



Centrum voor Wiskunde en Informatica

REPORTRAPPORT

SEN

Software Engineering



Software ENgineering

Bundling and Pricing for Information Brokerage:
Customer Satisfaction as a Means to Profit Optimization

D.J.A. Somefun, J.A. La Poutré

REPORT SEN-E0329 DECEMBER 19, 2003

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)

Copyright © 2004, Stichting Centrum voor Wiskunde en Informatica

P.O. Box 94079, 1090 GB Amsterdam (NL)

Kruislaan 413, 1098 SJ Amsterdam (NL)

Telephone +31 20 592 9333

Telefax +31 20 592 4199

ISSN 1386-369X

Bundling and Pricing for Information Brokerage: Customer Satisfaction as a Means to Profit Optimization

ABSTRACT

Traditionally, the study of on-line dynamic pricing and bundling strategies for information goods is motivated by the value-extracting or profit-generating potential of these strategies. In this paper we discuss the relatively overlooked potential of these strategies to on-line learn more about customers' preferences. Based on this enhanced customer knowledge an information broker can-- by tailoring the brokerage services more to the demand of the various customer groups-- persuade customers to engage in repeated transactions (i.e., generate customer lock-in). To illustrate the discussion, we show by means of a basic consumer model how, with the use of on-line dynamic bundling and pricing algorithms, customer lock-in can occur. The lock-in occurs because the algorithms can both find appropriate prices and (from the customers' perspective) the most interesting bundles. In the conducted computer experiments we use an advanced genetic algorithm with a niching method to learn the most interesting bundles efficiently and effectively.

1998 ACM Computing Classification System: I.2.11

Keywords and Phrases: dynamic bundling; dynamic pricing; information brokerage; recommender systems

Bundling and Pricing for Information Brokerage: Customer Satisfaction as a Means to Profit Optimization.

D.J.A. Somefun* J.A. La Poutré*,†

*CWI, Center for Mathematics and Computer Science
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands.

†School of Technology Management, Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands.

E-mail: {koye, hlp}@cwi.nl.

Abstract

Traditionally, the study of on-line dynamic pricing and bundling strategies for information goods is motivated by the value-extracting or profit-generating potential of these strategies. In this paper we discuss the relatively overlooked potential of these strategies to on-line learn more about customers' preferences. Based on this enhanced customer knowledge an information broker can—by tailoring the brokerage services more to the demand of the various customer groups—persuade customers to engage in repeated transactions (i.e., generate customer lock-in). To illustrate the discussion, we show by means of a basic consumer model how, with the use of on-line dynamic bundling and pricing algorithms, customer lock-in can occur. The lock-in occurs because the algorithms can both find appropriate prices and (from the customers' perspective) the most interesting bundles. In the conducted computer experiments we use an advanced genetic algorithm with a niching method to learn the most interesting bundles efficiently and effectively.

1 Introduction

An information broker gathers/buys information from various sources and sells it to his customers. Traditionally, due to savings in production and transaction costs the information is bundled and sold via subscriptions. Examples of this practice are newspapers, scientific journals, and subscriptions services such as Reuters' Stockmaster service. The coming of the Internet Economy changed the necessity of bundling. Due to the digitization of information goods—pushed by the existence of the Internet Economy—the costs of producing and selling an additional information good has become virtually zero. Consequently, it is currently economically feasible to deliver, for example, indi-

vidual news items or individual stock quotes to consumers and have them pay per item. On the other hand, conditional on consumers' preferences, bundling various information goods might still be useful because it can facilitate the extraction of consumers' valuation (cf. [2]).

In this changed environment, an information broker has to reconsider his business strategy. More specifically, he has to reconsider the way he bundles and positions his information goods. As a consequence of the very low marginal costs various different selling strategies can be profitable. Selling, for example, only individual items, larger bundles, or a few smaller bundles aimed at particular market niches can all be profitable strategies. What the best strategy is depends on consumers' actual preferences.

Current developments in the machine learning community provide an extra incentive for an information broker to reconsider his business strategy. Typically, a seller of information goods does not have complete information about consumers' preferences. By employing machine learning techniques, a seller can learn more about his customers; it is even possible to apply some of these techniques on-line, i.e., the obtained knowledge is directly and automatically put to use. Two important application areas for these on-line techniques are (on-line) dynamic bundling and/or pricing of (information) goods and (automatic) recommender systems. In this paper the focus lies on the economic importance of these on-line learning techniques for information brokerage.

