ELSEVIER

Computer Networks 32 (2000) 731–752

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

Dynamic pricing by software agents

Jeffrey O. Kephart [a,*], James E. Hanson [a], Amy R. Greenwald [b]

[a] *IBM Institute for Advanced Commerce, IBM Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA*

[b] *Department of Computer Science, Brown University, P.O. Box 1910, Providence, RI 02912, USA*

Abstract

We envision a future in which the global economy and the Internet will merge, evolving into an *information economy* bustling with billions of economically motivated software agents that exchange information goods and services with humans and other agents. Economic software agents will differ in important ways from their human counterparts, and these differences may have significant beneficial or harmful effects upon the global economy. It is therefore important to consider the economic incentives and behaviors of economic software agents, and to use every available means to anticipate their collective interactions. We survey research conducted by the Information Economies group at IBM Research aimed at understanding collective interactions among agents that dynamically price information goods or services. In particular, we study the potential impact of widespread shopbot usage on prices, the price dynamics that may ensue from various mixtures of automated pricing agents (or "pricebots"), the potential use of machine-learning algorithms to improve profits, and more generally the interplay among learning, optimization, and dynamics in agent-based information economies. These studies illustrate both beneficial and harmful collective behaviors that can arise in such systems, suggest possible cures for some of the undesired phenomena, and raise fundamental theoretical issues, particularly in the realms of multi-agent learning and dynamic optimization.

Keywords: Information economy; Electronic commerce; Dynamic pricing; Economic software agents; Multi-agent systems; Emergent behavior; Shopbots; Pricebots

1. Introduction

We believe that, over the course of the next decade, the global economy and the Internet will merge into an *information economy* bustling with billions of autonomous software agents that exchange information goods and services with humans and other agents. Agents will represent — and be — consumers, producers, and intermediaries. They will facilitate all facets of electronic commerce, including shopping, advertising, negotiation, payment, delivery, and marketing and sales analysis.

In the information economy, the plenitude and low cost of up-to-date information will enable consumers (both human and agent) to be better informed about products and prices. Likewise, producers will be better informed about and more responsive to their customers' needs. Low communication costs will greatly diminish the importance of physical distance between trading partners. These and other reductions in economic

* Corresponding author.

E-mail addresses: kephart@watson.ibm.com (J.O. Kephart), hanson@watson.ibm.com (J.E. Hanson), amygreen@cs.brown.edu (A.R. Greenwald).

PII: S1389-1286(00)00026-8

friction will be exploited and contributed to by software agents that will respond to new opportunities orders of magnitude faster than humans ever could.

The agents we envision will not be mere adjuncts to business processes. They will be *economic* software agents: independent, self-motivated economic players, endowed with algorithms for maximizing utility and profit on behalf of their human owners. From other agents, they will purchase inputs, such as network bandwidth, processing power, or database access rights, as well as more refined information products and services. They will add value to these inputs by synthesizing, filtering, translating, mining, or otherwise processing them, and will sell the resultant product or service to other agents. In essence, these agents will function as miniature automated businesses that create and sell value to other agents, and in so doing will form complex, efficient economic webs of information goods and services that respond adaptively to the ever-changing needs of humans for physical and information-based products and services. With the emergence of the information economy will come previously undreamt-of synergies and business opportunities, such as the growth of entirely new types of information goods and services that cater exclusively to agents. Ultimately, the information economy will be an integral and perhaps dominant portion of the global economy.

Today's Internet is inhabited by a relatively modest assortment of software agents, some of which play useful roles in electronic commerce. Two familiar examples are *shopbots*, such as those at mySimon.com and DealPilot.com, which collate product and price information to aid shoppers, and *recommender systems*, such as the one at Amazon.com, which automatically suggests potentially interesting items to shoppers. Almost all such agents interact directly with humans, and almost none of them behave as economic decision-makers in their own right. Given this state of affairs, how and why do we expect the information economy to emerge?

The transition to the information economy will begin as an evolutionary step. The tremendous pressures that have fueled the rapid growth of electronic commerce in the last few years will continue to drive automation, and some of this automation will be encapsulated in the form of software agents. As they grow in sophistication and variety, software agents will begin to interact, not just with humans, but with one another. Interactions among agents will be supported by a number of efforts that are already under way, including standardization of agent communication languages, protocols, and infrastructures by organizations such as FIPA [9] and OMG [21], myriad attempts to establish standard ontologies for numerous products and markets (CommerceNet [5] being one prominent player in this arena), and the development of various electronic payment schemes such as IBM MicroPayments [13].

When interactions among agents become sufficiently rich, a crucial qualitative change will occur. New classes of agents will be designed specifically to serve the needs of other agents. Among these will be "middle agents" [6] that provide brokering and matchmaking services. By and large, middle agents (and in fact virtually *all* agents) will charge for their services, both to prevent abuse of their valuable resources and to provide income for their human owners. [1] Even functions that are not directly associated with commerce, such as search engines or more sophisticated computational services offered by application service providers, may be represented by agents that charge other agents or humans.

Just as today's global economy functions as a decentralized mechanism that adjudicates and satisfies the myriad conflicting needs of billions of human economic agents, it seems plausible that the information economy could provide coherency to an even larger population of economic software agents. However, it would be dangerous to assume that theories and intuitions based on centuries of experience with human economic agents will be directly applicable to understanding, anticipating,

[1] The currently prevalent means of supporting information resources on the Web — selling banner ads — will not suffice for agents unless they learn to read ads. It is unclear why anyone would make the effort to endow agents with such a capability.

and controlling the behavior of markets in which software agents participate. In effect, we are contemplating the release of a new species of economic agent into the world's economy — an environment that has heretofore been populated solely with human agents. This new species may be created in our image, but it will not behave exactly as we do. Economic software agents will make decisions and act upon them more quickly than humans. While they may be more expert at certain narrowly-defined tasks, they are likely be less generally capable and flexible than we are. Before unleashing them on the world's economy, it is essential that we consider their economic incentives and behaviors very seriously, and we must use every available means to anticipate their collective interactions with one another and with us.

This survey reviews several recent results obtained by collaborators affiliated with the Information Economies project at IBM Research [2] on collective interactions among agents that dynamically price information goods and services. It describes how we have used the tools of modeling, analysis and simulation to provide insights into the interplay among dynamics, optimization, and learning in agent-based economies. This work represents one aspect of a more comprehensive effort to understand and foster the transition to a global agent-based information economy. The Information Economies project as a whole encompasses theoretical work on agent-based economies, development of software infrastructure and componentry for supporting economic agents, and cooperative work on designing industry-wide standards for interoperability of software agents and multi-agent systems.

Section 2 gives an overview of recent work on dynamic pricing by agents, and sets the stage for a detailed investigation of the dynamics of an economy of *shopbots* and *pricebots*, presented in Section 3. In Section 4, we move beyond simple pricing, showing how agents might base decisions concerning product attributes or configurations on economic considerations. We shall conclude in Section 5 with a summary and a brief preview of our future research on information economies.

[2] See acknowledgments and bibliography for a full list of collaborators.

2. Software agents and dynamic pricing

Regardless of its vocation, practically every economically motivated agent will have a component that is involved in negotiation over price and other attributes of information goods and services. Agents will use a wide variety of negotiation mechanisms, including auctions (of which there are at least several million enumerable types, many of which can be further parametrized [33]), one-on-one negotiation, exchanges, and posted pricing. Research on all of these types of agent-mediated negotiation is already proceeding, and a few simple examples of these technologies can already be found on the Internet today. Thus automated negotiation stands on its own as an important field of study, quite apart from its eventual importance in the information economy.

A simple present-day example of an agent that participates in *auctions* is the Bid–Click proxy agent available at Amazon.com's auction site. A buyer interested in an auctioned item merely specifies a minimum and maximum bid, and the Bid–Click agent increments the bid automatically in response to challenging bids until either another bidder exceeds the agent's maximum or the agent wins the auction on behalf of the buyer. Kasbah [4] is an agent-mediated marketplace developed by researchers at MIT's Media Lab, in which human users delegate the responsibility for buying or selling physical goods to agents that engage in *one-on-one negotiations* with another agent. Park et al. [22] at the University of Michigan have studied the use of Markov modeling to create successful bidding strategies for agents participating in a *continuous double auction*; this can be regarded as a sort of *exchange* in that it supports trade among multiple buyers and sellers that freely enter and leave the market. Books.com is an online bookseller that employs *posted pricing* (i.e., it announces a non-negotiable, take-it-or-leave-it price), but uses a price-comparison agent that dynamically undercuts the price offered by its chief competitors: Amazon.com, bn.com, and Borders.com.

Despite the popularity of the online auctions that one finds at Amazon, eBay, and hundreds of other sites, posted pricing is the dominant form of pricing on the Internet today, and certainly it is the most common retail pricing model — the one with which we are most familiar in our everyday purchases of physical goods. Another feature of posted pricing that makes it attractive for agents is that price determination is quicker than it is for other forms of negotiation. A human buyer may tolerate waiting two weeks for the close of an auction of a Willie Mays baseball card at Amazon.com. However, consider a CD-review-finder agent looking for reviews of the Karajan performances of the Nine Beethoven Symphonies on behalf of a human or another agent. Part of its strategy might be to issue specially formulated queries to search engines in an effort to locate likely candidate reviews. The review-finder agent will not want to have a protracted negotiation with the Excite search engine over the price of 100 hits. It will need to settle the price and make the purchase within a fraction of a second. Multi-round auctions or negotiations will almost certainly be too slow in such a situation. Continuous double auctions may be viable, but posted pricing is guaranteed to be the fastest mechanism.

