in B^*$.*

Remark 1 *A deal (b, p) is Pareto efficient if there is no (b', p') such that $u_j(b, p) \le u_j(b', p')$ for all $j \in \{c, s\}$ and the inequality is strict for at least one j.*

Proposition 1 means that a deal is Pareto efficient if and only if it entails a bundle with the highest gains from trade. For the proof of this proposition the following lemma is very useful.

Lemma 1 *For any two deals (b^*, p^*) and (b, p) with $p^*, p \in P$, $b^* \in B^*$, and $b \in B \setminus B^*$ we have $x_c(b, p) < x_c(b^*, p^*)$ or $x_s(b, p) < x_s(b^*, p^*)$.*

PROOF. We prove the above lemma by contradiction. Suppose that for any $b^* \in B^*$ and $b \in B \setminus B^*$ we have $x_c(b, p) \ge x_c(b^*, p^*)$ and $x_s(b, p) \ge x_s(b^*, p^*)$. A necessary conditions for this to hold is that $v_c(b) - v_s(b) \ge v_c(b^*) - v_s(b^*)$. However, $b^* \in B^*$ and $b \in B \setminus B^*$ means, by definition of B^*, that $v_c(b) - v_s(b) < v_c(b^*) - v_s(b^*)$.

QED

We are now ready to prove proposition 1.
PROOF.

1. (If) Pick any $j \in \{c, s\}$. Suppose that j's position improves by moving from any deal (b, p) with $b \in B^*$ to (b', p'), that is, $u_j(b, p) < u_j(b', p')$. It then suffices to show that the opponent denoted by j' will always be made

worse off, that is, $u_{j'}(b, p) > u_{j'}(b', p')$. From the properties of g_j and $g_{j'}$ it follows that a bargainer's position improves/worsens whenever the net monetary value increases/decreases. Since j's position improves, it follows from lemma 1 that j' is made worse off whenever $b \in B \setminus B^*$. Moreover, if $b^*, b \in B^*$ then the gains from trade remain unchanged, hence j' is made worse off.

2. (Only if) We will prove this part by contradiction. Suppose that $b \notin B^*$ with the price being any $p \in P$. Pick any $b' \in B^*$ and set the bundle price to $p' = p + v_s(b') - v_s(b)$, so that $p' - v_s(b') = p - v_s(b)$. It follows from $p \in P$ that $p' \in P$ (recall that $P = \mathbb{R}$)[6] and the properties of g_s that the seller is indifferent between the deals (b, p) and (b', p'). Also, it follows from lemma 1 and the properties of g_c that the customer is made better off. That is, any $b' \in B^*$ Pareto dominates $b \notin B^*$. Thus $b \notin B^*$ cannot be a Pareto efficient solution.

QED

4 Recommendation Mechanism

The idea is to develop a mechanism for a seller to find Pareto improvements by changing the bundle content during a negotiation. The mechanism we propose contains two subprocedures. The first procedure—telling the shop when to recommend—monitors the negotiation process and determines when to pass control to the second procedure, which generates recommendations based on aggregate knowledge and the ongoing negotiation process. Figure 1 already showed the interaction between these two procedures; they are discussed in more detail in Sections 4.1 and 4.2, respectively.

4.1 Deciding *When* to Recommend

The shop needs a procedure for deciding when he should recommend negotiating about a different bundle. The obvious input for this decision is the progress of the current negotiation process, which can be described as a sequence of offers and counteroffers. An offer O contains a bundle definition and a price: $O = (b, p)$ with $b \in B$ and $p \in P$. (B and P denote the collections of all possible bundles and prices, respectively.) Let $h = (O(1), O(2), \ldots, O(k))$ denote a finite history of offers (k is a natural number), where $O(i + 1)$ is the counter offer for $O(i)$, for all $i < k$. Furthermore, let H denote the universe of all possible finite offer

---

[6] If we choose to a priori rule out $p < 0$ and $v_j(b) < 0$ (for $j \in \{c, s\}$ and all $b \in B$), then $p \geq v_s(b)$ should hold because otherwise the seller will not be willing to sell the bundle in the first place. Consequently, $p' \in P$ still holds.

7

sequences (thus $h \in H$). The problem of when to advise can now be specified as the mapping $f : H \mapsto \{\text{yes}, \text{no}\}$, where "yes" means: recommend a new bundle.