We make the observation that (automatic) recommender systems can persuade customers to engage in repeated transaction. That is, they especially enhance the value-creating potential of an information broker through customer lock-in. Within a competitive setting this is an important virtue because it reduces the competition with rival firms. On the other hand, traditionally the study of on-line dynamic pricing and/or bundling focuses on enhancing, what we call, the value-extracting (or profit-generating)

capacity of an information broker. Within a competitive setting the important drawback of focusing on the value-extracting aspects of these techniques is that they can increase the competition with rival firms. The idea of this paper is to focus on the value-creating potential of on-line dynamic pricing and/or bundling techniques, especially on their potential to create customer lock-in.

Properly working on-line dynamic pricing/bundling algorithms require that, at least implicitly, more is learned about customers' preferences. The idea is that it can be beneficial to make this knowledge explicit.

One possibility of generating customer lock-in (partly) through the use of dynamic pricing/bundling algorithm is by integrating these algorithms into an automatic recommender system. In this paper, we present a framework for such an integrated recommender system. Additionally, we conduct computer experiments to illustrate how, by using on-line dynamic pricing and bundling algorithms, customers' preferences can explicitly be learned. The experiments also show how this enhanced customer knowledge leads to lock-in. In the conducted computer experiments we use an advanced genetic algorithm with a niching method to learn (from the customers' perspective) the most interesting bundles efficiently and effectively.

The remainder of this paper is organized as follows. In Section 2, we discuss the value-extracting potential of on-line dynamic pricing and bundling. In Section 3, we use the framework of [1] to analyze the value creating potential of (automatic) recommender systems for information brokerage. In Section 4, we discuss integrating dynamic bundling/pricing into recommender systems that recommend based on sales statistics. In Section 5 we show how, by using an on-line bundling algorithm, customer knowledge can be enhanced. Conclusions follow in Section 6.

2 On-line Bundling & Pricing

Bundling is the practice of combining two or more items together and selling them as one product. From a cost perspective, it has become economically feasible for an information broker to offer a large variety of different bundles for sale (targeted at various consumer groups).

Typically a seller of information goods has (very) limited information about the distribution underlying customers' valuation. Therefore (optimal) bundling of information goods, generally, requires on-line learning of the appropriate bundle-price combination. In [6] they reduce the complexity of this on-line learning problem somewhat by simplify the problem to essentially a dynamic pricing problem of finding the optimal prices for a particular pricing scheme. (Note that, unlike [6], Section 5 introduces an explicit bundling and pricing framework.)

This bundling approach of [6]— and dynamic pricing in general— entails the on-line automation of the process of refining via trial and error the used pricing schemes. The use of on-line learning algorithms has the advantage of making frequent adjustments of the prices (and/or bundle content) possible at little additional costs. Consequently, past experience is incorporated into the price more frequently. Hence, more exploitation is possible. To summarize: this kind of on-line dynamic pricing and bundling enhance the value-extracting or profit-generating capacity of an information broker.

Dynamic pricing (and/or bundling) may in particular be important in an environment with relatively frequent and unpredictable demand shocks. Additionally, in a setting of imperfect competition, the need to apply dynamic pricing may simple arise because the competition utilizes dynamic pricing algorithms. The use of dynamic pricing algorithms can increase the competition between sellers of information goods [5]. Whether fierce competition actually occurs depends, among other things, on the on-line learning algorithms used [4].

3 Value Creation

3.1 Four Value Drivers

Partly due to the developments in the research on on-line dynamic pricing and bundling it is very tempting to predict the coming of an economists' utopia of frictionless electronic trade, i.e., low search costs, strong price competition, low margins, and low deadweight loss.

Clearly, a frictionless market is no utopia for firms. They can bring friction back to the market place by competing on more than just price characteristics. More specifically, firms should focus on all aspects of the value-creating potential of electronic business.

In [1] they analyze the value drivers in electronic business. The four key value drivers they determine are: (transaction) efficiency, complementarities, lock-in and novelty.

- Transaction efficiency increases when the costs per transaction decrease ("cost" is defined broadly). E.g., low search costs on the internet make it relatively easy for buyers to compare various sellers.
- Goods are complementary whenever bundling them together generates more value than the total value of having each of the goods separately.
- Lock-in refers to the ability to persuade customers to engage in repeated transactions. The two main components of lock-in are switching costs and positive network externalities. On-line vendors can, for instance, introduce switching costs by customizing their web

site. Network externalities occur in the context of electronic business when the value created for customers increases with the size of the customer base (i.e., the number of customers increases).