In this paper, we focus on what we call *dynamic posted pricing* — that is, take-it-or-leave-it pricing in which the seller may change the price at any time. We particularly emphasize the collective interaction among multiple sellers that attempt to maximize profits by employing price-setting algorithms that are more — or less — sophisticated than what Books.com is using today. We use the adjective "posted" to emphasize that we are not considering online auctions, which constitute another important class of dynamic pricing mechanisms. Technically, posted pricing can be regarded as a simple form of auction, but there is an important distinction: in auctions the price changes occur during the course of each transaction, while in dynamic posted pricing the price changes occur across different transactions.

3. Shopbots and pricebots

Shopbots [8,20], or comparison shopping agents, offer an early glimpse of the vast reductions in economic friction that are likely to occur as the world makes the transition to the information economy. Over the past few decades, several economists, including Diamond [7], Wilde and Schwartz [32], Varian [30], Salop and Stiglitz [25], Burdett and Judd [3], and Hopkins and Seymour [14] have studied the question of how sellers could make any profit at all in a competitive commodities market, since the traditional Bertrand equilibrium argument [29] suggests that competitive pressures should drive prices down to the marginal production cost. They found that the costliness of discovering prices — a factor not taken into account in the traditional Bertrand argument — could make it rational for buyers to forego comparing prices. [3] This in turn allows sellers to charge more than the marginal production cost.

For example, suppose that you want to purchase *Harry Potter and the Sorcerer's Stone*. If you were to drive around to two or three local bookstores, it could take at least an hour or two to find and purchase the cheapest copy of the book, and the savings would almost certainly not justify the time and expense of shopping. One could reduce the shopping time to perhaps 15 minutes by calling bookstores on the phone, and may be even to 5 minutes by checking Amazon.com, bn.com (the online outlet for Barnes and Noble), and Borders.com on the Internet. As of this writing, one would discover that bn.com was just edging out the other two with a price of \$8.97 as opposed to \$8.98 for each of the other two — only a penny saved for 5 minutes of work.

However, suppose that you use the book shopbot at DealPilot.com, one of several such shopbots that have come into existence within the last year or two. Once you specify that you want to shop for *Harry Potter and the Sorcerer's Stone*, you press the *Start Price Comparison* button, and

[3] Non-zero profits can also arise from other factors such as product differentiation, which is discussed in Section 4 of this paper.

within 20 seconds a table of roughly 50 combinations of booksellers and shipping options appears, ordered from least to most expensive. At the top of the list, one finds that shopping.com is offering the book for just \$8.47 — at least half a dollar saved for less than one minute's work, which many would consider worthwhile.

As an increasing number of buyers begin to avail themselves of shopbots, and as shopbots become more pervasive and powerful, one of the frictional forces that has sustained profits for sellers in commodities markets will be reduced substantially. Price-aware buyers will be price-sensitive buyers, and they will force sellers to become extremely responsive in their pricing. We anticipate that humans will not be able to keep up with the demands of responsive pricing on millions of goods and services, and that instead sellers will rely increasingly on what we term *pricebots* — agents that adjust prices automatically on the seller's behalf in response to changing market conditions. We can regard Books.com's agent as an early example of a pricebot; it automatically adjusts the price to slightly less than the minimum price offered by Amazon, Barnes and Noble, and Borders.

3.1. Model

In order to understand the implications of widespread adoption of shopbots and pricebots, we have modeled a simple market in which S sellers compete to provide B buyers with a commodity, such as a specific book. [4]

The objective of each seller s is to set its price p_s so as to obtain the maximum profit, given a production cost r. Each buyer b behaves in a very simple way: it compares s_b prices and purchases the good from the seller that charges the least, provided that the price is less than the buyer's valuation v_b. Assuming that the search strategy s_b and the valuation v_b are uncorrelated, the buyer population can be represented by a strategy vector w (the ith component of which represents the fraction of buyers that compare i prices) and a valuation distribution $\gamma(v)$. The strategy vector w may be fixed exogeneously — for example, 25% of the population may shop manually, just trying a random seller and buying if the price is right ($s = 1$) while the remaining 75% may use a shopbot to search *all* sellers' prices ($s = S$). Alternatively, if buyers set their search strategies so as to minimize the expected cost of the good plus the cost of the searching, endogeneous forces may cause w to evolve over time.

In the next three subsections, we explore a few points in a broad spectrum of possible utility maximization strategies that sellers, buyers, or shopbot intermediaries might employ in a competitive environment, focusing on collective phenomena that arise from various strategy choices.

3.2. Sellers and automated pricing

First, we focus on the sellers and their efforts to set prices so as to maximize profits. Here, we consider three basic types of pricebot algorithms that span a wide range of requirements on the availability of market information and computational power, and observe the price dynamics that ensue when various combinations of these algorithms are run against one another.

The first such algorithm, GT, is based on a game-theoretic computation. It can be shown that, unless the price quantum is very coarse, there is no pure-strategy Nash equilibrium [10] (i.e., there is no stable set of equilibrium prices). However, there is a mixed-strategy Nash equilibrium in which each seller chooses prices randomly from a distribution $f(p)$ that can be computed from the buyer parameters w and γ and the number of sellers S. The GT algorithm simply computes $f(p)$ and periodically chooses a different random price from that distribution. An implicit assumption in the game-theoretic derivation is that no seller observes another seller's price before it sets its own.

The second algorithm, MY, is referred to as the *myoptimal* ("myopically optimal") or "best-response Cournot" algorithm. Like GT, it requires perfect knowledge of the buyer population and the number of competing sellers. In addition, it requires knowledge of the current prices of all

[4] For a more complete presentation and study of the model, see Refs. [10,11,17].

competitors. Using this information, it sets the price to the value that will maximize profits in the short term — up until the moment when some other seller changes *its* price. We have experimented somewhat with variations in which the optimization is imperfect, either due to imperfect search or imperfect information.

The third algorithm, DF, or *derivative-follower*, is the least informationally and computationally intensive of the algorithms. It does not require any knowledge or assumptions about buyers or competitors. It simply experiments with incremental increases (or decreases) in price, continuing to move its price in the same direction until the observed profitability level falls, at which point the direction of movement is reversed.

Fig. 1 summarizes the wide range of dynamical behaviors that result from various mixtures of pricebots. It is organized as a 3×3 matrix depicting the 9 possible combinations of 1 vs. 4 pricebots of strategy types GT, MY and DF. Table 1 summarizes the corresponding time-averaged profits for each of the 9 mixtures.

In all of these simulations, each of $S = 5$ sellers held its price fixed for a short random interval, and then (asynchronously) used its pricing algorithm to generate a new price. The buyer strategies were held fixed: 25% checked just one price, 25% checked two prices, and the remaining 50% compared all prices and selected the seller with the lowest price. The buyers' valuations were distributed uniformly between 0 and 1. The marginal production cost r turns out to have no qualitative effect on the results; it was set to zero.

The pure combinations (5 GT, 5 MY, and 5 DF sellers) lie along the diagonals of Table 1 and Fig. 1. First, consider the 5 GT sellers. Since GT does not observe its opponent's choice of price prior to making its own, and since an undercutter can grab a significant portion (80%) of the market share, there is a strong incentive to price low. The prices for 5 GT pricebots are strongly clustered just above the threshold price $p^* = 0.0196$. Below this price, undercutting is not worthwhile because the profit margins are so low that it is better to charge the monopolist price $p_m = 0.5$ and accept the low 5% market share consisting of $1/S$ of the $s = 1$ buyers. By pricing just above this p^*, a GT pricebot is playing a game of "chicken" — a higher price yields a higher profit, but increases the chances of being undercut. Due to the generally low prices, each GT pricebot earns relatively low profits of 0.0129 on average, as compared to the theoretical maximum of 0.05 on average [5] that could be obtained by a collusive cartel in which each seller charges the monopolistic price of 0.5.

Now consider the 5 MY sellers. As seen in the central cell in Fig. 1, they undercut one another until the price falls to p^*, at which point undercutting becomes less profitable than charging the monopolistic price of 0.5. Thus prices suddenly jump up to this level. However, as soon as this occurs, undercutting once again becomes attractive, and the price-war cycle begins anew. We have observed similar cyclic behavior in other models of markets in which myoptimal agents are present [18,24], some of which will be illustrated in the next section of this paper. Although the MY sellers fall into endless price wars, their average prices are higher than those of GT sellers. This is reflected in their much higher average profit: 0.0337 vs. 0.0129 in this example.

Now consider the 5 DF sellers. Interestingly, although they are the least informed, they maintain the highest prices and therefore the highest profits. Their average profit of 0.0387 is not very much less than the optimal value of 0.05 that could be attained by a cartel.

Can a less-informed and less computationally intensive algorithm such as DF really fare better than MY and GT? The off-diagonal elements of Fig. 1 and Table 1 illustrate what happens when different types of pricebot algorithms are mixed together. In particular, consider the rightmost column of Fig. 1 and Table 1. Note that, when a single GT or MY agent is pitted against 4 DFs, it fares much better than does a DF. GT consistently undercuts the 4 DFs because it is biased towards low prices; MY is even more effective because it undercuts the 4 DFs by the minimal amount necessary to grab the 80% market share. Thus, if

[5] The price 0.5 is acceptable to half of the buyers, who split their purchases evenly among 5 sellers, each purchase bringing a profit of 0.5 to that seller.