We construct a heuristic for f based on the assumption that there is a probability of not reaching a deal with a customer (e.g., a break off, endless repetition, or deadline): the longer the negotiation is expected to take, the less likely a deal is expected to become. Furthermore, as a deal becomes less likely, the incentive for the shop to recommend negotiation about an alternative deal should increase. Given the seller's bargaining strategy then, our heuristic extrapolates the time the current negotiation process will need to reach a deal, from the pace with which the customer is currently giving in. More precisely, if we let $O = (b, p)$ and $O' = (b, p')$ denote the customer's current and previous offers for bundle b, then Δt, the predicted remaining number of negotiation rounds needed to reach a deal, is defined as follows:

$$\Delta t = \frac{v_s(b) - p'}{p - p'}, \tag{1}$$

where $v_s(b)$ denotes the seller's monetary value for bundle b. The higher Δt, the higher the likelihood of a recommendation. Specifically, the probability of a recommendation depends on Δt as follows:

$$pr_{\text{recommendation}} = 1 - \exp(-0.25 \Delta t),$$

which means that the probability that the shop recommends an alternative bundle approaches 1 as Δt increases.

4.2   Deciding *What* to Recommend

Our mechanism combines aggregate knowledge (obtained from the analysis of sales data and (anonymous) data on previous negotiations, market research, or expert knowledge) with data on the ongoing bargain process, to recommend bundles to customers while negotiating with them. Suppose, for example, that a customer offers to buy a bundle b at a price p. When a recommendation is needed (see Section 4.1) the idea is to select from within the "neighborhood" of bundle b, the bundle b' that maximizes $E[v_c(b') - v_s(b')|v_c(b) \geq p]$, the expected gains from trade—given that a customer is willing to pay at least the price p for bundle b. Since the seller knows its own monetary value for bundle b', $v_s(b')$, the aim is really to maximize $E[v_c(b')|v_c(b) \geq p] - v_s(b')$. The difficulty here lies in estimating the customer's expected valuation of the bundle:

$$E[v_c(b')|v_c(b) \geq p] = \sum_{i \in P} i \cdot pr(v_c(b') = i|v_c(b) \geq p), \tag{2}$$

where $pr(v_c(b') = i|v_c(b) \geq p)$ denotes the probability that the customer's valuation for bundle b is equal to i, given that she is willing to pay at least p for bundle b. (To simplify notation we will write $E[v_c(b')|b]$ instead of $E[v_c(b')|v_c(b) \geq p]$.)

Aggregate knowledge can provide an estimation of $E[v_c(b')|b]$. Given that the shop sells n individual goods, there are $2^n - 1$ possible bundles containing at

least 1 good. To determine $E[v_c(b')|b]$ for all possible bundle pairs, requires—worst case—an order of $(2^n)^2$ estimations. When the customer's valuation for a bundle is just the sum of her valuations of the individual goods comprising the bundle, as it is assumed in the current paper, this complexity is reduced significantly. Given that a customer's valuation of bundle b, $v_c(b)$, is simply the sum of the valuations of the goods comprising bundle b,

$$E[v_c(b')|b] = \sum_{i \in b'} E[v_c(i)|b]. \tag{3}$$

This requires at worse "only" $n \cdot 2^n$ estimations of conditional expectations, which is manageable for $n = 10$, as in the current paper. For larger values of n it quickly becomes infeasible to estimate these expectations. In that case, it is necessary to approximate most of the conditional probabilities based on a limited number of explicit estimations. In this paper the focus lies on applications where $n \cdot 2^n$ is still a manageable number, so discussing approximation techniques is beyond the scope of this paper.

4.2.1 Generating Recommendations

A customer initiates the negotiation process by proposing an initial bundle and offering an opening price: let $O(0) = (b, p)$ denote the customer's opening offer (with $b \in B$ and $p \in P$). The shop stores the bundle proposed by the customer as the customer's "interest bundle," in the neighborhood of which he searches for promising alternative bundles to recommend if, at any time, the shop decides he should make a recommendation (see Section 4.1). This neighborhood of bundle b, $Ng(b)$, is defined as follows.

$$Ng(b) \equiv \{b' \in B : (b' \subset b \text{ and } |b'| + 1 = |b|) \text{ or } (b' \supset b \text{ and } |b'| - 1 = |b|)\}, \tag{4}$$

In other words, $Ng(b)$ contains the bundles which, in binary representation, have a Hamming distance to b of 1.[7] The advantage of advising bundles within the neighborhood of b is that the advice is less likely to appear haphazard.