- Novelty involves the introduction of new products, processes, or services on the internet.

The first three value drivers provide a good framework for analyzing how an information broker can generate value. Clearly, novelty is also important, but too case specific for the purpose of this paper.

Traditionally, the study of on-line dynamic pricing and bundling strategies for information goods is motivated by the value-extracting or profit-generating potential of these strategies (see Section 2). The kind of dynamic pricing and bundling discussed in Section 2 may, however, also result in value creation through (transaction) efficiency and complementarities. For information brokerage, another more obvious way of creating value is to recommend information to customers. In Sections 4 and Section 5, we will show how dynamic pricing and bundling of information may result in value creation through customer lock-in. We will, however, first briefly discuss the recommendation of information and how that can result in value creation.

Recommending information to customers could eventually reduce their search costs for obtaining the desired information. These search costs can be significant if, for example, the “market value” of the information is low, the information is hard to categorize in advance, and/or at the same time a lot of seemingly similar information is being offered for sale. We will focus on (automatic) recommender systems which are machine learning systems specialized to recommend products in (electronic) commerce applications (cf. [11]). For an information broker, in particular, automatic recommender systems are interesting. (Henceforth we will drop the adjective automatic.) They allow for a high level of personalized recommendation at little cost relative to the value of the offered information goods. In the next subsection we will discuss recommender systems and how they can result in value creation, especially customer lock-in.

3.2 Recommender Systems

The heart of an information brokerage recommender system should be an information filtering system. An information filtering system divides a large-volume data stream into substreams. We can roughly distinguish between three types of information brokerage (recommender) systems. (Henceforth we will drop the adjective recommender.)

1. *Collective system.* The system keeps track of so called stereotypes. It composes and updates the stereotypes

based on *anonymous* data, i.e., data from which it is not possible to derive individual user history (e.g., aggregate sales statistics). A stereotype roughly coincides with the behavior and preferences of certain groups of customers. The system uses the developed stereotypes as a basis for its recommendations.

2. *The personalized system.* The system keeps track of (individual) user profiles. It composes and updates a profile based on the past behavior (and provided feedback) of the user.
3. *The hybrid system.* This system combines (1) and (2). The method of collaborative (or social) filtering underlies the hybrid system. It develops and keeps track of both individual user profiles and stereotypes.

By reducing the search cost of customers, all of the above three types of information brokerage systems create value via the value driver (transaction) efficiency. Additionally, all three generate value through customer lock-in. (They, however, differ in the way they create customer lock-in.)

From the consumers perspective the hybrid system has the important drawback that it invades customers’ privacy. Moreover it uses this private information to advice other customers. In the extreme case, the system could implicitly use a particular customer’s expertise of filtering out the right information to advice others. Consider, for instance, the case of an information broker of financial news. A particular customer of such a brokerage system, e.g. an investor, may regard information on the type of news he consults proprietary. Most likely he will not be willing to share this with others.

The personalized system also has possible infringement of customers’ privacy as an important drawback. It does, however, not use this private information to advice others. The main advantage of the collective system is that (since it only uses anonymous data) it does not invade customers’ privacy at all, nor can it potentially do so. A collective system thus actually tries to recommend based on domain specific “non-private” knowledge. In this paper we further focus on such systems, and in particular the use of sales data as the domain specific knowledge. We will call the collective system henceforth the *market oriented* system.

4 Bundling & Recommending

Thus far we have treated recommending and the dynamic bundling/pricing of goods as two separate approaches. In this section we discuss recommendation based on sales statistics. Especially for this type of recommender systems, integrating the dynamic bundling/pricing of goods into (active and automatic) recommendation can be beneficial.

4.1 Recommendation, Bundling, & Pricing

Simply put, the idea of recommendation based on sales statistics is that customers' buying behavior could reveal their preferences. A great advantage of recommendation based on sales statistics is that it does not require customers to instruct the system about their preferences which is rather time consuming for the customers and costly in terms of (human) resources. Moreover, what customers say about their preferences might not coincide with their actual behavior. Furthermore sales statistics can be derived from anonymous data; hence such a system does not necessarily require the infringement of customers' privacy.