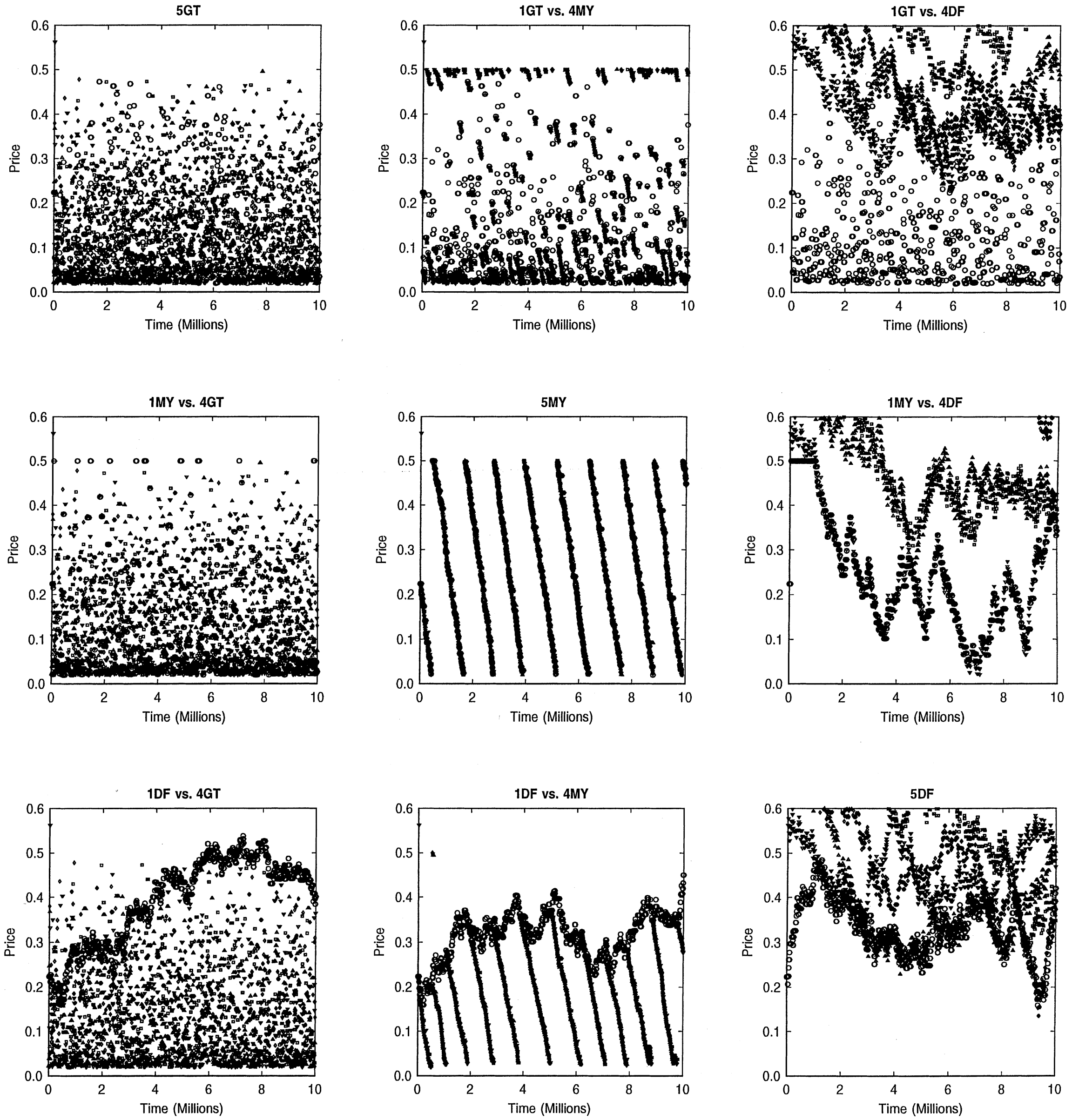


Fig. 1. Prices vs. time for 4-on-1 pricebot simulations. Open circles represent the "one" pricebot that may employ a different strategy than the other four. First row: One GT agent vs. 4 GT, 4 MY and 4 DF agents. Second row: One MY agent vs. 4 GT, 4 MY and 4 DF agents. Third row: One DF agent vs. 4 GT, 4 MY, and 4 DF agents.

agents were permitted to select pricing strategies on the basis of expected long-term payoff, a society of 5 DFs would be unstable. The first agent to reconsider its strategy choice would switch to MY, as would each successive agent until all were converted to MY. The situation is analogous to the Prisoner's Dilemma: self-interest compels all of the sellers to defect to the MY strategy, even though this leads to a lower profit than would be obtained if they were to adhere to the DF strategy.

Table 1
4-on-1 profit matrix[a]

	4 GT	4 MY	4 DF
1 GT	0.0129, 0.0129	0.0119, 0.0169	0.0536, 0.0226
1 MY	0.0185, 0.0109	0.0337, 0.0337	0.0690, 0.0225
1 DF	0.0134, 0.0159	0.0136, 0.0361	0.0387, 0.0387

[a] Within each cell, the profit of the 1 agent is given on the left; the average profits of each of the 4 agents of identical type are given on the right.

The bottom row of Fig. 1 and Table 1 illustrates the inferiority of the DF algorithm from another perspective. Here we find that, when a lone DF pricebot is introduced into a society of 4 GTs or 4 MYs, it is exploited: the GT's and MY's fare better with DF than they do with an agent of their own kind, and the DF pricebot fares worse than the other agents.

Can we conclude from this that MY will be the algorithm of choice for pricebots? Certainly not. If detailed buyer information is unavailable, then a simpler strategy like DF may be the only choice. In practice, the detailed buyer information required by MY and GT is unlikely to be obtained very easily. At best, one might be able to design an approximate version of MY that employs adaptive learning techniques to set prices. Additionally, even if reasonable approximations to the MY policy can be made, it still cannot be regarded as the preferred strategy because we have only considered three strategies out of an infinite spectrum. Is there another pricing algorithm that can beat MY? Certainly. We have explored *two* ways of improving MY: making it faster and making it less myopic. We now discuss each of these in turn.

3.2.1. Fast myoptimal

First, we explore what happens when MY's re-pricing rate is increased. Intuitively, it is clear that such a pricebot would spend more of its time undercutting its competitors. Analysis and simulations confirm that, if all sellers adopt MY, then the one that re-prices fastest makes the most profit. (This observation seems likely to extend to other pricing strategies as well.) An arms race in re-pricing rate seems inevitable.

Let us develop this scenario a little further. Suppose that a bookseller like Amazon were to implement a pricebot that resets prices on one million titles every day. In order to compute the new prices using MY or a related strategy, Amazon would have to obtain quotes from each competitor on each title. To do this, Amazon could use its own special-purpose shopbot. However, it might prefer to write a simple agent that automatically sends requests for price quotes to DealPilot or another commercial shopbot. This would take much less effort, and it would also hide the fact that Amazon was seeking valuable price information from its competitors. If Amazon requests one million price comparisons per day, each of which takes 20 seconds on average, the shopbot would be hit with several hundred requests simultaneously.

For this herculean task, the shopbot would gain absolutely nothing. Many shopbots earn their living by selling advertising space on their Web pages or by making a commission on sales made through referrals. However, Amazon would not make any actual purchases from its competitors, so DealPilot would get no commissions, and Amazon's agent would not read any of the ads put up by DealPilot.

The scenario gets worse. Assume that DealPilot is somehow able to meet Amazon's constant demand. In order to gain a competitive advantage over Amazon, bn.com might ask DealPilot to check prices on one million titles *every hour*. Then Borders might up the ante to one million titles *per ten minutes*, and another dozen online booksellers could follow suit. DealPilot could wind up receiving over a billion requests per day. It is doubtful that such a load could be handled. Even if the booksellers were relatively conservative and just focussed on the ten thousand most popular titles, the load on DealPilot could easily dwarf the number of requests coming from legitimate buyers. Keep in mind that each request for a price comparison from a bookseller translates into several requests for price quotes that the shopbot submits to booksellers. Thus the total network traffic and the hit rate on each bookseller's server is potentially enormous.

A reasonable solution to the problem of excess demand for shopbot services would be for shopbots to charge pricebots for price information. Even if the cost per price-comparison were just a fraction of one cent, one might expect to reach a balance point beyond which the benefit of requesting extra price comparisons would be less than the cost of obtaining these comparisons from the shopbot. Once shopbots begin charging for pricing information, it would seem natural for sellers — the actual owners of the desired information — to themselves charge shopbots (and possibly other clients) for their information. The sellers could use another form of pricebot to dynamically price their own price information. This illustrates how dynamic pricing of information services could quickly percolate through an entire economy of software agents. We expect to see such situations repeatedly as the Internet makes the transition to the information economy: services or agents that charge for their information goods or services will create the proper incentives to encourage business but deter excessive, counterproductive resource usage.

Since it is pointless and inefficient to change the price of an item more frequently than it is being requested by buyers, another likely development is that sellers will price their wares only on demand. This is precisely what Books.com does today — it dynamically sets a price on a book when a buyer expresses interest. However, suppose for example that Borders were to adopt the same policy as Books.com. When Books.com checks Borders' price on a book, Borders does not have a ready-made price; instead, it must check other booksellers' prices — including Books.com's. Unless some care is taken, an endless pricing loop could be generated. One way to avoid such problems would be to set up a reverse auction whenever a buyer expresses interest in a book. Booksellers would bid against one another, and the lowest bidder would be obligated to sell the book to the buyer for the agreed-upon price. Thus the natural limit of ever-faster *non-negotiable* dynamic pricing would appear to be *negotiable* dynamic pricing mechanisms such as auctions, in which the dynamics may occur over the course of the transaction.

3.2.2. Foresight and machine learning

According to Table 1, the five myoptimal sellers in Fig. 1 each receive an average profit of 0.0337, or about two-thirds of the theoretical maximum of 0.05 that would be achieved by a cartel. One would expect an improvement if the algorithm were modified to take into account the anticipated future pricing behavior of competitors. Sellers might avoid undercutting if they could foresee that it might invite retaliation. Such thinking is very natural and intuitive for humans. The challenge is to endow software agents with such economic foresight without requiring that they be anywhere near the level of sophistication required to pass the Turing test.

