Business-to-Consumer Mobile Agent-Based Internet Commerce System (MAGICS)

Hui Chen, Perry P. Y. Lam, Henry C. B. Chan, *Member, IEEE*, Tharam S. Dillon, *Fellow, IEEE*, Jiannong Cao, *Senior Member, IEEE*, and Raymond S. T. Lee, *Member, IEEE*

Abstract—We present MAGICS, a mobile agent-based system for supporting business-to-consumer electronic commerce (e-commerce) or mobile commerce (m-commerce) applications. To use the system, consumers first provide their buying requirements to a proxy/agent server through a Web browser or a wireless application protocol (WAP) terminal. Having obtained the requirements, mobile agents are generated to carry out tasks for the consumers including getting offers from merchants, evaluating offers, and even completing purchases. In the case of mobile commerce, consumers can generate a mobile agent to conduct a search and evaluation in the digital marketspace before making a purchase in the physical marketplace. To make it possible to choose an offer that best satisfies the consumer's requirement(s), we present a mathematical model for evaluating multiple decision factors. To test the basic functions of the mobile agent-based Internet commerce system (MAGICS), we have built a prototype system. To minimize the average cost of a product (including the cost of sending agents), we have also developed an analytical model that can determine how many agents should be sent to compare prices. Four different price distributions and some real price information are analyzed based on the model. The analysis provides valuable insights into the design of mobile agent-based shopping applications for m-commerce, in particular, and for e-commerce, in general.

Index Terms—E-commerce, internet, mobile agents.

I. INTRODUCTION

IN GENERAL, e-commerce is concerned with the sale and purchase of goods or services through an electronic channel, particularly the Internet [1], [2] (see [1] for various definitions of e-commerce). With advances in mobile computing and wireless networking technologies, e-commerce can now be extended to the mobile environment [3]. Commonly known as m-commerce, this extension of e-commerce not only allows consumers to use mobile terminals to buy or sell anytime and anywhere but also offers location-based services [3], [4]. In current e-commerce or m-commerce systems, transactions are often processed by transferring many request/response messages between clients and servers over the Internet (e.g., see [5] and [6]). For some applications (e.g., searching for the lowest price), a mobile agent-based system can be more efficient [5], [6]. This is particularly useful in m-commerce since mobile terminals have many limitations in terms of input–output functions [4]. Hence, it is more effective to employ a mobile agent to assist a user to perform a certain task. In essence, a mobile agent can be sent over the Internet to communicate with other agents (e.g., to make a purchase). After accomplishing its mission, the mobile agent automatically returns to the originating server/terminal with the results. Inspired by [5] and [6], mobile agents can be employed to support the following tasks (i.e., distributed/parallel processing) in particular.

1) *Searching for information*. In consumer-oriented e-commerce or m-commerce, price comparison is an important task. A consumer can employ mobile agents to search for the lowest price for a product in the digital domain. The use of mobile agents allows a search to be run in parallel between multiple merchants.
2) *Making routine purchases*. Mobile agents are autonomous so they are particularly suited to making routine purchases in a distributed manner.
3) *Conducting negotiations*. In more advanced applications, mobile agents can be employed to conduct negotiations in parallel for their users, possibly using techniques of artificial intelligence.

Inspired by some agent-based commerce projects (see [7]–[9] and the later discussions), a mobile agent-based system called mobile agent-based Internet commerce system (MAGICS) is presented to support business-to-consumer e-commerce or m-commerce. In particular, our main contributions are as follows. First, we present an overview of MAGICS with a focus on facilitating the consumer buying process: search, evaluation, and purchase. Second, to support the mobile agent-based buying process and the satisfaction of a consumer's needs, we develop a mathematical model for evaluating attributes. Lastly, we formulate and solve a price comparison problem. The analysis provides valuable insights into the design of MAGICS for e-commerce- and m-commerce applications. This paper can be viewed as an extension of earlier work in [10] and [11].

The remaining sections of this paper are as follows. Section II provides an overview of some agent-based commerce systems. Section III presents MAGICS and its prototype. Section IV analyzes a price comparison problem related to the proposed system. Section V gives our conclusion.

II. RELATED WORK

This paper focuses on a consumer-oriented e-commerce- or m-commerce system. Fundamentally, its aim is to address the basic process of consumer buying, which consists of three main stages [8], [9], [12].

1) *Search for alternatives*. After specifying the buying requirements, a consumer needs to search for the alternatives that can satisfy the requirements.
2) *Evaluate the alternatives*. In this step, the consumer evaluates the alternatives (e.g., price comparison).
3) *Purchase the chosen product*. Lastly, the consumer chooses the best offer and makes the purchase.

In a physical or traditional retail system, a human consumer must visit a number of physical shops to search for alternatives, and then evaluate the alternatives, probably using some simple and subjective methods. After choosing the product, he/she then makes the purchase in the selected physical shop. Although some consumers may enjoy such a shopping experience, the conventional consumer buying process is, in general, time-consuming. Furthermore, a purchase decision is often made subject to many geographical limitations (e.g., a consumer can only visit shops within a small area) and with a limited amount of information (e.g., a consumer cannot know whether the best price found is indeed within the lowest range for the item). Current Web-based commerce systems eliminate these limitations by removing geographical constraints and giving more convenience to consumers. Specifically, a consumer can visit many more electronic stores in the marketspace and get much better information for making the buying decision. The next generation of e-commerce systems is likely to provide even more convenience to consumers, possibly through agents. In particular, a consumer will be able to assign shopping tasks to an agent. This means that, based on a consumer's requirements, an agent can be sent to perform the necessary shopping tasks automatically. In an m-commerce scenario, a consumer can first send a mobile agent to search and evaluate the digital marketspace before buying something in the physical marketplace.