For a seller of information goods it may be advantageous to integrate the dynamic bundling/pricing of goods into recommendation. Using on-line dynamic pricing/bundling techniques has— unlike more traditional data-mining techniques— the advantages that knowledge about customers is obtained interactively. The information broker can thus explicitly offer various bundles for sale and frequently experiment with the composition and price of these bundles. Consequently, the seller can improve the quality of the recommendations.

An additional advantage is that by buying a bundle which captures some of a customer's preferences the customer is saved the trouble of searching and paying for all the individual items that constitute the bundle. Thus, whenever the offered bundles closely match various customers' categories they could reduce the search and transaction costs for customers belonging to these categories.

4.2 An Integrated Recommender System

We present a framework for an integrated recommender system. The idea is to identify a collection of (sub)bundles with customer stereotypes. This collection of (sub)bundles represent the currently conjectured building blocks for the actual product bundles. The actual bundles offered are obtained by combining one or more customer stereotypes (or (sub)bundles). The sales data generated by the online dynamic pricing/bundling algorithms are used, on the one hand, to update the bundles and possibly the stereotypes. Customer feedback on recommended items is used, on the other hand, to update the stereotypes and possibly the bundles. In figure 1 we draw the basic flowchart of the integrated recommender system.

Initiating such an integrated recommender system can be particular difficult because there are so many bundling options. Moreover, very little is known about the various consumers' stereotypes. A reasonable approach could be to start off with a limited number of stereotypes, which— based on the already available knowledge of consumers' preferences— are expected to result in a reasonably prof-

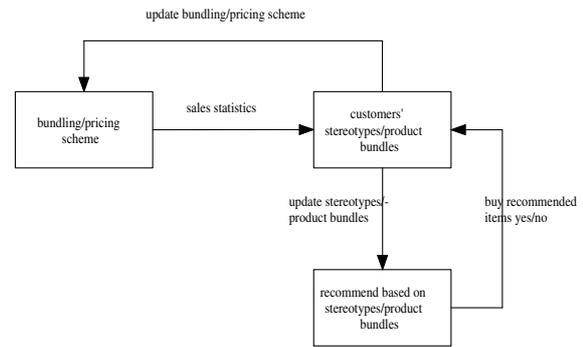


Figure 1. A integrated recommender system.

itable partition of the potential customers. The offered bundles could at least initially coincide with these stereotypes. A richer and more diverse collection of customer stereotypes can then gradually be developed based on the repeated sales statistics. We will present an explicit approach in Section 5.

5 Bundling Scheme & Experiments

In this Section, the focus lies on the dynamic bundling problem (as oppose to the dynamic pricing problem). Typically this problem is solved as a dynamic pricing problem (see Section 2). We, instead, explicitly address the problem of finding dynamically the appropriate bundle contents.

The conducted computer experiments follow the suggestion made in Section 4.2 to identify customer stereotypes with bundles. We use an advanced genetic algorithm (GA) with a niching method to *on-line* search for the bundles that coincide with the preferences of the various customer groups. As a feedback for costumers' preferences we use aggregate sales date, only. Thus, we learn more about customers' preference without any infringement of customers' privacy.

The experiments show how— through the use of on-line dynamic bundling— an information broker can develop more accurate customer stereotypes. Additionally, they show how exploitation of these more accurate stereotypes leads to customer lock-in given a small customer bias towards bundles. Another (not explicitly demonstrated) way to exploit these more accurate stereotypes is to have the integrated recommender system, outlined in Section 4.2, use these updated customer stereotypes to make more accurate recommendations. Thus, the experiments illustrate how the use of on-line dynamic bundling algorithms can both directly and indirectly increase the value-creating potential of an information broker.

We choose a very common but basic consumer model to simulate the customers. A limitation of this basic consumer model is that whenever bundles are not explicitly being offered aggregate sales data will reveal very little

about consumer preferences. By explicitly offering bundles for sale, the seller generates sales data on consumer demand for various groups of goods. From these data he can learn more about the preferences of the various customer groups. Hence, the basic consumer model does illustrate how, due to the interactive characteristics of the on-line dynamic bundling process, it is possible to manipulate the informativeness of the generated sales data. This *interactive* aspect of the learning process stands in stark contrast with traditional data-mining techniques.

5.1 Setup Market Simulation

In the conducted experiments we consider two types of markets: a monopolistic market (with 1 information broker); and a duopolistic market (with 2 information brokers). The monopolist offers a set of n individual goods and m bundles for sale in every trading period. The bundles are constructed from the set of all n individual goods. In the duopolistic market both brokers offer an identical set of n individual goods for sale in every trading period. One broker also offers m bundles for sale. To reduce consumers' problem of finding the desired information goods m is significantly smaller than 2^n (all possible bundles).