One promising method that we have investigated is a reinforcement learning technique called *Q-learning*. A seller employing this technique learns a Q function that expresses the anticipated future-discounted profit for each possible price that it might charge, given the current prices of all of its competitors [26–28]. The anticipated future-discounted profit is simply the expected profit during the current time interval plus γ times the expected profit one time step in the future plus γ^2 times the expected profit two time steps in the future, etc., where the future discount parameter γ lies between 0 and 1. The seller's pricing policy is to select the price that offers the highest future-discounted profit.

Q-learning is known to converge when the function to be learned is stationary. Thus a single Q-learner is guaranteed to develop an optimal policy against a single non-adaptive DF, GT, or MY agent. Against DF and GT, Q behaves essentially like MY [11]. However, against MY, Q learns a superior policy. Fig. 2(a) depicts Q's policy $p_{\mathrm{Q}}(p_{\mathrm{MY}})$, i.e., its price as a function of MY's price. Rotated and superimposed on this is MY's pricing policy $p_{\mathrm{MY}}(p_{\mathrm{Q}})$. Successive iteration of the two policies from a particular initial condition yields a trajectory in price vector space that is

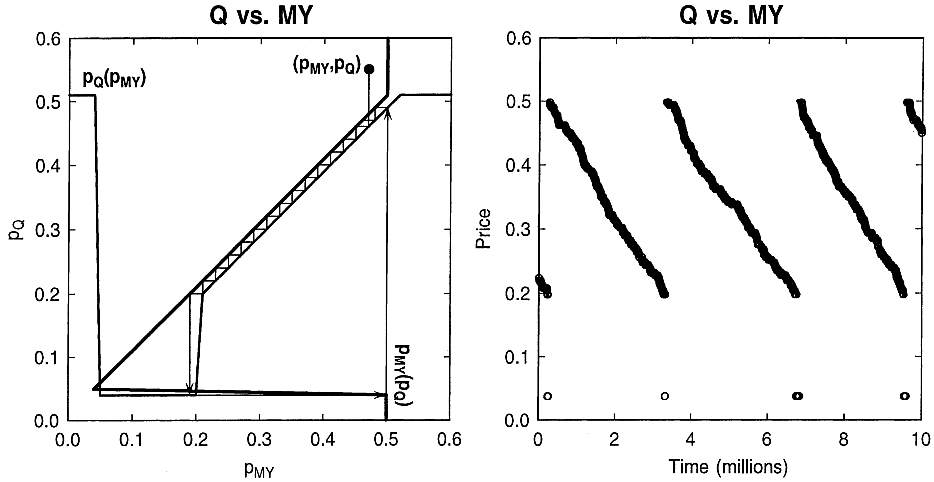


Fig. 2. (a) Cross plot of Q pricing policy vs. MY pricing policy. The future discount parameter is $\gamma = 0.5$. The thin line shows a typical price trajectory from a given initial condition (represented by dot). (b) Associated price-war time series.

represented in Fig. 2(a) by a thin zigzag curve. Assuming that Q moves first from the initial condition $(p_{\mathrm{MY}}, p_{\mathrm{Q}}) = (0.49, 0.56)$, the trajectory moves vertically down to the $p_{\mathrm{Q}}(p_{\mathrm{MY}})$ curve, then horizontally to the $p_{\mathrm{MY}}(p_{\mathrm{Q}})$ curve, and then alternates between vertical and horizontal movements. The price trajectory is represented more conventionally in Fig. 2(b).

Q's policy is identical to that of MY for moderate to high prices, so the price trajectory begins as a price war similar in form to that of the 5MY cell in Fig. 1. However, below a critical price $p_{\mathrm{MY}} = 0.196$, Q's policy differs from that of MY. Instead of slightly undercutting MY, Q drops its price aggressively to $p_{\mathrm{Q}} = p^* = 0.037$. MY responds by raising its price to the monopolist price 0.5, whereupon Q undercuts it, beginning the price-war cycle anew. By eliminating the portion of the price-war cycle below price 0.196, Q improves its average profit from 0.0892 to 0.1089, and it also improves MY's average profit to 0.1076.

When two Q-learners are pitted against one another, each creates a non-stationary environment for the other. There are no theoretical guarantees of convergence in this case, and in fact we have observed both convergence and non-convergence. Convergence typically occurs when the future discount parameter γ is small (i.e., relatively little emphasis is placed on future earnings). In this case, two Q-learners typically converge to a stable, symmetric pair of policies, each of which is quite similar to Q's optimal policy against a single MY. When a moderate to strong emphasis is placed on the value of future rewards, the *Q*-learners' policies *nearly* converge to a strongly asymmetric pair of policies in which one of the agents has a clear advantage over the other. The convergence is imperfect, however, and the *Q*-functions and associated policies continue to make small oscillations around a well-defined but not quite attainable asymmetric solution. This asymmetric quasi-solution has no analog in ordinary single-agent *Q*-learning. Our early efforts to characterize this solution and the circumstances under which it arises hint of a rich field for investigation by theorists in machine-learning and dynamical systems.

The asymmetric solution is illustrated in Fig. 3(a), which shows the crossed policies for Agents 1 and 2. Both pricing policies are approximately describable by a small number of simple line segments, except for slight perturbations that arise from the instability of the asymmetric solution. The positions and sizes of the perturbations shift unpredictably as the *Q*-learners continue to learn. Superimposed on these two-crossed policies is a trajectory in price vector space that starts from

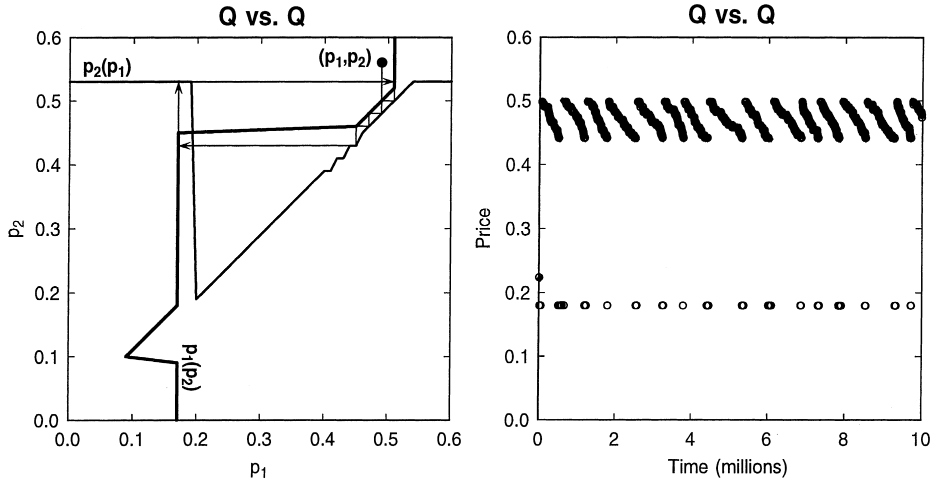


Fig. 3. (a) Cross plot of asymmetric pricing policies derived from simultaneous learning by two Q-learners. The future discount parameter $\gamma = 0.5$. The thin line shows a typical price trajectory from a given initial condition (represented by dot). (b) Associated price-war time series.

the initial condition $(p_1, p_2) = (0.49, 0.56)$ and proceeds vertically and horizontally as in Fig. 2(a). The classic price-war pattern is again evident in Fig. 3(a) and (more conventionally) in Fig. 3(b). However, in this case Agent 1 ends the price war quite early by aggressively lowering its price, inducing Agent 2 to set its price to the monopolistic value of 0.5. Since prices never fall below 0.44, the price war is short-lived, and profits remain very high. On average, Agents 1 and 2 receive 0.1254 and 0.1171, respectively. This exceeds what two MY sellers would obtain (0.0892 each), and it even exceeds the profits that would be obtained by two DFs (0.1127 each).

An important challenge is to extend the Q-learning technique so that it can feasibly handle more than one opponent. The natural way to express the Q-function is as a lookup table with one Q value per possible price vector. For a system with s sellers, the table is s-dimensional. Although we have found $s = 2$ to be manageable, the lookup table becomes infeasible in both size and trainability for $s \geqslant 3$. The obvious solution is to replace the lookup table with a function approximator. We have explored the use of neural nets [27] and regression trees as function approximators. In tests on two-seller systems, neural nets appear to have difficulty approximating the lookup table accurately and take a very long time to train, but regression trees appear to offer good approximation and good training times. We are beginning to assess their performance on systems with three sellers.

3.3. Buyers and their search strategies

In the information economy, buyer agents will also make strategic choices based on economic considerations. We have explored economic decision making by buyers within the context of the shopbot model [17]. Suppose that there is a cost c_s for obtaining s price quotes. This might represent an intrinsic, implicit cost that reflects the time and effort required to obtain the quotes, or it may represent a real fee paid to a shopbot. Then a rational buyer b would not blindly adhere to a fixed search strategy. Instead, it would select a search strategy s_b to minimize the expected total cost of the item *plus* the search cost c_{s_b}, given current market conditions.