Some agent-based systems have been developed to accomplish the aforementioned goal [7]–[9]. However, based on the studies by Moukas *et al.* [8] and Guttman *et al.* [9], many agent-based commerce systems only address part of the consumer buying process (see [8] and [9] for details). Hence, additional work and research is required to fully realize an agent-based commerce system. Moreover, there is considerable interest in developing an agent-based system for m-commerce. In recent years, a number of more advanced agent systems have also been proposed, such as Kasbah [13] and Tete-a-Tete (e.g., see [8] and [9]). Kasbah employs a centralized server to enable agents to communicate and carry out sales transactions [13]. To support negotiations, each agent is given a "price-time" function to determine how the desired price should vary over time (see [13]

for details). Tete-a-Tete attempts to conduct negotiations in a consumer-oriented system that involves more than one decision factor [8], [9]. Apart from business-to-consumer e-commerce, there has also been much interest in investigating agent-based auction systems. For example, a mobile agent-based auction system has been presented in [14]. It comprises a subsystem called Nomad for managing mobile agents and an auction server for conducting auctions (see [14] for details). Whereas many agent-based systems use a centralized architecture, MAgNET employs a distributed model [6]. Based on Java Aglet [5], buying agents are sent to sellers at different locations to buy the required goods. Mobile agents can also be employed for the sale and purchase of services. For instance, TabiCan allows mobile agents to buy/sell travel-related products/services (e.g., see [5]). Lastly, XPect provides a basic infrastructure that uses various kinds of agents to conduct e-commerce [15].

III. OVERVIEW OF MAGICS

MAGICS is distinct from other agent-based systems in that it is a mobile agent-based system designed to address the general consumer buying process. Before presenting the technical details, we first highlight the unique characteristics of MAGICS as well as our contributions. As mentioned before, very few agent-based systems are mobile agent-based and most only address some part(s) of the consumer buying process. Many of them use a centralized or semicentralized architecture (e.g., operating in a central server). MAgNET [6], Nomad [14], and TabiCan (e.g., see [5]) are also mobile agent-based systems, but each of them differs from MAGICS in some way(s). MAgNET [6] is a business-to-business system mainly designed for supply chain management. Like MAGICS, it is based on a distributed architecture, but it has a different application and the procurement procedure is sequential rather than parallel. Nomad is a mobile agent-based system for supporting an auction service. Hence, its operation is different from MAGICS. TabiCan (e.g., see [5]) is designed for the sale and purchase of travel-related products/services. This means that it is limited to a specific application. Using the consumer buying process as the basic research model, the MAGICS project is aimed at addressing the consumer buying process in an integrated manner. Specifically, our contributions are as follows.

1) Similar to MAgNET [6] and the Web model, and unlike Kasbah [13] and TabiCan (e.g., see [5]), which use a centralized server, MAGICS adopts a distributed model in which sellers can set up their own servers. The use of a distributed model allows the system to be implemented more effectively and efficiently on the global Internet. In particular, the use of a distributed architecture not only enables parallel processing but also facilitates system extension. Furthermore, we aim to incorporate MAGICS into the current Web system.
2) MAGICS uses the extensible markup language (XML) for agent communications rather than for storing and presenting information. This distinguishes it from many agent-based systems (e.g., Kasbah [13]), which communicate using predefined commands. The use of XML allows us

TABLE I
MAGICS AND WEB-BASED COMMERCE SYSTEMS

MAGICS	Web-based commerce systems
Mobile agents are served by MAGICS servers.	Client requests are served by Web servers.
Mobile agents are transferred using IBM's ATP [5] (e.g., the URL is atp://magics.company.com).	Information is transferred using HTTP (e.g., URL is http://www.company.com).
Information is carried by mobile agents.	Information is carried by HTTP messages.
Mobile agents are managed by means of a MAGICS browser/proxy server.	Information is viewed by means of a Web browser.
MAGICS search engines/agents are used to direct mobile agents to visit required servers.	Web-based search engines are available for searching purposes.

to build a systematic and extensible language for agent communications.

3) MAGICS makes use of parallel processing by means of mobile agents. Moreover, we develop an analytical model that can be used to determine the optimal number of agents for evaluating alternatives in parallel. As far as we know, this problem has not been investigated previously.

4) Besides basic price comparison, MAGICS can also process complex user requirements and constraints. To a certain extent, this is similar to Tete-a-Tete (e.g., see [8] and [9]), but in the paper we address both e-commerce- and m-commerce applications, such as a location-based application.

5) Resembling XPect [15], MAGICS not only provides other important transaction functions (e.g., payment) but also considers other pragmatic requirements (e.g., security). These functions and requirements, which have not been addressed fully in most of the earlier-mentioned systems, are essential to building a practical consumer-oriented e-commerce system.

In summary, MAGICS makes a contribution to the field of agent-based commerce in that it is designed to work for both e-commerce and m-commerce. Further, it not only combines some of the advantages of previous agent-based systems but also provides a number of original and distinctive features. Table I compares MAGICS with existing Web-based commerce systems.

Fig. 1 shows the four-layered model for implementing MAGICS. The first layer, the agent programming interface (API), provides the programming interface and package for developing stationary and mobile agents. For instance, the IBM Aglet can be used for this purpose. In the second layer (i.e., MAGICS agents), various component-based agents (e.g., buying agents) are built by means of the API. To enable interagent communications using a common model, various protocols are developed in the third layer: the MAGICS Protocols. Based on the earlier-mentioned layers, a wide range of consumer-oriented- and business-oriented MAGICS applications (i.e., the top layer) can be developed. The focus of this paper is on the consumer-oriented aspect.