Although the brokers in the duopolistic market sell identical (individual) goods they, nevertheless, enjoy some monopoly power because consumers incur a search cost whenever they visit the other broker to compare prices. It goes beyond the scope of the paper to introduce an elaborate search model to formalize these search costs. For the purpose of the paper it suffices to simply introduce a tendency towards customer lock-in due to search costs. We obtain this tendency by assuming that there is no comparison between brokers within trading periods. There is, however, cross-period comparison.

At the beginning of every trading period a consumer chooses an information broker and sticks to that choice throughout the trading period. In most cases, the consumer chooses to visit the information broker with the highest expected utility. With a small probability of ϵ the customer deviates from this strategy. The utility of previous visits are used to compute the expected utility. More precisely, a consumer uses Q-learning to update the expected utilities and the ϵ -greedy policy rule to determine the next choice (cf. [12]).

We use a slightly modified version of the basic consumer models introduced by [3] to describe consumers' valuation for the various goods. The model consists of two (main) parameters: w , which indicates a consumer's value for its most preferred good and k , which (as long as the valuation is positive) determines, in combination with n , the difference between the i th and $(i + 1)$ th preferred good. For consumer j , the valuation $v_j(i)$ of the i th preferred good is

determined by the following function:

$$v_j(i) = \begin{cases} w_j(1 - \frac{i-1}{k_j n}) & \text{if } i \leq l_i + 1 \\ 0 & \text{if } i > l_i + 1 \end{cases} \quad (1)$$

with $l_i \leq k_j n$; l_i denotes the number of goods with a nonzero value. In order to create various relatively homogenous consumer groups w_j is relatively similar within a consumer group and relatively different between consumer groups. (See Figure 2 for an illustration of the consumer model.)

We assume that consumers optimize a linear utility function without any budget constraints. Let b denote the collection of individual goods and bundles that are being offered (by the information broker that is currently visited) and suppose a consumer purchases the collection b' out of b ($b' \subseteq b$). Moreover, let $G(b')$ denote the collection of individual goods that is associated with the purchase of b' ($G(b') \equiv \cup_{x \in b'} x$). Then utility maximizing by the j^{th} consumer implies $\max_{b' \subseteq b} U_j(b')$ with

$$U_j(b') = \sum_{i \in G(b')} v_j(i) - (p(b') + c_s \cdot |b'|). \quad (2)$$

The three additive parts of the utility function are from left to right: the value consumer j attaches to consuming the goods in $G(b')$; the cost of purchasing the bundles and individual goods in b' ; and the number of transactions ($|b'|$) times the search cost per transaction (c_s). The third part of the utility function introduces a consumer bias towards bundles (i.e., fewer transactions). The underlying idea is that it— is more convenient— requires less (search) efforts to obtain the desired goods through fewer transactions. (Note that whenever c_s is not zero it should be a relatively small positive number.)

5.2 Bundling Approach

Ideally, the sales of a particular bundle should imply that a utility maximizing customer is actually interested in all the goods which make up the bundle. To warrant such a conclusion we let the bundle price be a direct function of the price a consumer pays whenever all the goods are bought individually.

Suppose b_i is the collection of elements that define the i th bundle, $p(b_i)$ denotes the price of the i th bundle, $p(j)$ the (individual) price of the j th good, and $G(b)$ denotes the collections of all goods that are part of at least one of the m bundles currently being offered for sale by the information broker (i.e., $G(b) = \cup_{x \in b} x$ and $b = \{b_1, \dots, b_m\}$). We can then define the i th bundle price as follows:

$$p(b_i) = \sum_{j \in b_i} p(j) - \alpha_i \cdot \min(\{p(j) : j \in G(b)\}) \quad (3)$$

with $0 < \alpha_i < 1$; α_i is a function of the bundle size $|b_i|$ (the bigger $|b_i|$, the bigger α_i). Note that α_i represents the bundle reduction. In the experiments it is set very close to 0 such that offering bundles does not significantly influence total profit.

Equation (3) stands for a whole family of bundle-price functions. The distinguishing property of this family is that the sales of a particular bundle means that a utility maximizing customer is interested in all the individual goods that make up the bundle and no bigger sized bundle for which this property holds is being offered. For the experimental results on learning customer preferences (see Section 5.4) it only matters that the used pricing scheme belongs to this family of bundle-price functions.