By making the simplifying assumption that all sellers use the GT pricing strategy, we have studied how the buyer strategy vector $\vec{w}$ evolves as a function of the costs c_s. Suppose that c_s is a sublinear function of s. One plausible justification for such a cost structure is that the first few quotes represent the overhead of going to a shopbot in the first place; additional quotes are relatively inexpensive because it takes little extra time to obtain them. At any given time step, we assume that a small fraction of buyers reconsider their strategy. Given the current GT price distribution $f(p;\vec{w})$, which itself depends on the buyer strategy vector, a buyer can compute the price it would expect to pay as a function of the number of quotes s. Of course, this is a monotonically non-increasing function of s. On the other hand, c_s can be assumed to be a monotonically non-decreasing function of s. Thus there is a balance point — an optimal s that minimizes the total expected expenditure. The buyers myopically switch from their current strategy to the one that is currently optimal. The sellers immediately readjust their distributions to reflect the updated value of $\vec{w}$, and a new set of buyers responds in turn to the updated $f(p;\vec{w})$.

Previous research has shown that, if the search costs c_s are equal to some constant times s, then the system evolves to an equilibrium in which only strategies 1 and 2 are present [3]. However, as depicted in Fig. 4, nonlinear search costs can lead to non-equilibrium evolutionary dynamics in which strategies other than 1 and 2 can co-exist. In related experiments, we have found that the buyer search behavior can be strongly influenced by the price structure c_s. The oscillations tend to grow in magnitude as the fraction of buyers that switch strategies at each time step grows, and the period can become shorter. Furthermore, different initial conditions can lead to very different final equilibria or limit-cycle attractors.

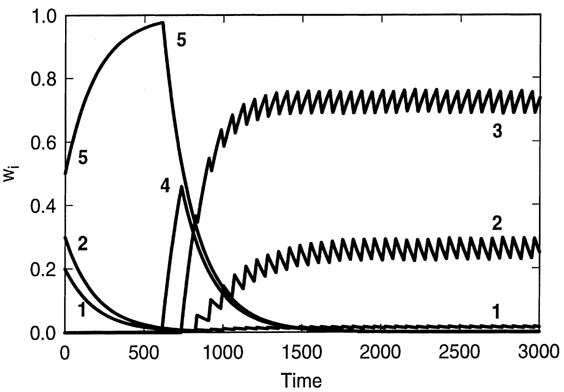


Fig. 4. Evolution of indicated components of buyer strategy vector $\vec{w}$ for 5 sellers, with nonlinear search costs $c_i = 0.05 + 0.02(i-1)^{0.25}$. At any given iteration, 0.005 of the buyers reconsider their strategy. Final equilibrium oscillates with period 15 around a mixture of strategy types 1–3.

3.4. Shopbots: how to price prices?

In the information economy, intermediaries will also be economic decision makers. In the shopbot model, we have investigated a scenario in which the cost structure c_s reflects an actual price paid by buyers to a shopbot, and have explored how a shopbot might manipulate c_s to maximize its own profit [17].

As a simplification, suppose that the shopbot offers two choices: a single quote for price c_1 and two quotes for price c_2. A competing mechanism for obtaining price information costs c' for one quote and $2c'$ for two quotes. For example, manual price comparison by a human (conducted by visiting multiple merchant web sites) might well cost an amount of time (and therefore money) proportional to s.

The optimal settings of c_1 and c_2 as a function of c' are depicted in Fig. 5. Regardless of c', the shopbot should always just undercut the alternate mechanism on a single quote. The price of the second quote $\delta = c_2 - c_1$ has a more complicated dependence on c'. For low c', the second quote should also be priced just less than c'. However, for intermediate values of c', the price of the second quote must be less than that of the first —

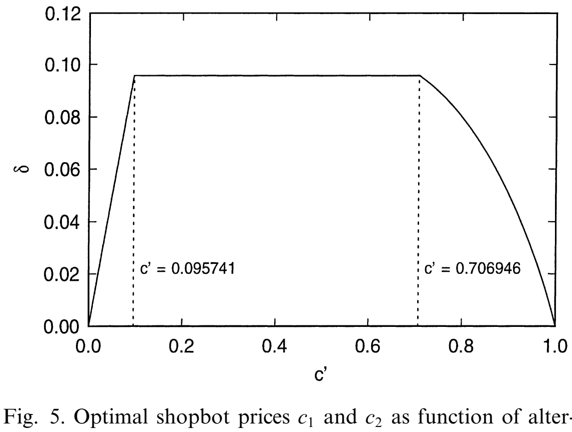


Fig. 5. Optimal shopbot prices c_1 and c_2 as function of alternative search cost c' (normalized to $v - r = 1$).

otherwise, too many buyers will be discouraged from buying two quotes. In this regime, δ should be a constant value 0.0957, which maximizes the product of δ times the fraction of buyers that purchase two quotes. For large values of c', δ must be reduced below 0.0957. Reducing δ encourages more buyers to purchase two quotes. Increased comparison shopping forces sellers' prices lower, making it possible for buyers to afford a high single-quote price c_1. At the extreme limit of $c' \to 1$, practically all buyers purchase two quotes. If almost all buyers are comparing prices, the sellers' prices drop to just above the marginal cost (zero). Thus the sellers get virtually no surplus. The buyers pay very little for the product itself, but pay almost their entire valuation to the shopbot for the price information, so they get no surplus either. Thus, if a shopbot has an effective monopoly on price information (i.e., the alternative search cost equals or exceeds the difference between buyer valuation and marginal production cost), then it can extract practically all of the surplus from the market.

4. Beyond simple pricing

The shopbot model portrays a commodities market in which all buyers seek to minimize the price they pay for the commodity. In many markets, however, products are a good deal more complex and configurable, and consumers may be concerned less with price than they are with other attributes of the product. Information goods and services will have a number of configurable parameters. Seller agents will both price and configure information goods and services, and buyer agents will need to apply complex multi-attribute utility functions in order to evaluate and select them.

In an effort to understand how agents might deal with complex multi-attribute goods and services, we have explored a variety of models that emphasize different aspects of product differentiation. This section provides an overview of three such models. The first deals with a vertically differentiated market in which products are distinguished by a simple "quality" parameter, and consumers have different tradeoffs between price and quality. Here, we observe that the market can be prone to price wars that are more complex in form than those observed in the shopbot model. The second and third models explore two related scenarios involving horizontal differentation: information filtering and information bundling. In both models, different consumers value various categories of information differently. Several important issues surface in these studies, including the emergence of cyclical behavior in price/product space and its detrimental effect on sellers and buyers alike, economic incentives for specialization, and the interplay among learning, optimization, and dynamics in multi-agent systems.

4.1. Vertical differentiation

Imagine a product with a single attribute that can be mapped into some notion of "quality". In other words, in the absence of any price differentials, all buyers would agree on which of two different values of that attribute was the more desirable. Examples of such an attribute might include processor speed, network communication rates, or monitor resolution. Of course, different buyers would have different tradeoffs between price and quality, represented as a utility function.

Suppose that each seller offers the product at a single level of quality, and that all buyers survey each seller's price and quality and choose the one that maximizes their utility. [6] Then, if buyers are sufficiently sensitive to prices, cyclical price wars such as that illustrated in Fig. 6 can be observed when sellers use the myoptimal algorithm to adjust prices but keep their qualities fixed. At the beginning of the cycle, there are two independent price wars: one between those with qualities $Q = 1.0$ and $Q = 0.9$ and another between those with $Q = 0.5$ and $Q = 0.35$. Suddenly, near time 125, the high-quality sellers suddenly drop their prices to join the other price war. The net result is that $Q = 1.0$, $Q = 0.9$ and $Q = 0.5$ wind up in a price war, while $Q = 0.35$ drops its price still lower, initiating a

[6] For further details of the model and the results, see Ref. [24].

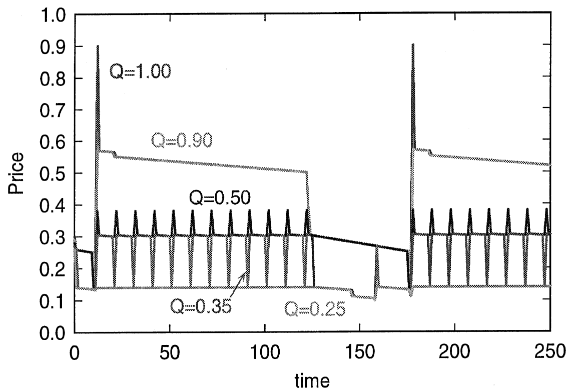


Fig. 6. Price vs. time for five myoptimal sellers of qualities Q equal to 1.0, 0.9, 0.5, 0.35, 0.25.

second price war with $Q = 0.25$. This rather complex cycle begins anew around time 175.

As in the shopbot model, a society of five derivative followers will maintain higher average prices and profits [24]. However, just as before, myoptimal pricing can take advantage of derivative following, so there will be a strong incentive to adopt myoptimal pricing even if this leads to lower profits when adopted by all sellers When the myoptimal sellers control both their quality level and their price, several different cyclical phenomena can occur. These include cyclical prices with fixed qualities, cyclical qualities with fixed prices, and simultaneous cycling in both prices and quality.

4.2. Information filtering and horizontal differentiation

One realm in which economically motivated software agents may play an important role is information filtering. Imagine an information source (perhaps a newsgroup or newsfeed) that produces a continual stream of articles in a wide variety of categories. Consumers, who are typically interested in only a very small subset of the categories, can avoid the high cost of receiving and examining a torrent of mostly irrelevant articles by subscribing to one or more brokers who purchase selected articles from the source and resell them to consumers. In such an environment, different consumers will be interested in different categories. Thus the market will be *horizontally* differentiated [29].

In our model of an information filtering economy [12,18,19], consumers experience a processing cost P_C for each article that they receive, and pay an additional fee P_b when they decide to consume an article offered by a broker b. Consumers hold a relevant article to be worth V, and an irrelevant one to be worth nothing. Broker b experiences a cost P_T for delivering an article to each consumer. Each broker b controls its price P_b and its selection of categories from among J possibilities (which can be thought of as its "product"). Consumers choose the set of brokers to which they subscribe, with brokers retaining the right to refuse subscriptions from consumers who appear unprofitable. Consumers seek to subscribe only to brokers whose selection of categories overlaps well with their own interests. Conversely, brokers wish to serve only consumers who are likely to be interested in their categories — otherwise they incur the cost P_T with little hope of being recompensed. Given the goals and capabilities of the consumers and brokers, we wish to understand the evolution of the brokers' prices, category selections, and profits and the consumers' broker selections and utilities.