As an extension of the work in [10] and inspired by the aforementioned agent-based commerce systems, particularly [6]
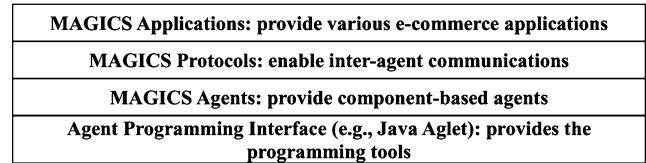
MAGICS Applications: provide various e-commerce applications
MAGICS Protocols: enable inter-agent communications
MAGICS Agents: provide component-based agents
Agent Programming Interface (e.g., Java Aglet): provides the programming tools

Fig. 1.   MAGICS model.

and [14], the MAGICS buying process for digital goods is described as follows (see also Fig. 2).

1) A user first communicates with the MAGICS proxy server via a wireless application protocol (WAP)-enabled device (e.g., mobile phone) or Web browser. The user then specifies the shopping requirements. For example, when buying a book, the user will enter the International Standard Book Number (ISBN), the number of copies required, the desired price, etc.

2) Having received the user's shopping request, a master agent is generated at the proxy server to control the whole shopping process. Based on the shopping request, the master agent will send an agent to search for merchants (e.g., booksellers) from the MAGICS search engines. Similar to a Web-based commerce system, it is expected that some search engines will be available to provide the locations of the merchants.

3) After identifying the MAGICS servers, the master agent sends some agents to those servers to obtain their offers. In a simple case, such as buying a book, an offer may only contain the price. In more complicated cases, the offer may include other information, e.g., possible options.

4) Having obtained all of the offers, the master agent then evaluates them according to the user's requirements. When buying standard goods, the requirement may simply be to find the lowest possible price. However, in complex cases where there are many decision factors, the master agent needs to evaluate the offers based on certain mathematical techniques to be discussed later. For example, taking into account the decision factors, the master agent needs to compare the weighted score of all of the offers to determine which offer is the best.

5) The master agent may show the evaluation to the user and the user may then make the final decision (similar to the approach used in [6]). Alternatively, the master agent may make the purchase directly.

6) In certain complex cases, the master agent may select a few merchants to engage in further negotiations (e.g., to lower the price). After selecting the best merchant, the master agent sends a buying agent to the target MAGICS server to complete the purchase.

7) At the merchant's server, the buying agent and selling agent communicate with each other to complete the purchase order. To enhance the extensibility of the system, the communication messages are specified using XML. When making payments, the buying agent may use the secure electronic transaction (SET) protocol. To build a comprehensive system, other payment options such as electronic cash and micropayment methods can also be developed.
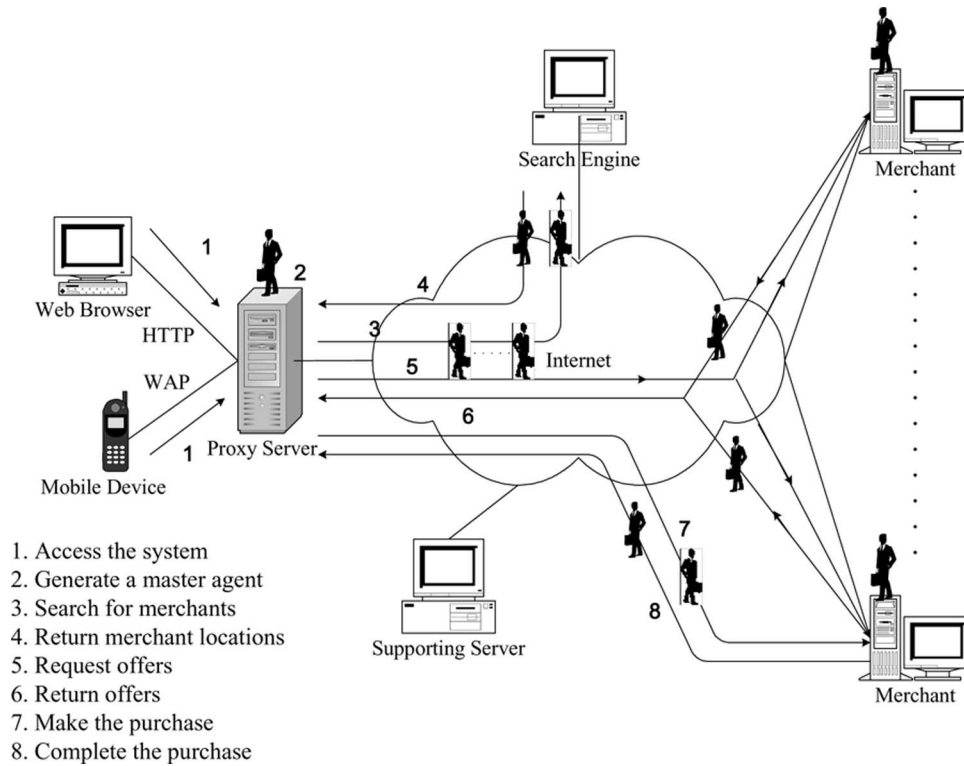
Fig. 2.   MAGICS buying protocol.

8) After receiving the payment, the selling agent provides the buying agent with a digitally signed receipt and arranges for the ordered goods to be delivered. In some cases (e.g., a digital book), the selling agent may provide the buying agent with the goods so that they can be carried back to the master agent directly. Having obtained the goods, the master agent stores them in the specified location and completes the shopping process.