Having defined a bundle's neighborhood, let the ordered set A denote the so-called "recommendation set," obtained by ordering the neighborhood $Ng(b)$ on the basis of the estimated expected gains from trade of all the bundles b' in bundle b's neighborhood, $\hat{E}[v_c(b')|b] - v_s(b')$, where $\hat{E}$ denotes the estimation of E. Let $\bar{A}$ denote the unordered set of previously proposed bundles.

To recommend a bundle b_k (the kth recommendation, with $k \geq 1$), our mechanism removes the first bundle from A, adds a price to it and proposes it as part of the shop's next offer, and then adds it to $\bar{A}$. Depending on the customer's counter offer for bundle b_k, a number of additional bundles may be added to the advice set: if the customer's response is very promising (to be defined below), bundle b_k will be taken as the customer's *new* interest bundle

---

[7]Remember that each bundle can be represented as a string containing n bits indicating the presence or absence in the bundle, of each of the shop's n goods.

(in the neighborhood of which the search continues), and the bundles in the neighborhood of b_k are added to A.

Intuitively, in order to determine how promising a bundle b_k is in terms of its potential for generating gains from trade, the shop needs to compare the net monetary value of the new bundle b_k, with the current highest net monetary value among all previous bundles. A relatively large improvement over the currently most promising bundle—the shop's estimation of the customer's 'interest bundle'—will cause the shop to update his estimation of the customer's 'interest bundle'. However, because the shop does not know the customer's valuation for a bundle, he simply compares offered and asked prices for bundles.

To specify this in more detail, let O_t^c denote the sequence of offers placed by the customer up until time t, and let $max(O_t^c)$ specify the customer's past offer with the highest difference between the customer's offered and the shop's asked price. Then the shop will determine the impact of recommending bundle b_k by comparing the customer's current offer for bundle b_k, $O(t+1)$ with that of offer $max(O_t^c)$, from the perspective of his own bid for bundle b_k. For this purpose, the shop uses the function $sign : \mathbb{R} \times \mathbb{R} \mapsto \{0, 1, 2\}$. If we let $max(O_t^c) = (b', p_c')$, the customer's current offer $O(t+1) = (b, p_c)$, and the shop's corresponding bids for bundles b and b' be $O(b', p_s')$ and $O(b, p_s)$, then

$$sign_{b,b'}(p, p') = \begin{cases} 2 & \text{if } \frac{p_c - p_s}{p_c' - p_s'} > (1 + \text{threshold}) \\ 1 & \text{if } 1 \leq \frac{p_c - p_s}{p_c' - p_s'} \leq (1 + \text{threshold}) \\ 0 & \text{otherwise} \end{cases} . \qquad (5)$$

If $sign(p, p') = 2$, then the shop's assessment of the customer's interest bundle is updated to be b_k: the customer's response is promising enough to divert the search towards the neighborhood of b_k, and add that neighborhood to A such that the first elements of A all lie in the neighborhood of b_k. That is, the first element of A becomes the bundle $b' \in Ng(b_k)$ with the maximum difference $\hat{E}[v_c(b')|b_k] - v_s(b')$, the second element of A becomes the bundle b'' with the second highest difference $\hat{E}[v_c(b'')|b_k] - v_s(b'')$, and so on. In addition, duplicates are removed from A, as are bundles already present in $\bar{A}$. In case $sign(p, p') = 1$, the customer's response is promising enough to continue negotiating about the current bundle b_k, but not promising enough to change the assessment of the customer's interest bundle, and if $sign(p, p') = 0$, the proposed bundle was not promising at all and the seller will immediately make the next recommendation.

5 Numerical Experiments

In order to test the performance of our proposed mechanism, we implemented it computationally, and tested it against many simulated customers. Valuations for the shop and the customers were drawn from random distributions. First we describe how we handled the approximation process and how we implemented negotiations in the simulation, and then we present our experimental design and simulation results.

5.1 Modeling $E[v_c(b')|b]$

In the experiments we abstract away from actually learning $E[v_c(b')|b]$, for example from sales data. Instead we derive these conditional expectations directly from the way we specified the underlying stochastic process.