For simplicity, assume that the aggregate demand for each category is exactly the same. Then, when there is just a single broker, all that matters is the *number* of categories it offers, not their identities. Analysis [12] shows that the broker can maximize its profit by offering J^* categories at a price P_b, where J^* and P_b depend on the costs P_C and P_T.

As illustrated in Fig. 7, three broad behavioral regimes are observed. When $P_C + P_T > V$, J^* is zero. In this "dead" regime, an article costs more to send and process than it is worth, even if the consumer is guaranteed to be interested in it. No articles will be bought or sold. At the other extreme, when the costs are sufficiently low ($P_C + P_T < vV$, where v is the average fraction of articles that are relevant to a typical consumer), the broker is motivated to offer *all* categories, i.e., $J^* \to \infty$. In real information filtering applications, one expects v to be quite small, since each con-

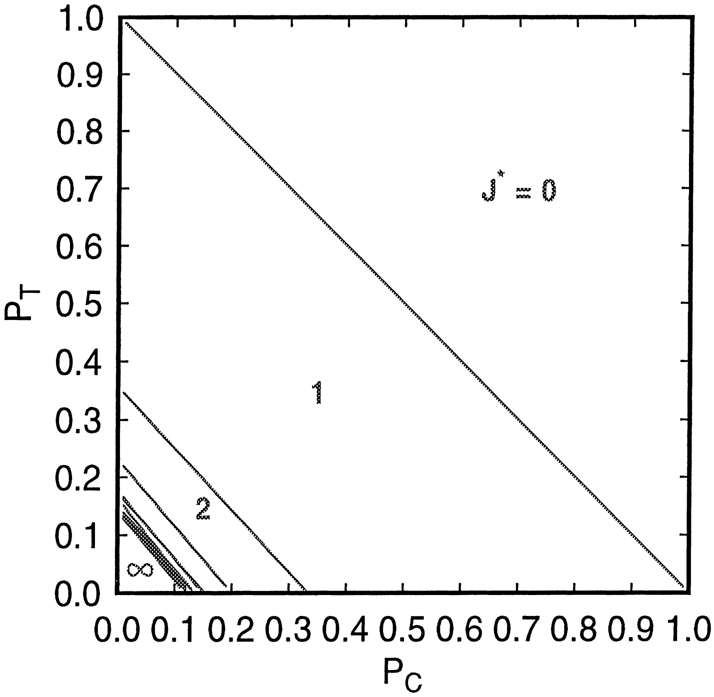


Fig. 7. Optimal number of categories J^* for monopolist broker to offer as a function of P_C/V and P_T/V, with $v = 0.1$.

sumer regards most information as junk. It is useful to think of $J^* \to \infty$ as a (presumably tiny) *spam* regime, in which it costs so little to send information, and the financial impact on a consumer of receiving junk is so minimal, that it makes economic sense to send all articles to all consumers. In between these two regimes, the optimal number of categories is finite.

When there is more than one broker, each will attempt to set its price and its category set (its product) optimally, taking into account both consumer demand and the current prices and categories offered by its competitors. In principle, the myoptimal strategy can be applied in the large space of possible prices *and* products: knowing consumer demand and other competitors' price and product parameters, a broker could choose a myopically optimal price and product. In practice, an exhaustive search for the optimal point in price/product space is only feasible for small numbers of brokers and categories, as the number of possible choices for each broker is $N2^J$, where N is the number of possible prices.

A more practical variant of the myoptimal strategy [19] replaces the exhaustive search with a limited search in which a fixed number of hypothetical price/product values are considered, and the one yielding the highest expected profit is selected. Candidate price/product values are generated in two ways, neither of which uses information about any of the consumers or other brokers: either by incremental changes to current parameter values or (less frequently) by choosing values completely at random.

Using this variant of the myoptimal algorithm, we simulated a system of 5 brokers, 5 categories, and 1000 consumers. The aggregate consumer demand for each category was identical. With P_C and P_T chosen such that $J^* = 1$, i.e., a monopolist broker would prefer to offer a single category, the system eventually evolves to a niche-monopoly state in which each broker offers one distinct category. The system is perfectly specialized, and consumer utilities and broker profits are both maximized. A similar state is reached when P_C and P_T are decreased somewhat to values such that

$J^* = 2$. Evidently, competition creates an additional pressure to specialize. In both experiments, the transient period was characterized by rampant competition and undercutting, but the system stabilized once it reached the fully specialized state. Finally, we lowered P_C and P_T to values in the spam regime, i.e., a monopolist would prefer to offer all five categories. In this case, competition never ceases, although it does cause the average number of categories offered by the brokers to drop from 5 down to 2.1.

When the aggregate demand is not identical for all categories, the system is susceptible to cyclical wars in price/product space. Even when $J^* = 1$, the niche-monopoly state is unstable because the broker occupying the most profitable niche is vulnerable to undercutting by other brokers that willingly abandon their own niches. Fig. 8 illustrates such a situation. There are three myoptimal brokers that may offer any combination of three categories. Prices are quantized such that there are 501 possible prices ranging from 0 to 1. Thus, every time a broker re-evaluates its price and product choice, it does an exhaustive search over 4008 possibilities.

In the depicted simulation run, each broker started from the same initial state (0.480, 111), i.e., each charges $P = 0.480$ for an offering that includes categories 1–3. After a brief initial transient in which two brokers compete for the (1 0 0) configuration (specialization on category 1), the system enters a cycle consisting of two price wars. The cycle begins with a short-lived competition between two brokers for the (0 1 0) configuration. When the price drops a bit, the (1 0 1) niche occupied by the third broker becomes more attractive, and all three brokers compete for this niche until the price drops a bit below 0.54, at which point a broker discovers that sole possession of (0 1 0) is more profitable. But as soon as it does, a second broker makes the same discovery, and the price war begins anew. (There are minor variations from one cycle to the next because the order in which the brokers re-evaluate their parameters is random.)

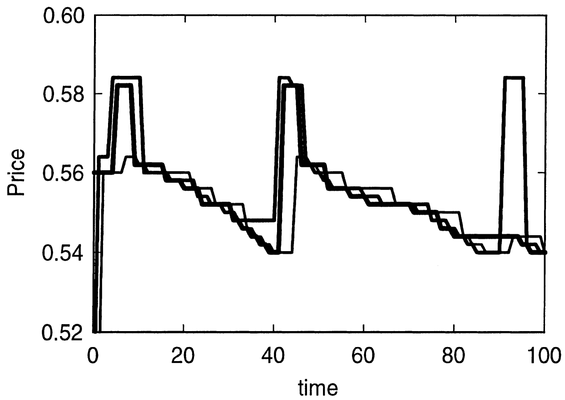


Fig. 8. Price and niche war timeseries: $P_b(t)$ vs. t for 3 myoptimal brokers and 3 categories, with $P_C = P_T = 0.3$, $V = 1$.

During the price wars, consumers who prefer less popular categories may suffer because no brokers are satisfying their needs. Thus, despite somewhat lower prices in favored categories, the aggregate consumer utility is often reduced during price and product wars.

The situation is improved when brokers use the myoptimal variant in which the search over the space of possible choices is limited to a small number of candidates clustered mainly in the vicinity of the current choice. In this case, we observe that the niche-monopoly can be metastable. In other words, specialization can occur and it can persist for long periods of time. Eventually, the period of calm and prosperity will be disturbed when a broker in a less profitable niche discovers that it can improve its profits (temporarily) by abandoning its niche and undercutting another broker. After a tumultuous round of competition, the brokers again settle into a niche-monopoly, and peace and prosperity reign again for a while [19].

4.3. Information bundling

A related realm in which economically motivated software agents are likely to play a significant role is *information bundling*. To a much greater extent than is possible in print, an electronic journal publisher can unbundle an issue and sell individual articles, or re-bundle articles together into personalized sets. Negotiation over the price and composition of bundles is likely to become a natural application for software agents.

Just as in the information filtering domain, agents will need to set or interpret a number of price and product parameters simultaneously. For example, suppose that N items are available for inclusion in a bundle. [7] Then each seller could offer 2^N different products. Rather than requiring the seller to choose one of them (as we did in the information filtering model), we permit the seller to offer all of these choices. This requires the seller to maintain prices for all 2^N possible products.

The seller can reduce the complexity of managing so many separate prices by introducing a pricing structure. One simple pricing scheme has just a single parameter: all items are priced identically at Δ, with no volume discounts. Thus the consumer pays $n\Delta$ to purchase n of the N possible items. Another one-parameter pricing scheme amounts to what is traditionally meant by "bundling": all N items are included in the bundle at a fixed bundle price of F — regardless of how many the consumer actually wants. An example of a two-parameter price structure is the "two-part tariff" scheme, in which the charge for n items is $F + n\Delta$, i.e., there is a per-item charge of Δ, plus a fixed fee F that is assessed if any items are purchased at all. If prices are based solely on the *number* of items in the bundle, then the most general price structure is a *nonlinear* scheme with N parameters — potentially an arbitrary monotonically nondecreasing function of the number of items purchased.

Suppose that there is just a single seller. Then its objective is to choose a price structure and the optimal price parameters for that structure. If the buyers' valuations of that seller's wares are known by the buyers and by the seller, this becomes a standard optimization problem. In general, the seller can extract greater profits from more complex price structures [1].