This buying process can also be extended to support location-based shopping services in m-commerce. In this case, mobile agents can be employed to perform search and evaluation in the "digital space." The consumer will then make the final decision and complete the purchase in the "physical place." Essentially, after selecting the merchants, the master agent provides their physical locations (e.g., ranked in ascending order of distance to the consumer) to the consumer to complete the remaining buying tasks. This approach can combine the advantages of physical commerce and m-commerce.

As mentioned earlier, a consumer may need to evaluate multiple attributes in order to make a buying decision. Inspired by [16]–[18], the following presents an effective mathematical model for achieving this purpose. Suppose that there are some merchants for the agents to visit, where the ith agent visits the ith merchant. After receiving all of the offers, the master buying agent calculates the average weighted score $\bar{S}_i$ of the ith merchant so as to determine how well the consumer's requirements can be satisfied. Suppose that there are N attributes for evaluation, such as selling price and traveling time to a shop. The jth attribute is denoted as A_j and has a weight w_j to reflect its relative importance. To perform the evaluation based on the consumer's preference, we assume that all of the attributes can be "quantified" by using a rating or score. We define $S_{i,j}$ as the rating of A_j of the ith shop (i.e., the one visited by the ith agent). The average score $\bar{S}_i$ of the ith shop can be found by finding the weighted sum of all $S_{i,j}$ as follows:

$$\bar{S}_i = \sum_{j=1}^{N} w_j \times S_{i,j}. \tag{1}$$

Lastly, the preferred merchant is selected based on the highest $\bar{S}_i$.

We now discuss how to calculate an attribute score. In essence, it is computed by means of a satisfaction function. For each attribute (e.g., jth attribute), we assume that the offer by the ith merchant can be represented by $V_{i,j}$. For example, the ith merchant may return a selling price of \$300. In this case, A_j is the price and $V_{i,j}$ is 300. Note that we can have boolean attributes (e.g.., "Yes" or "No") or discrete attributes (e.g., red, green or blue, each associated with a score), as well. In this paper, we focus on continuous attributes (e.g., price, which has continuous values). Nevertheless, the model can be extended easily to cover other attributes. Based on an attribute value, the corresponding satisfaction score can be found by using a predefined satisfaction function $F(V_{i,j})$. As an example, we consider a linear function as follows (see Fig. 3):

$$S_{i,j} = F(V_{i,j}) = \begin{cases} \dfrac{h_j - V_{i,j}}{h_j - l_j} & \text{if } F(h_j) < F(l_j) \\[2mm] \dfrac{V_{i,j} - l_j}{h_j - l_j} & \text{if } F(h_j) > F(l_j) \end{cases} \tag{2}$$
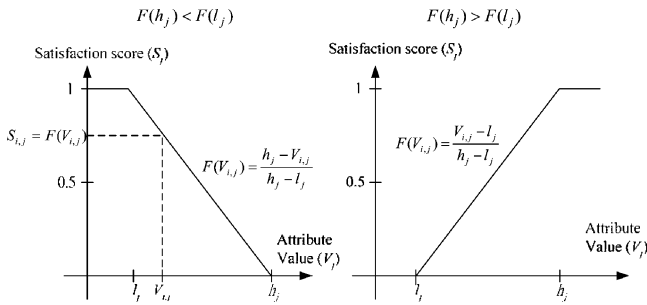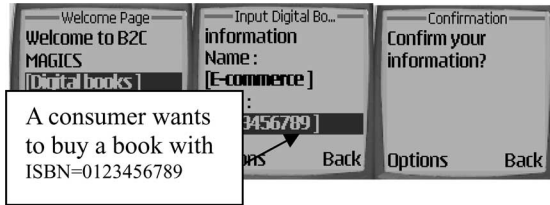
Fig. 3. Linear satisfaction function in two different cases.

Fig. 4. Accessing MAGICS via a WAP interface (Nokia simulator).


```
Agent 1 : The search result(s) for Shop1 is/are :
 <1> : ISBN=0123456789 Title=E-commerce Price=$380
Agent 2 : The search result(s) for Shop2 is/are :
 <1> : ISBN=0123456789 Title=E-commerce Price=$320
Agent 3 : The search result(s) for Shop3 is/are :
 <1> : ISBN=0123456789 Title=E-commerce Price=$350
Agent 4 : The search result(s) for Shop4 is/are :
 <1> : ISBN=0123456789 Title=E-commerce Price=$335
MinPrice Shop=Shop2  Price=320
```

Fig. 5. Compares the offers returned by the master buying agent.


```
ByteEncoder Converting total 10997 bytes.
1000 bytes (9.1%) are encoded successfully.
2000 bytes (18.2%) are encoded successfully.
3000 bytes (27.3%) are encoded successfully.
4000 bytes (36.4%) are encoded successfully.
5000 bytes (45.5%) are encoded successfully.
6000 bytes (54.6%) are encoded successfully.
7000 bytes (63.7%) are encoded successfully.
8000 bytes (72.7%) are encoded successfully.
9000 bytes (81.8%) are encoded successfully.
10000 bytes (90.9%) are encoded successfully.
10996 bytes (100%) are encoded successfully.
The encoding process is completed.
```

Fig. 6. Transferring the digital book to the buying agent.