As explained earlier, we assume additively separable customer preferences. To compute the customer's valuation for a bundle b we simply add up her valuations for the individual goods that constitute the bundle:

$$v_c(b) = \sum_{i \in b} v_c(i). \tag{6}$$

Let N denote the collection of all the individual goods from which bundles are constructed, with $|N| = n$. Suppose we specify the joint probability density function of the customer's valuations for the individual goods, $pr(z_1, \ldots, z_n)$, as an n-variate normal distribution. Let the vector $\mu = (\mu_1, \cdots, \mu_n)$ denote the mean of the distribution and let the matrix $\mathbf{\Sigma} = [\sigma_{ij}]$ denote the covariance matrix. Then $pr(z_1, \ldots, z_n) \sim N[\mu, \mathbf{\Sigma}]$.

The joint probability mass function of all bundle valuations, $pr(z_1, \ldots, z_{2^n})$, is simply a linear transformation of $pr(z_1, \ldots, z_n)$. Since a linear transformation of a multivariate normal distribution is also a multivariate normal distribution [14], we have

$$pr(z_1, \ldots, z_{2^n}) \sim N[\mathbf{T}\mu, \mathbf{T}\mathbf{\Sigma}\mathbf{T}'], \tag{7}$$

where the matrix $\mathbf{T}$ specifies the linear transformation (a row in $\mathbf{T}$ specifies a bundle in binary representation). Given $N[\mathbf{T}\mu, \mathbf{T}\mathbf{\Sigma}\mathbf{T}']$ we can derive the value of $E[v_c(b')|b]$ for any bundle pair. This approach implies that we simply hand the shop the distributions underlying customers' valuations.

5.2 Modeling Negotiations

Besides setting customer preferences it is necessary to specify how the shop and the customer actually negotiate. To allow initiation of the negotiation process by the customer, we assume that the customer starts negotiating about an initial bundle b_{init}. In order to give the shop some room for improvement, we initialize the customer's initial bundle as the bundle containing all the goods for which her valuation is lower than her average valuation across all goods. Although this seems to make it very easy for the shop to make an improvement, bare in mind that performance refers to gains from trade, which depends on both the customer's and the shop's valuations. Besides, we measure performance relative to this starting point in our experiments.

Time-dependent Strategy For the customer (shop), the time-dependent bidding strategy is monotonically increasing (decreasing) in both the number of bidding rounds (t) and her (his) valuation. In particular, a bidding strategy is characterized by the gap the customer leaves between her initial offer and her valuation, and by the speed with which she closes this gap. The gap is

specified as a fraction of the bundle valuation and it decreases over time as $gap(t) = gap_{init} \cdot \exp(-\delta t)$, so over time, she approaches the valuation of the bundle she is currently negotiating about. Note that changes in the gap are time-dependent, but not bundle-dependent! This strategy is therefore called "time-dependent-fraction" (TDF).[8] The initial gap, gap_{init}, and δ are drawn randomly from a uniform distribution between $[0, 0.5]$ and $[0.1, 0.4]$, respectively. Almost the same holds for the shop's bidding strategy, *mutatis mutandis*. Since δ already fluctuates for the customer's strategy we do, however, set $\delta = 0.1$ for the shop, in order to reduce the number of jointly fluctuating parameters somewhat. Summing up, the customer (shop) starts her bidding for a bundle at a randomly chosen point between her valuation and (one and a) half her valuation, and gradually approaches her valuation with her bids.

Tit-for-Tat Strategy The time-dependent strategy described above generates bids irrespective of what the opponent does. As an example of a strategy that responds to the opponent, we implemented a variant of tit-for-tat (TFT) [15]. The initial 'move' is already specified by gap_{init} like in the TDF-strategy. If in subsequent moves the utility level of the opponent offer improves, then a fraction δ of that amount is conceded by the bargainer. Note that it is the increment in the utility level as perceived by the bargainer. Furthermore, this perceived utility improvement can also be negative. To make the bidding behavior less chaotic, no negative concessions are made. That is, we used a so-called monotone version called tit-for-tat-monotone-fraction (TFTMF) which can never generate a bid with a worse utility than the previous bid.

5.3   Results

A Benchmark In order to assess the relative performance of the system we conducted the same series of experiments (see below) with a benchmark procedure, which randomly recommends a bundle from the current bundle's neighborhood. That is, the benchmark does not base the order in which it advises the next bundle on the estimated expected gains from trade like our system does.