Since bundle prices are a direct function of the individual good prices, we can now use two learning algorithms: one to adjust the individual good prices (and consequently the bundle prices) and another to adjust the bundle definitions. The task of the pricing algorithm is to adjust the prices with the objective of maximizing total revenue. The bundling algorithm adjusts the bundle definitions with the objective of maximizing the revenue generated by the individual bundles. Due to the way bundle prices are constructed this means that the algorithm searches for the bundles that coincide with the preferences of the various customer groups.

5.3 The On-Line Learning Algorithms

The focus in the experiments lies on on-line discovering the best bundle definitions. The purpose of the dynamic pricing algorithm is mainly to mimic the more realistic situation where the bundling algorithm works in conjunction with an algorithm that on-line adjusts the individual good prices (and consequently the bundle prices). We therefore choose a relatively standard and well documented dynamic pricing algorithm; consequently, it also not necessary to discuss the pricing algorithm in great detail.

Like [6] we use the Amoeba algorithm as the on-line dynamic pricing algorithm. Amoeba is a hill-climbing algorithm. Given n individual goods the Amoeba algorithm stores $n + 1$ price vectors. Every price vector represents a possible pricing scheme for the n goods. The geometrical representation of these price vectors is a simplex. By manipulating the shape of the simplex the algorithm searches for the pricing scheme which results in the highest profit. For further details on the Amoeba algorithm (or simplex method) cf. [9].

We use an advanced genetic algorithm (GA) with a niching method to on-line adjust the bundle definitions. The problem of finding the best bundle definitions is a combinatorial optimizing problem of finding *multiple* optima in a search space. A GA combined with a niching method can relatively quickly find multiple optima, which is essentially

our problem. Two elegant aspects of this technique are that it does not require explicit information about the number of optima in the search space, and that significantly lower optima can also be detected.

We follow the recommendations of [10] and [8] by using a clearing procedure as the niching method, combined with stochastic uniform sampling, and an elitist strategy. (See [8] for a discussion of the clearing procedure and [7] for a discussion of stochastic uniform sampling and elitist strategies.) The clearing procedure is called after evaluating the fitness of individuals and before applying the selection operator. Given a list of all individuals in the population sorted in decreasing order according to the fitness, the procedure works roughly as follows (cf. [8]):

1. Pick from the remaining individuals the next individual. This individual *dominates* the current niche.
2. The remaining individuals of the current niche are all individuals within the “clearing radius” σ of the dominant individual. The fitness of all except the first γ individuals in the current niche is reset to zero.
3. All individuals of the current niche are removed from the list.
4. Go to line 1.

The clearing method uses a dissimilarity measure (in our case the normalized Hamming distance) to determine if two individuals belong to the same subpopulation. Individuals belong to the same niche if the clearing radius is less than σ . The capacity γ of a niche specifies the maximum number of elements that this niche can except. Clearing preserves the fitness of the γ best individuals of a subpopulation and resets the fitness of the others that belong to the same subpopulation.

In each generation of the GA, the dominant individual of each niche competes with the corresponding one of the previous generation. The winners of the resulting *tournaments* are directly place in the new population. Stochastic uniform sampling is used to determine the remaining individuals in the new population.

Given n goods an individual has a chromosome of length n . The definition of one bundle is encoded onto one chromosome, where a bit value of 1 (0) at the i th position means that the i th good is (not) part of the bundle. The fitness of an individual is determined by the revenue of the bundle it represents.

By construction of the bundle price a consumer does not want a bundle if it already contains one good she is not interested in (see Section 5.2). Consequently, an almost but not completely “correct” bundle might result in a zero revenue. To prevent that certain almost correct bundles in the GA die out too soon, sigma scaling is used to recompute the “raw” fitness (cf. [7]).

5.4 Results Experiments

The GA has a population of 90 individuals that evolve for at most 100 generations. The initial population constitutes a random draw from all individuals that define bundles of 2 goods. The GA uses two point crossover. The mutation and crossover probability are 0.02 and 0.6, respectively. Throughout the conducted experiments n , the number of individual goods, is set to 20 and m , the number of offered bundles, is at most 5.