Fig. 7. Input information for the second demonstration.

where h_j and l_j are the highest and lowest values for the jth attribute. For example, in the case of price, $h_j = 300$ and $l_j = 250$ if the price range is between \$250 and \$300. Note that depending on the attribute, a high value may lead to more or less satisfaction. For instance, in the case of price and discount, a consumer prefers a lower and higher value, respectively. Hence, the satisfaction function can be decreasing [i.e., $F(h_j) < F(l_j)$] or increasing [i.e., $F(h_j) > F(l_j)$], as shown in Fig. 3. In the decreasing case, h_j and l_j result in a satisfaction score of 0 and 1, respectively. In the decreasing case, we have the reverse situation. To determine h_j and l_j, we can ask a consumer two questions such as: "At what value (e.g., price) will you be fully satisfied?" and "At what value (e.g., price) will you be totally dissatisfied?" Based on the answers, the linear satisfaction function can be determined for the concerned attribute. Note that the linear function is just an example. Other functions can also be used.

For purposes of demonstration and evaluation, a prototype system has been developed based on Java Aglet [5] to show some of the features. The current system has been extended and enhanced from the previous work in [10]. Users can forward the shopping requirements to the proxy server via a WAP phone. At each merchant's server, a selling agent is created to communicate with the buying agents, and the product information is kept in a database. Fig. 4 shows the WAP interface at the MAGICS proxy server. In the first demonstration, we consider using MAGICS to purchase a digital book. After the user fills in and confirms the required information (e.g., ISBN = 0123456789 and Name = E-commerce) as shown in Fig. 4, the request is submitted to the proxy server. A master buying agent is then generated to control the buying process for the user. For simplicity, we assume that all of the prospective merchants/retailers are available in the proxy server. In this example, there are four target sellers/retailers. Having identified the four retailers, the master buying agent sends several agents to each of them to obtain their offers. At each seller's server, the buying agent
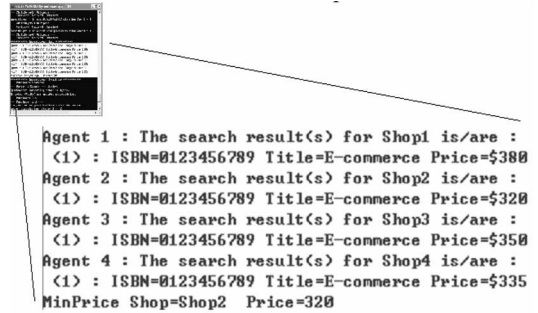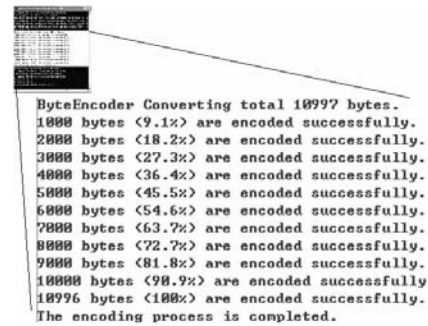
passes a request (in XML format) to the selling agent. Based on that request, the selling agent searches for the product's price in its database, and then provides a reply to the buying agent. After obtaining an offer, a buying agent carries it to the proxy server. After collecting all (four) of the offers and then determining which is the best (i.e., the lowest price in this example, as shown in Fig. 5), the master buying agent sends a purchase agent to the chosen retailer (Shop 2) to purchase the book. At the seller's server, the selling agent provides a copy of the digital book (i.e., an electronic file) for the purchase agent to carry back to the proxy server (see Fig. 6). At the proxy server, the buying agent passes the digital book to the master agent, which stores it in the specified location.

In the second demonstration, we assume that a consumer needs to buy the same book at the nearest physical bookstore. Two attributes, namely price and traveling time to the bookstore, are considered (see Table II). For simplicity, they have equal weights. As shown in Fig. 7, the consumer enters the required

TABLE II
ATTRIBUTES AND THEIR VALUES

Parameter	Value
Attribute Set (A_j)	{price, traveling time}
Min. Price (l_1) (HK\$)	330
Max. Price (h_1) (HK\$)	380
Best Time (l_2) (mins)	5
Worst Time (h_2) (mins)	30

TABLE III
AVERAGE SCORE OF EACH SHOP FOR THE SECOND DEMONSTRATION

Shop	Price ($V_{i,1}$) (\$)	Traveling Time ($V_{i,2}$) (mins)	Attribute Score 1 (S_1)	Attribute Score 2 (S_2)	Score ($\overline{S_i}$)
1	380	23	0	0.285	0.143
2	320	16	1.2 => 1	0.573	0.787
3	350	10	0.6	0.805	0.703
4	335	5	0.9	1.0107 => 1	0.95

information through his/her WAP phone. It can be seen that the consumer is fully satisfied and dissatisfied if the book is priced at \$330 and \$380, respectively. Also, the preferred traveling time is 5 min, and a traveling time of 30 min is unacceptable. Based on the information, four mobile shopping agents are sent to look for the book from a nearby bookstore. After obtaining the offers, the master agent evaluates the offer according to the satisfaction scores as discussed earlier. The evaluation results are shown in Table III. It can be seen that Shop 4 is preferred. Although its offered price is not the lowest, it gives the best satisfaction score after considering both attributes. The previous discussion is just a simple example to demonstrate the basic function of business-to-consumer MAGICS. Many similar applications can be built using this model.

Table IV compares traditional/physical commerce, Web-based commerce, and agent-based commerce systems. It shows that agent-based e-commerce can greatly facilitate the consumer buying process because it offers customers greater convenience. This is particularly useful in an m-commerce scenario.