However, if the seller does not know the buyers' valuations a priori, it must use an adaptive procedure to adjust its price parameters. We have adapted the "amoeba" optimization method to this problem [23]. Starting from an arbitrary setting of price parameters, amoeba selects new parameters, measures the profit obtained at those parameters for a while, and uses these measurements to guide the choice of the next set of parameters. It is typically very successful at finding optimal or near-optimal price parameters, but while it is exploring the parameter space it may visit unprofitable regions. Thus it is important to minimize the number of evaluations required to attain near optimality.

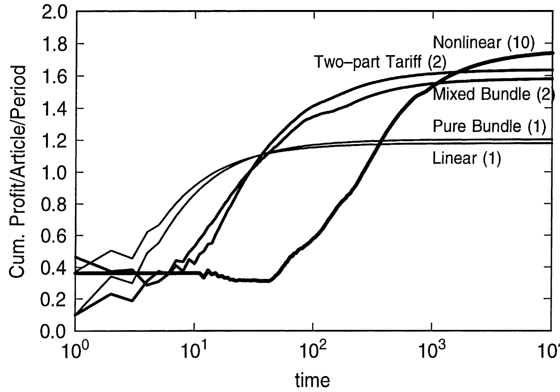


Fig. 9. Average cumulative profit vs. time for five different pricing structures using the amoeba algorithm.

Fig. 9 shows the time-averaged profit extracted by a monopolist seller that uses amoeba to learn the optimal setting of its parameters. The five curves represent various pricing structures ranging in complexity from 1 to 10 parameters. Although the nonlinear pricing structure with 10 parameters yields the highest profit asymptotically, it takes much longer to learn than the simpler pricing structures. If the time scale on which the market changes is shorter than the amount of time it takes the amoeba to conduct 1000 or more evaluations, the two-part tariff scheme may be preferable [2].

If the buyers themselves do not know their own valuations until they have a chance to sample the seller's wares, and if they allow for the possibility that the valuations shift in time, then the problem becomes much more complex. Suppose that the seller has adopted a two-part tariff scheme, and that buyers must pay the subscription fee F prior to examining the items and deciding how many to purchase at Δ per item. Suppose as well that the consumers use a simple form of maximum likelihood estimation to estimate the likely value to be obtained by subscribing.

[7] For further information about this model, see Refs. [2,16].

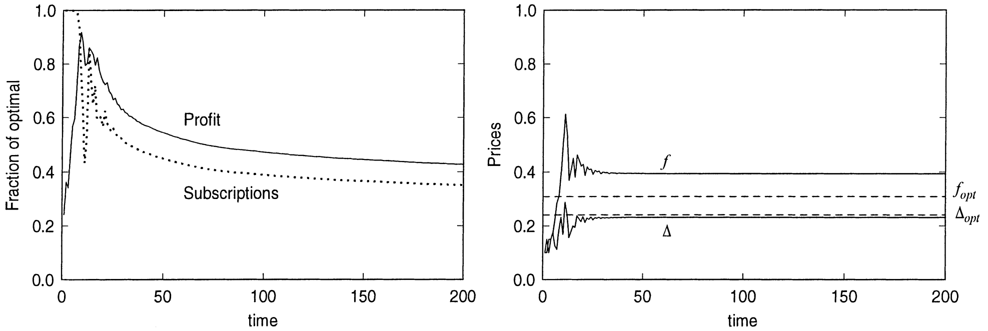


Fig. 10. (a) Profit π (solid line; normalized to "ideal" value of 0.41367) and proportion of subscribed consumers m (dashed line) vs. time (in subscription periods). (b) f and Δ vs. time when the producer uses amoeba for online learning. Horizontal dashed lines indicate the optimal f and Δ values for fully informed consumers. The market consists of $M = 10\,000$ consumers and one seller offering $N = 10$ articles per subscription period.

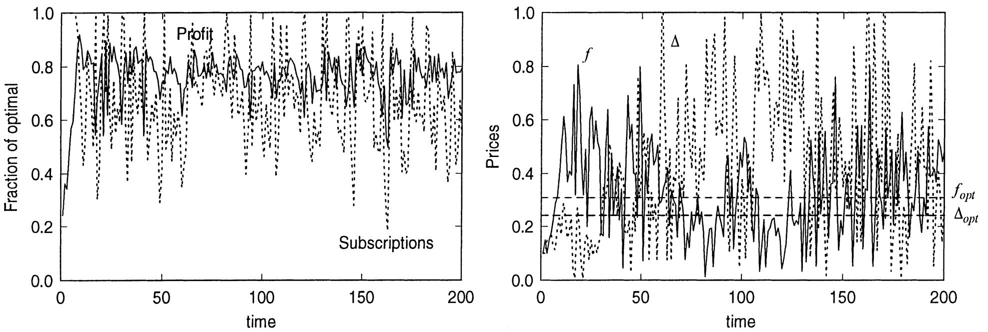


Fig. 11. (a) Profit π (solid line; normalized to "ideal" value of 0.41367) and proportion of subscribed consumers m (dashed line) vs. time (in subscription periods). (b) f (solid) and Δ (dashed) vs. time when the producer uses the modified amoeba algorithm for online learning. The parameter settings remain unchanged from Fig. 10.

In this scenario, we have observed an interesting "leakage" effect. If the seller sets F and Δ to the values that are optimal for perfectly informed buyers, profits decrease over time. Profit leakage occurs because, as buyers estimate the average value of the seller's wares, statistical errors sometimes lead a buyer to the false conclusion that the subscription fee F is not worthwhile. Once that buyer stops subscribing, it will stop receiving the information that it would need to revise its estimate, and therefore it will become disenfranchised permanently unless the seller lowers its prices.

One solution is for the buyers to periodically resample what the seller has to offer. However, even if buyers do not do this, the seller can still prevent profit erosion by lowering its prices. Once disenfranchised buyers re-enter the market, they will soon discover that their previous valuations were overly pessimistic, and they will (temporarily) stay in the market even if prices are raised back to previous levels. Unfortunately, as illustrated in Fig. 10, the amoeba algorithm as it is typically described is unable to discover how to manipulate prices dynamically so as to maintain profits [16]. Because the standard amoeba algorithm assumes that the optimization problem is not changing with time, it fails to notice that prices that were once profitable may no longer be after a while. This causes it to get stuck at a fixed setting of parameters, leading to profit erosion.

However, we have implemented a simple modification of the amoeba that recognizes the dynamic nature of the optimization problem. This version adjusts prices dynamically. As illustrated in Fig. 11, the price fluctuations are large and rapid, but they are generated in such a way that steady long-term profits are maintained. Ironically, it is precisely the fact that the modified amoeba recognizes the dynamic nature of the market that causes it to interact with buyers in such a way as to stabilize the market in the long term.

5. Conclusions

The information economy will be by far the largest multi-agent system ever envisioned, with numbers of agents running into the billions. Building economic behavior into the agents themselves offers a twofold promise: their myriad interactions and conflicts will be governed by the same economic principles that have lent coherency to the activity of billions of humans; and they will be able to operate in the same economic space as humans, to the benefit of both species. But there are potential pitfalls too. Economic software agents differ from economic human agents in significant ways, and their collective behavior may not closely resemble that of humans. Software agents must be designed with the understanding that they will be operating in (and helping to create) complex non-linear dynamical environments.

Even before the advent of the full-fledged information economy, automated pricing of physical goods such as books will be an interesting and important topic in its own right. Our hope is that, by exploring the non-linear, collective dynamics of automated posted pricing and by focusing particularly on the interplay among learning, optimization, and dynamics, we may obtain insights that are both intrinsically useful and more generally applicable to other aspects of agent-based information economies. Naturally, it will be important to study the relationship between individual agent strategies and collective market dynamics for one-on-one negotiation and auctions of all varieties. Beyond negotiation lie other important collective issues such as agent reputations. Some of the qualitative lessons that we have drawn from our studies of dynamic posted pricing seem likely to apply in these realms as well.

One general point is that agents will have to learn, adapt, and anticipate. In order to do so they will use a variety of machine learning and optimization techniques. Much of the work on machine learning and optimization has implicitly assumed a fixed environment or opponent. But agent economies are guaranteed to be dynamic by virtue of the fact that the agents are all learning. This violation of standard assumptions has important consequences.

For example, ordinary single-agent Q-learning is guaranteed to converge to optimality. When we introduced Q-learning into the shopbot model we found that two learners could *fail* to reach convergence (although they still reached a mutually beneficial state). Understanding the dynamic interactions among a society of learners is of fundamental theoretical and practical interest, and only a few beginning efforts have been made in this area [15,26,28,31]. A second example was our use of the popular amoeba optimization technique as a means by which an information bundler might optimize its profit. Here, the amoeba's implicit assumption that it was optimizing a static function caused it to fail miserably. The reason it was *not* optimizing a static function was that other agents in the system (the buyers in this case) were learning. In this case, it was possible to rectify amoeba by having it periodically re-evaluate its solutions. This resulted in short-term price volatility but long-term profit stability. It remains to be seen whether such a technique will be sufficiently adaptive to work against competition.

Another general lesson is that plausible agent strategies can lead to both beneficial and harmful collective behaviors. Economic incentives can drive the consumers and brokers in the information filtering economy to a niche-monopoly in which the consumers' utilities and brokers' profits are both quite high. However, all three models in which we allowed for multiple sellers (all but the information bundling model) were vulnerable to cyclical price-war dynamics. In the shopbot model, the myoptimal sellers were somewhat hurt by the moderately low average prices. In the information filtering model, the consumers were hurt as well because the cycling in price/product space caused brokers to ignore all but the most profitable market segments. The cycling can be eliminated or ameliorated by machine-learning techniques that allow agents to account for the future consequences of their present actions. We are currently investigating how to make these techniques practical for more than two sellers. Cycling behavior can also be reduced somewhat if the sellers are not sufficiently capable of computing (myopically) optimal decisions. This is likely to occur when information about buyers is hard to obtain, or when optimality is difficult to compute. In the first case, techniques such as amoeba can enable a seller to learn aggregate buyer information (at least in a monopoly). The second case is most likely to arise when the seller is setting product parameters as well as prices (as in the information filtering model) or several price parameters simultaneously (as in the information bundling model).