For the consumer model (discussed in Section 5.1) We consider 4 consumer groups each interested in a different group of products. The number of customers per group and the value of the most preferred goods are randomly drawn from uniform distributions. There are 245, 102, 235 and 160 consumers in group 1, 2, 3, and 4, respectively. In order to test the robustness of the bundling scheme we consider 8 different cases for the customer groups. Case 1 is the case where all 4 customer groups are interested in 5 different goods (see figure 2).

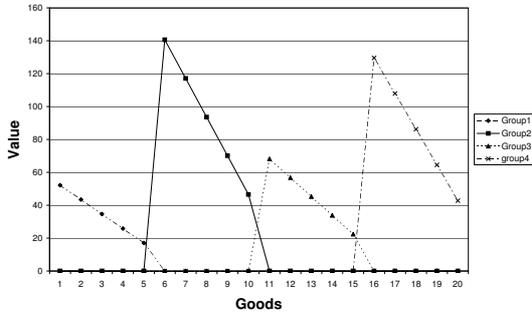


Figure 2. Average values of groups for case 1

We generate the other cases by slightly permutating the ordering of the i^{th} preferred good, and by adjusting l the number of nonzero valued goods and k the speed at which the i^{th} preferred good becomes 0 (see Section 5.1) such that the number of goods all customer groups are interested in increases. We consider the cases where the customer groups have a common interest in 0, 1, 2, 3, 4, 8, 12 and 16 out of the 20 goods (see table 1). (To put it differently, we consider the cases where the hamming distance between the various customer groups ranges from 10 to 2.)

Case	Int ^a
1	\emptyset
2	{6}
3	{6, 11}
4	{6, 11, 16}
5	{1, 6, 11, 16}
6	{1, 2, 6, 7, 11, 12, 16, 17}
7	{1, 2, 3, 6, 7, 8, 11, 12, 13, 16, 17, 18}
8	{1, 2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 16, 17, 18, 19}

^a Goods all customer groups are interested in.

Table 1. Shared interest

We put an upper and lower bound on the individual good prices determined by the Amoeba algorithm. These bounds are chosen so that the goods demanded by the four customer groups remain unchanged. Consequently, we can judge how well the GA performs given stable niches but changing prices. The bounds are chosen so that they maximize the interval in which a price can move. Due to the price changes it becomes more difficult for the GA to compare the fitness of two bundles traded in different periods which belong to the same generation. To reduce the negative effects of these asynchronous price changes somewhat, the fitness of an individual is measured as a percentage of the total profit in that period.

We conducted two series of experiments: the first series apply to the monopolistic market and the second to the duopolistic market. In the first series of experiments there is only one information broker (the monopolist) who uses the GA to find the various customer groups. In the other series of experiments there is also another information broker who only offers the 20 individual goods for sale. Table 2 shows the performance of the GA for the first series of experiments. It shows the average result of 25 conducted experiments for the 8 cases.

Case	σ	NF ^a	BF ^b	Gen ^c	P ^d	FE ^e
1	0.15	3.68	98.2	72	1099	5376
2	0.15	3.68	98.6	70	1112	5420
3	0.15	3.68	98.6	70	1159	5632
4	0.15	3.84	98.9	66	1115	5407
5	0.15	3.88	99.4	58	1001	4849
6	0.10	3.60	98.2	74	1386	6706
7	0.10	3.68	99.4	66	1488	7246
8	0.10	2.76	92.0	82	2026	10003

^a Average number of niches found out of 4.

^b Average % of goods found in niches.

^c Average number of generation to find niches.

^d Average number of trading periods to find niches.

^e Average number of fitness evaluations.

Table 2. Results monopolistic market

For most cases the clearing radius (σ) is set to 0.15, which means that individuals belong to the same niche as a dominant individual if their Hamming distance is 2 or less. Because the problem changes so dramatically from case 1 to case 8 it is necessary for a good performance of the GA to set σ to 0.10 for some cases (individuals belong to the same niche as a dominant individual if their Hamming distance is 1 or less). Throughout the experiments γ , (which denoted the number of individuals in a niche that are not reset) is set to 4. The performance of the GA for the first seven cases is very good. The fact that customer groups have common interest does not necessarily complicate the task of the GA. The reason is that there are more useful building blocks than whenever the customer groups are completely disjunct. For the last case the performance of the GA drops (but remains good) because the difference between the various customer

groups becomes very small.