IV. RELATED PRICE COMPARISON PROBLEM

As discussed in the previous section, one of the key issues in the consumer buying process is the evaluation of alternatives, particularly in terms of price. Agent-based m-commerce allows more efficient price comparison. As an extension to the work in [11], we now investigate an interesting and important price comparison problem. Let us first present a simple example to illustrate the problem. Consider that you want to buy a book through your mobile phone, and you know that the price is between \$400 and \$500 (e.g., it is uniformly distributed). In a purely physical system of commerce, you would visit a number of physical bookstores, one by one (i.e., sequentially), to compare the list prices before buying the book from the cheapest bookstore. In an agent-based m-commerce system, you can search and evaluate (in the digital space) by sending agents to multiple virtual bookstores in parallel before making the purchase. In this case, an interesting problem arises: how many agents should we send if there is a cost for sending an agent (e.g., some network/system resources need to be consumed)?

TABLE IV
COMPARISON OF PHYSICAL COMMERCE, WEB-BASED
COMMERCE, AND AGENT-BASED COMMERCE

	Physical commerce	Web-based commerce	Agent-based Commerce
Search	A human consumer searches for alternatives among physical shops, which can be exhausting and time-consuming.	A human consumer searches through many virtual stores in the global marketspace efficiently via a single interface.	A human consumer delegates searching to an agent. The search may be in parallel using many agents; hence, it is more efficient.
Evaluation	Very often, a human consumer can only evaluate the alternatives by using some simple and subjective methods.	A human consumer can conduct the evaluation with the aid of some supplementary services (e.g., price-comparison services).	A human consumer can delegate the task of evaluation to an agent, which can run more effective and intelligent evaluation algorithms to reach a better decision.
Purchase	A transaction can only be completed by using a physical channel.	A transaction can be completed by using an electronic channel.	An agent can complete a transaction on behalf of its owner and even store the digital goods in a designated location.

Note that if we send too few agents, the best price found may not be attractive enough (e.g., you may only discover a price of \$450 when the lowest price is, in fact, \$430). In contrast, sending too many agents is not cost-effective because many search results may not be useful (e.g., you may get the following results: \$430, \$450, \$450, \$450, etc., so that in fact only one of the search results of \$450 is useful). Somewhere in between there exists an optimum number of agents to be sent. As far as we know, this problem has not been investigated in detail before, so it is of great interest to formulate the problem and solve it. The results should give us valuable insights into the design not only of agent-based m-commerce systems but also of price comparison services in general.

We now formulate this price comparison problem mathematically. Suppose that the cost of sending an agent is c (e.g., an Internet usage charge) and the price distribution function for a particular product is $G(x)$. The objective is to determine how many agents should be sent so as to minimize the overall average cost (i.e., the best price found plus the cost of sending the agents). Note that, in general, the cost c is a design or modeling parameter. It can be implementation-specific such that the service operator can determine c, depending on his/her requirements. It may serve many different purposes. It may represent the consumption of system resources (i.e., the system incurs the cost). It may be related to the processing, transmission, or storage of an agent. In these cases, the server or the network incurs the cost. This cost is particularly applicable to m-commerce because of the bandwidth-limited nature of a wireless channel. While the Internet connection/usage charge is usually fixed, it is

also possible for the Internet service providers to provide value-added services to increase their revenues. One such service may be a price comparison service. From a consumer point of view, the service is attractive because by paying a small charge (e.g., using a micropayment method), consumers can buy the product at a lower price. In this scenario, the cost can be associated with the service charge for such a price comparison service (i.e., the consumer incurs the cost).

Suppose that, upon the return of the nth agent, the probability that the current best price is less than x is $P_n(x)$. Based on $P_n(x)$ and $G(x)$, we can compute the average cost if n agents are sent. Hence, the optimum number of agents can also be found. Compared to the previous work in [11], we consider the case of a continuous rather than a discrete price distribution (i.e., a more realistic price distribution), and we have obtained a closed-form mathematical expression for the situation with a uniform price distribution. The details of the analysis are presented as follows.

It is not difficult to see that $P_n(x)$ and $P_{n-1}(x)$ are related by the following recursive equation:

$$P_n(x) = P_{n-1}(x) + (1 - P_{n-1}(x)) \times G(x). \qquad (3)$$

Equation (3) shows that if the best price upon the return of the nth agent is less than x, there are only two cases. In the first case (i.e., the first term), the best price upon the return of the $(n-1)$th agent is less than/equal to x. Note that in this case, the price returned by the nth agent can be greater than or less than/equal to x. In the second case (i.e., the second term), the best price upon the return of the $(n-1)$th agent is greater than x and the price returned by the nth agent is less than/equal to x.

Equation (3) can be modified as follows:

$$P_n(x) - 1 = (1 - G(x)) \times (P_{n-1}(x) - 1). \qquad (4)$$

It is not difficult to see that

$$P_n(x) = (1 - G(x))^{n-1} \times (P_1(x) - 1) + 1$$
$$= 1 - (1 - G(x))^n. \qquad (5)$$

Having found $P_n(x)$, we can find the average cost if n agents are sent, as follows:

$$C_n = n \times c + \int_a^b p \, dP_n(p) \qquad (6)$$

where a and b are the lowest and highest prices, respectively. Substituting (5) into (6), we obtain the following:

$$C_n = n \times c + \int_a^b p \, dP_n(p)$$
$$= n \times c + \int_a^b p \, d(1 - (1 - G(p))^n)$$
$$= n \times c - \int_a^b p \, d(1 - G(p))^n. \qquad (7)$$