In our experiments there are 10 individual goods. We generate the means of the underlying probability density function $pr(z_1, \ldots, z_{10})$ by randomly sampling numbers between 40 and 250 without repetition. To exclude negative numbers we then randomly draw the variance of the 10 marginal distributions from a uniform distribution such that the probability of a negative valuation for an individual good is negligible, viz. at most 0.0003. In order to ensure sufficient differences in valuations between customers, however, we fix the correlation matrix (but not the covariance matrix). To test the robustness of our procedure to quantitative changes in the underlying distributions we conducted a series of experiments with 100 different distributions $pr(z_1, \ldots, z_{10})$. For each distribution we tested the influence on the system's performance of changes in

---

[8] We originally implemented a time-dependent strategy that decreases an *absolute* gap over time, but the results from that strategy were qualitatively similar to the fraction strategy's results, so we describe only the latter as they are more intuitive.

the ease with which the shop updates his estimation of the customer's "interest" (see the discussion of the *sign* function in Section 4.2.1). The "threshold" used in the *sign* function captures this sensitivity; we experimented with 11 values between 0 and 0.5, with stepsize 0.05. For each of these settings we simulated negotiations between the shop, with randomly drawn valuations, which were kept constant across negotiations with 100 customers, each with her valuations drawn randomly from the particular distribution used. The values in the graphs are averages across 100 customers per distribution, and across 100 different distributions.

The shop's bundle valuations are not additively separable, due to the follwoing nonlinear pricing strategy. Bundles with a higher than average expected customer valuation—compared to bundle containing the same number of individual goods—are relatively expensive. That is, for expensive bundle it is less likely that customers are actually willing to buy the offered good. Similarly, bundles with a lower than expected valuation are relatively inexpensive (compared to bundles of the same size).

Concerning the legends throughout, the graphs for 'rounds' give the number of rounds it took, if a deal was reached, to reach that deal, and the graph for 'deals' shows the total number of deals reached in negotiations with the 100 customers. The '$(s-b)$' graphs show the difference between our system's and the benchmark's performance. For an explanation of 'perc' and 'relP,' consider that given the shop's and the customer's valuations and the configurations of all possible bundles, certain maximum and minimum gains from trade are attainable. The graph for 'perc' shows the gains from trade of the final bundle, as a percentage of the difference between the maximum and minimum gains from trade, and 'relP' shows the gains from trade of the final bundle, relative to the difference between the maximum gains from trade and the gains from trade of the initial bundle (the starting point of the negotiation). As a final remark, bear in mind that the lines in the graphs are included to allow easy identification of patterns, but only the symbols represent actual values measured in the simulation.

Figure 2 reports the results of three series of experiments (see the caption) where we vary the bargaining strategy of the customers. For low thresholds, our system generates roughly 70% of the maximum gains from trade and roughly 60% of the gains of trade attainable given the initial bundle. Its performance in both cases is roughly 20% better than the benchmark. Additionally, more deals are reached and it requires less time to reach these deals than the benchmark.

Initially, larger threshold values improve the performance of the benchmark. As it searches the current interest bundle's neighborhood in a random order, increasing the threshold makes it change the interest bundle less haphazardly, forcing it to focus longer on a particular interest bundle's neighborhood, which may contain better alternatives than the ones it encounters initially. Usually, these better alternatives do exist, and they turn up in such a slower, and more prudent search, leading to more deals as the threshold increases. As the threshold increases further, however, the interest bundle is not updated as easily anymore, thereby blocking the search in regions in the neighborhood of promising
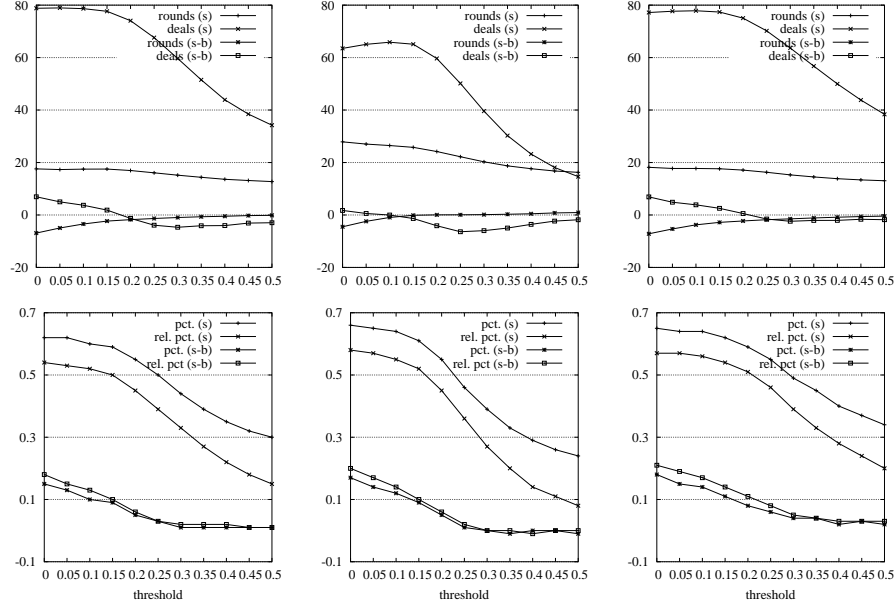
13