In the second series of experiments we repeat the experiments of the first series only now with an additional information broker. The performance of the GA drops from finding around 98% of all goods in the niches to finding around 94% of the goods. Table 3 shows the results with a search costs (c_s) of 2 (see Eq. (2)) and ϵ — the probability of deviating from visiting the “best” store— is set to 0.05 (see Section (5.1)). The GA does find most niches. However, every

Case	σ	NF ^a	BF ^b	p ^c	FE ^d	MS ^e
1	0.15	3.44	95.8	1193	5814	40
2	0.15	3.00	91.1	1523	7427	28
3	0.15	3.00	94.0	1647	8006	52
4	0.15	3.28	95.9	1607	7799	68
5	0.15	3.12	96.3	1579	7650	68
6	0.10	2.76	94.1	1897	9205	76
7	0.10	2.76	95.9	2059	10079	84
8	0.10	2.72	90.6	2094	10310	96

^a Average number of niches found out of 4.

^b Average % of goods found in niches.

^c Average number of trading periods to find niches.

^d Average number of fitness evaluations.

^e Average number of times market share $\geq 60\%$.

Table 3. Results duopolistic market

now and then all or almost all customers interested in a particular niche end up choosing the competitor. Consequently, it becomes difficult for the GA to maintain the optimal solution generation after generation. Although the GA does not generally preserve knowledge of all niches throughout the simulation, the broker with the bundling scheme does obtain a market share of 60% or more in most of the experiments from case 3 up until including case 8 (see last column table 3). Thus, customer lock-in does not require perfect performance of the GA. The reason why customer lock-in occurs in especially the latter cases is that the absolute search cost (due to the fact that customers are interested in more goods) increases. (The bundle reduction consumers get by buying bundles does not contribute to customer lock-in in these experiments due to the relatively insignificant absolute value of the reduction.)

6 Concluding Remarks

In this paper we argue that it can be advantageous to integrate on-line dynamic pricing and/or bundling techniques into a recommender system; particularly, if such a system recommends based on (non-personal) sales statistics. We present the framework of such an integrated recommender system. In addition we presented on-line learning techniques by which the interest of the various customer groups can be learned effectively and efficiently based on the sales data. We explicitly show through computer experiments how on-line dynamic bundling techniques can facilitate the

(on-line) development of more accurate customer stereotypes. Additionally, the experiments show how exploitation of these more accurate stereotypes leads to customer lock-in given a small bias towards bundles. Based on these more accurate stereotypes the integrated recommender system can also reduce customers’ search costs by making more accurate recommendation and consequently create customer lock-in.

Acknowledgment

We would like to thank David van Bragt and Pieter Jan ’t Hoen for their helpful comments. This research has been performed within the framework of the project “Autonomous Systems of Trade Agents in E-Commerce”, which is funded by the Telematics Institute in the Netherlands.

References

- [1] R. Amit and C. Zott. Value creation in ebusiness. *Strategic Management Journal*, 22:493–520, 2001.
- [2] Y. Bakos and E. Brynjolfsson. Bundling information goods: Pricing, profits and efficiency. *Management Science*, 45(12), December 1999.
- [3] J. C.-I. Chuang and M. A. Sirbu. Optimal bundling strategy for digital information goods: Network delivery of articles and subscriptions. *Information Economics and Policy*, 11(1):147–176, July 1999.
- [4] A. Greenwald and J. O. Kephart. Probabilistic pricebots. In *Fifth International Conference on Autonomous Agents*, 2001.
- [5] A. R. Greenwald and J. O. Kephart. Shopbots and pricebots. In *Proceedings of Sixteenth International Joint Conference on Artificial Intelligence*, volume 1, Stockholm, August 1999.
- [6] J. O. Kephart, C. H. Brooks, and R. Das. Pricing information bundles in a dynamic environment. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, pages 180–190. ACM Press, 2001.
- [7] M. Mitchell. *An Introduction to Genetic Algorithms*. The MIT Press, Cambridge Massachusetts, 1996.
- [8] A. Pétrowski. A clearing procedure as a niching method for genetic algorithm. In *Proceedings 1996 IEEE International Conference on Evolutionary Computing*, pages 798–803. IEEE, 1996.
- [9] W. H. Press. *Numerical Recipes*. Cambridge University Press, 1992.
- [10] B. Sareni and L. Krähenbühl. Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computing*, 2(3):97–106, September 1998.
- [11] J. B. Schafer, J. A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5(1/2):115–153, January - April 2001.
- [12] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. The MIT Press, Cambridge Massachusetts, 1998.