To summarize, if we know the price distribution of a product, we can use (7) to determine the optimal number of agents to send, and, hence, the optimal cost involved in buying the product. In general, we may have some ideas about the mean (m) and variance (v) of the prices of a product. For example, before

buying a book, we may expect that it is about \$350 with a deviation of \$50. Based on the central limit theorem, it should be reasonable to assume that the price distribution follows a normal distribution. Hence, we have

$$G(x) = F(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}\sigma} e^{-(t-\mu)^2/2\sigma^2} dt \qquad (8)$$

where $\mu = m$ and $\sigma = v$ are the mean and variance of the normal distribution. The average cost if n agents are sent is

$$C_n = n \times c - \int_a^b p \, d(1 - G(p))^n$$
$$= n \times c + \int_a^b p \times n \times (1 - G(p))^{n-1} dG(p). \qquad (9)$$

Note that

$$dG(p) = d \int_{-\infty}^p \frac{1}{\sqrt{2\pi}\sigma} e^{-(t-\mu)^2/2\sigma^2} dt$$
$$= \frac{1}{\sqrt{2\pi}\sigma} e^{-(p-\mu)^2/2\sigma^2} dp. \qquad (10)$$

Hence, we have

$$C_n = n \times c + \int_a^b p \times n \times (1 - G(p))^{n-1}$$
$$\times \frac{1}{\sqrt{2\pi}\sigma} e^{-(p-\mu)^2/2\sigma^2} dp. \qquad (11)$$

Based on (11), we can determine the optimal number of agents, n^* numerically. Assume that there are M merchants, we need to send agents to visit n^* of them.

In our model, we assume that the cost is the same for each agent (i.e., irrespective of how many agents are sent). This should be a reasonable assumption because each agent is generated and operated independently. Nevertheless, our model can, in fact, cater for a scenario in which the cost depends on the number of agents sent. In this case, we can simply replace $n \times c$ with a function such as $K(n)$, which represents the cost for sending n agents. The total cost and the optimal number of agents can then be found in a similar manner. In particular, if $K(n)$ gives a decreasing marginal cost, it would be justifiable to send more agents. Hence, the optimal number of agents to be sent should be higher, and the lowest price found will likely be lower. Note that the model can generally be used to find the optimal number of agents as well as the optimal cost for any $K(n)$ (i.e., not just $K(n) = n \times c$) and any price distribution.

It is also of interest to discuss some implementation issues. To offer the price comparison service, the system should know the mean and variance of the price distribution as well as the possible price range for each type of product. This information can be obtained from past statistics and stored in a database. In general, this assumption should be reasonable because a merchant should not sell a product with a price that deviates too much from the market benchmark. The service provider can determine the number of agents based on the statistics. However, one drawback is that all the consumers will be served in the same manner,
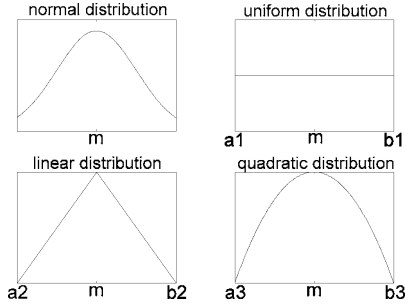
Fig. 8. Four price distributions.

irrespective of their preferences. To provide a customized service, we can allow each consumer to enter the desired price range. In this case, one expects that most consumers will enter a reasonable price range, but at the same time there should be a cost attached to certain behaviors that may otherwise undermine the system, e.g., the entering of unreasonable price ranges. Such a cost may be a minimum usage charge. Alternatively, it may be that if a price is not found within the price range entered by a consumer, no result is returned. Another possible offering may be the provision of a semicustomized service. In this case, the service provider can provide some possible price ranges for the consumers to choose. This would serve the needs of both general consumers and price-sensitive consumers and combines the advantages of the earlier-mentioned approaches in that the service provider can provide a certain degree of customized service while maintaining a large degree of control. Certainly, the price comparison model proposed in this paper offers the opportunity for the development of many such services as well as much scope for future work.

Regarding system efficiency, it is well-known that the distributed and parallel processing capabilities of mobile agent-based systems make them efficient and scalable. In MAGICS, too, high system efficiency is achieved because merchants can process mobile agents in parallel and in a distributed manner. Note that for the price comparison model, the optimal number of agents to be sent depends on the price distribution, not on the number of merchants. For example, if the system determines that n^* agents should be sent, it should choose n^* potential merchants from the search space. This means that if there are more merchants (i.e., the search space is increased), the searching time as well as the processing complexity should not increase dramatically. Even in the worst case (i.e., assuming that all the merchants need to be processed), the complexity is approximately of the order M (i.e., the number of merchants). While for purposes of illustration and discussion, this paper considers the use of just one proxy server, more proxy servers can indeed be set up for load-balancing purposes (i.e., multiple proxy servers can be used to serve consumers). In summary, MAGICS should be efficient and scalable because it uses distributed/parallel processing and load balancing.

Unfortunately, it is computationally intensive to use a normal distribution to calculate the expected cost. In the following, we use three different distributions: uniform, linear, and quadratic to estimate the number of agents. Fig. 8 shows these distributions with the respective maximum and minimum prices

as compared to the normal distribution. As shown later, these three distributions provide a good estimation, in general, while simplifying the calculations. In particular, some close-form expressions can be derived for the uniform distribution to give us valuable insights into the price comparison problem. For each distribution, we choose the maximum and minimum prices such that it can give the same mean and variance of the prices. For example, for the uniform distribution, the mean and variance are $(a_1 + b_1)/2$ and $(b_1 - a_1)^2/12$, respectively. Thus, we have $(a_1 + b_1)/2 = m$ and $(b_1 - a_1)^2/12 = v$. The maximum and minimum prices for the uniform distribution can then be found as follows:

$$a_1 = m - \sqrt{3v}$$
$$b_1 = m + \sqrt{3v}. \tag{12}$$

Similarly, we can get

$$a_2 = m - \sqrt{6v}$$
$$b_2 = m + \sqrt{6v} \tag{13}$$

for the linear distribution and

$$a_3 = m - \sqrt{5v}$$
$$b_3 = m + \sqrt{5v} \tag{14}$$

for the quadratic distribution. Having found the maximum and minimum prices, we can proceed to the calculations as follows.