Figure 2: Results for our system ('s'), when the shop uses the TDF strategy with $\delta = 0.1$, and the customers use either the TDF strategy with random δ (on the left), or the TFTMF strategy with random δ (in the middle) or with $\delta = 1$ (on the right), as described in section 5.2. The difference with the benchmark is indicated by the '$(s - b)$'-graphs.

bundles. This effect is also visible in the results of our system, which zeroes in on the most promising bundle more quickly. For larger threshold values this effect becomes increasingly more important. Consequently, the difference in performance reduces for larger values; ultimately, for a threshold of 0.5, there is virtually no distinction in performance.

6 Conclusions and Future Work

We consider the problem of negotiating over both bundle contents and price, which permits a high degree of personalization of bundles to the preferences of customers. We develop a procedure for a seller to search for Pareto improvements in bundle contents, while negotiating about the price of bundles. Computer experiments show how this procedure increases both the speed with which agreements are reached, as well as the number and quality of agreements reached.

In the current paper, we have only considered additively separable consumer preferences. The most important issue currently under investigation is the extension of our procedure to cases involving non-linear preferences.

Another issue concerns the distribution of valuations we used. Such a distribution generates preferences for customers, which may make them buy the shop's (bundles of) goods. Data about such sales, in turn, enable the shop to estimate the distribution underlying his customers' preferences. In the current paper we provided the shop with the aggregate knowledge required for our procedure directly. Even without modeling this process explicitely, we are interested in providing the shop with only an estimate of the real distribution and in testing the robustness of our procedure to variations in the accuracy of the estimate.

References

[1] Baumol, W., Willig, R., Panzar, J.: Contestable Markets and the Theory of Industry Structure. Dryden Press (1987)

[2] Stigler, G.J.: United states v. Loew's inc.: A note on block-booking. Supreme Court Review (1963) 152–157

[3] Schamlensee, R.L.: Gaussian demand and commodity bundling. Journal of Business **57** (1984) S211–S230

[4] Bakos, Y., Brynjolfsson, E.: Bundling information goods: Pricing, profits and efficiency. Management Science **45** (1999)

[5] Kephart, J.O., Fay, S.A.: Competitive bundling of categorized information goods. In: Proceedings of ACM EC'00. (2000)

[6] Somefun, D.J.A., La Poutré, J.A.: Bundling and pricing for information brokerage: Customer satisfaction as a means to profit optimization. In: Proceedings of Web Intelligence 2003 (WI2003), IEEE Computer Society (2003) 182–189

[7] Klein, M., Faratin, P., Sayama, H., Bar-Yam, Y.: Negotiating complex contracts. Group Decision and Negotiation **12** (2003) 111–125

[8] Faratin, P., Sierra, C., Jennings, N.R.: Using similarity criteria to make issue trade-offs. Journal of Artificial Intelligence **142** (2003) 205–237

[9] Ehtamo, H., Hämäläinen, R.: Interactive multiple-criteria methods for reaching pareto optimal agreements in negotiation. Group Decision and Negotiation **10** (2001) 475–491

[10] Somefun, D.J.A., Gerding, E., Bohte, S., La Poutré, J.A.: Automated negotiation and bundling of information goods. In: Proceedings of Agent-Mediated Electronic Commerce V (AMEC'03). (2003)

[11] Mas-Collel, A., Whinston, M.D., Green, J.R.: Mircoeconomic Theory. Oxford University Press (1995)

[12] Mankila, M.: Price bundling applied to retail banking. FE Rapport 367, Göteborg University, School of Economics and Commercial Law (1999) online available at "http://swoba.hhs.se/gunwba/".

[13] Rubinstein, A.: Perfect equilibrium in a bargaining model. Econometrica **50** (1982) 97–109

[14] Green, W.H.: Econometric Analysis. Prentice Hall, New Jersey (1993)

[15] Axelrod, R.: The Evolution of Cooperation. Basic Books, New York (1984)