It would be imprudent to use the world's economy as an experimental testbed for software agents. Our approach will continue to be to use the familiar tools of modeling, analysis, and simulation to study and redesign agent strategies, protocols, and market mechanisms in the laboratory before releasing agents and agent infrastructures into the world's economy. An especially exciting aspect of this work is that it requires us to go beyond traditional tools and techniques, and leads us into new realms such as multi-agent learning in which new fundamental scientific developments and breakthroughs are required. We eagerly anticipate creative contributions from researchers in many fields, particularly economics, computer science, and applied mathematics.

Acknowledgements

This paper describes the collaborative efforts of several researchers in the Information Economies group at the IBM Thomas J. Watson Research Center and in the Computer Science and Economics departments and the School of Information at the University of Michigan. Our colleagues and co-authors at IBM have included Gerry Tesauro, Rajarshi Das, David Levine, Richard Segal, Benjamin Grosof, and Jakka Sairamesh. Our Michigan collaborators on information bundling research include Jeff MacKie-Mason (Economics department and School of Information), Scott Fay (Economics department), and Chris Brooks and Ed Durfee (Computer Science department). Special thanks go to Steve White, who helped conceive the Information Economies project, and has remained a guiding influence throughout its existence. Finally, we thank the IBM's Institute for Advanced Commerce and Steve Lavenberg and the rest of the Adventurous Systems and Software Research program committee for their continued financial support and guidance.

References

[1] Y. Bakos, E. Brynjolfsson, Bundling information goods: prices, profits and efficiency, in: B. Kahin, H. Varian (Eds.), Internet Publishing and Beyond: The Economics of Digital Information and Intellectual Property, MIT Press, Cambridge, MA, 2000.

[2] C.H. Brooks, S. Fay, R. Das, J.K. MacKie-Mason, J.O. Kephart, E.H. Durfee, Automated search strategies in an electronic goods market: Learning and complex price scheduling, in: Proceedings of the First ACM Conference on Electronic Commerce, ACM Press, New York, November 1999.

[3] K. Burdett, K.L. Judd, Equilibrium price dispersion, Econometrica 51 (4) (1983) 955–969.

[4] A. Chavez, P. Maes, Kasbah: an agent marketplace for buying and selling goods, in: Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, UK, April 1996.

[5] CommerceNet web page. URL: http://www.commerce.net.

[6] K. Decker, K. Sycara, M. Williamson, Middle-agents for the Internet, in: Proceedings of IJCAI-97, January 1997.

[7] P. Diamond, A model of price adjustment, Econom. Theory 3 (1971) 156–168.

[8] R.B. Doorenbos, O. Etzioni, D.S. Weld, A scalable comparison-shopping agent for the WorldWide Web, in: Proceedings of the First International Conference on Autonomous Agents, February 1997.

[9] Foundation for Intelligent Physical Agents web page. URL: http://www.fipa.org.

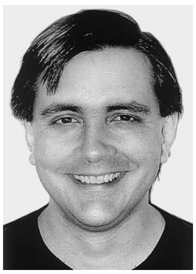
[10] A. Greenwald, J.O. Kephart, Shopbots and pricebots, in: Proceedings of 16th International Joint Conference on Artificial Intelligence, August 1999.

[11] A.R. Greenwald, J.O. Kephart, G.J. Tesauro, Strategic pricebot dynamics, in: M. Wellman (Ed.), Proceedings of the First ACM Conference on Electronic Commerce, ACM Press, New York, November 1999.

[12] J.E. Hanson, J.O. Kephart, Spontaneous specialization in a free-market economy of agents, in: Proceedings of the

Workshop on artificial Societies and Computational Markets, 1998.
[13] A. Herzberg, H. Yochai, MiniPay: Charging per click on the web, in: Proceedings of the Sixth International World Wide Web Conference, April 1997. URL: http://www.hr.il.ibm.com/mpay.
[14] E. Hopkins, R.M. Seymour, Price dispersion: An evolutionary approach, Unpublished Manuscript, October 1996.
[15] J. Hu, M.P. Wellman, Online learning about other agents in dynamic multiagent systems, in: Proceedings of the Second International Conference on Autonomous Agents (Agents '98), 1998.
[16] J.O. Kephart, R. Das, J.K. MacKie-Mason, Two-sided learning in an agent economy for information bundles, in: Agent-mediated Electronic Commerce, Lecture Notes in Artificial Intelligence, Springer, Berlin, 2000.
[17] J.O. Kephart, A.R. Greenwald, Shopbot economics, in: Proceedings of Fifth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, July 1999.
[18] J.O. Kephart, J.E. Hanson, D.W. Levine, B.N. Grosof, J. Sairamesh, R.B. Segal, S.R. White, Dynamics of an information filtering economy, in: Proceedings of the Second International Workshop on Cooperative Information Agents, 1998.
[19] J.O. Kephart, J.E. Hanson, J. Sairamesh, Price and niche wars in a free-market economy of software agents, Artificial Life 4 (1) (1998) 1.
[20] B. Krulwich, The BargainFinder agent: Comparison price shopping on the Internet, in: J. Williams (Ed.), Agents, Bots and Other Internet Beasties, SAMS.NET Publishing, 1996, pp. 257–263. URLs: http://bf.cstar.ac.com/bf, http://www.geocities.com/ResearchTriangle/9430.
[21] Object Management Group web page. URL: http://www.omg.org.
[22] S. Park, E.H. Durfee, W.P. Birmingham, Emergent properties of a market-based digital library with strategic agents, in: Proceedings of the Third International Conference on Multi-Agent Systems, July 1998, pp. 230–237.
[23] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, Cambridge, 1988.
[24] J. Sairamesh, J.O. Kephart, Price dynamics of vertically differentiated information markets, in: Proceedings of First International Conference on Information and Computation Economies, October 1998.
[25] S. Salop, J. Stiglitz, A theory of sales: A simple model of equilibrium price dispersion with identical agents, Amer. Econom. Rev. 72 (5) (1982) 1121–1130.
[26] G. Tesauro, J. Kephart, Pricing in agent economies using multi-agent Q-learning, in: Proceedings of Fifth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, July 1999.
[27] G.J. Tesauro, Pricing in agent economies using neural networks and multi-agent Q-learning, in: Proceedings of Workshop on Learning About, From and With Other Agents (IJCAI '99), August 1999.
[28] G.J. Tesauro, J.O. Kephart, Foresight-based pricing algorithms in an economy of software agents, in: Proceedings of First International Conference on Information and Computation Economies, October 1998.
[29] J. Tirole, The Theory of Industrial Organization, MIT Press, Cambridge, MA, 1988.
[30] H. Varian, A model of sales, Amer. Econom. Review, Papers Proc. 70 (4) (1980) 651–659.
[31] J.M. Vidal, E.H. Durfee, The moving target problem in multi-agent learning, in: Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98), July 1998.
[32] L.L. Wilde, A. Schwartz, Equilibrium comparison shopping, Rev. Econom. Stud. 46 (1979) 543–553.
[33] P.R. Wurman, M.P. Wellman, W.E. Walsh, The Michigan Internet AuctionBot: A configurable auction server for human and software agents, in: Proceedings of the Second International Conference on Autonomous Agents, May 1998.

Jeffrey O. Kephart manages the Agents and Emergent Phenomena group at the IBM Thomas J. Watson Research Center. He studied electrical engineering and physics at Princeton University and Stanford University. After a post-doctoral stint at Xerox PARC, he joined IBM Research in 1990. His primary research interest is the exploration and exploitation of analogies between large-scale computer systems and large-scale natural systems like ecologies and economies. He has pursued a number of theoretical and practical issues relating to computer viruses, and led the development of some novel anti-virus technologies, including a neural net detector and an immune system for cyberspace. His work has been cited or published in the Wall Street Journal, The New York Times, Wired, IEEE Spectrum, and Scientific American. Recently, Kephart has led IBM's Information Economy research effort, aimed at understanding and designing the future of software agents and electronic commerce. His other research interests include emergent phenomena in computer networks and intelligent mail agents.



James E. Hanson earned his Ph.D. in physics from the University of California, Berkeley, 1993. After completing a post-doctoral fellowship at the Santa Fe Institute, he joined the Agents and Emergent Phenomena group at IBM Thomas J. Watson Research Center in 1996. His research interests include pattern formation in complex systems, emergent phenomena in dynamic networks, and software agents for electronic commerce.

Amy Greenwald is an assistant professor of computer science at Brown University in Providence, Rhode Island. Her primary area of interest is multi-agent learning on the Internet, which she approaches using game-theoretic models of computational interactions. Previously, she was employed by IBM T.J. Watson Research Center, where she was involved in the Information Economies Project. She completed her Ph.D. at the Courant Institute of Mathematical Sciences of New York University, where she was funded in part by a dissertation fellowship awarded by the American Association of University Women. Most recently, she has presented her work at SIGeCOMM, the International Joint Conference on Artificial Intelligence (IJCAI), and the Stony Brook Summer Festival on Game Theory. In addition, she has publications forthcoming in the Journal of Games and Economic Behavior and in the Springer-Verlag series of Lectures Notes in Artificial Intelligence.