A. Case 1: Uniform

With the uniform distribution, we have

$$G(x) = F_1(x) = \frac{x - a_1}{b_1 - a_1}. \tag{15}$$

Substituting (15) into (7), we have

$$C_n = n \times c - \int_a^b p \, d(1 - G(p))^n$$
$$= n \times c + b_1 - \frac{n}{n+1} \times (b_1 - a_1). \tag{16}$$

It can be seen that if, as expected, n is large such that $n/(n+1)$ is close to 1, the average cost becomes $a_1 + nc$ (i.e., the minimum price plus the cost of sending n agents). Note also that if one agent is sent, the average cost is $c + (a_1 + b_1)/2$ (i.e., the cost of sending one agent plus the mean price).

To obtain the optimal value of n, we assume that n is a continuous variable. We then differentiate C_n with respect to n to obtain

$$C_n' = c - (b_1 - a_1)\frac{1}{(n+1)^2} = 0. \tag{17}$$

Solving for n, we get the following equation:

$$n = \sqrt{\frac{b_1 - a_1}{c}} - 1. \tag{18}$$

Substituting the optimal value of n into (16), we obtain the optimal average cost as follows:

$$C_n^* = a_1 - c + 2\sqrt{c(b_1 - a_1)}. \tag{19}$$

Hence, by using a uniform distribution, a close-form solution can be obtained. Note that according to (18), the optimal number of agents depends only on $(b_1 - a_1)/c$ (i.e., the difference between the maximum and minimum prices and the cost of sending an agent). Also, if c is greater than $b_1 - a_1$, it is not worth sending any agents, as expected (i.e., we can buy it from any merchant). In practice, it should be reasonable to assume that c is smaller than $b_1 - a_1$, so we will only consider this case in the following analysis.

B. Case 2: Linear

In this case, we have

$$
G(x) = F_2(x) = \begin{cases} \dfrac{2(x - a_2)^2}{(b_2 - a_2)^2} & \text{if } a_2 \le x \le \frac{a_2+b_2}{2} \\ 1 - \dfrac{2(b_2 - x)^2}{(b_2 - a_2)^2} & \text{if } \frac{a_2+b_2}{2} < x \le b_2. \end{cases}
$$
(20)

Therefore, we get

$$
(1 - G(x))^n = \begin{cases} \left(1 - \dfrac{2(x - a_2)^2}{(b_2 - a_2)^2}\right)^n & \text{if } a_2 \le x \le \frac{a_2+b_2}{2} \\ \left(\dfrac{2(b_2 - x)^2}{(b_2 - a_2)^2}\right)^n & \text{if } \frac{a_2+b_2}{2} \le x \le b_2. \end{cases}
$$
(21)

Putting (21) into (7), we get

$$
\begin{aligned}
C_n &= nc + \int_{a_2}^{b_2} x \, dP_n(x) \\
&= nc + \int_{a_2}^{\frac{a_2+b_2}{2}} x \, d(1 - (1 - G(x))^n) \\
&\quad + \int_{\frac{a_2+b_2}{2}}^{b_2} x \, d(1 - (1 - G(x))^n) \\
&= nc - \int_{a_2}^{\frac{a_2+b_2}{2}} x \, d\left(1 - \frac{2(x - a_2)^2}{(b_2 - a_2)^2}\right)^n \\
&\quad - \int_{\frac{a_2+b_2}{2}}^{b_2} x \, d\left(\frac{2(b_2 - x)^2}{(b_2 - a_2)^2}\right)^n.
\end{aligned}
$$
(22)

As shown in Appendix II, it can be calculated that

$$
\begin{aligned}
\int_{\frac{a_2+b_2}{2}}^{b_2} x \, d\left(\frac{2(b_2 - x)^2}{(b_2 - a_2)^2}\right)^n &= 2^n \int_{\frac{a_2+b_2}{2}}^{b_2} x \, d\left(\frac{b_2 - x}{b_2 - a_2}\right)^{2n} \\
&= -\frac{2(n+1)b_2 + 2na_2}{2^{n+1}(2n+1)}.
\end{aligned}
$$
(23)

Let $\frac{x - a_2}{b_2 - a_2} = t$. Then we have $x = (b_2 - a_2)t + a_2$.

We can also find that

$$
\begin{aligned}
\int_{a_2}^{\frac{a_2+b_2}{2}} x \, d\left(1 - \frac{2(x - a_2)^2}{(b_2 - a_2)^2}\right)^n &= \int_0^{\frac{1}{2}} ((b_2 - a_2)t + a_2) \, d(1 - 2t^2)^n \\
&= \frac{a_2 + b_2}{2^{n+1}} - a_2 - (b_2 - a_2)I_n
\end{aligned}
$$
(24)

where $I_n = \int_0^{\frac{1}{2}} (1 - 2t^2)^n \, dt$.

Hence, we have

$$
\begin{aligned}
\frac{1}{2^{n+1}} - I_n &= \int_0^{\frac{1}{2}} x \, d(1 - 2x^2)^n \\
&= 2n \int_0^{\frac{1}{2}} ((1 - 2x^2)^n - (1 - 2x^2)^{n-1}) \, dx \\
&= 2nI_n - 2nI_{n-1}.
\end{aligned}
$$
(25)

Rearranging the terms in (25), we get

$$
I_n = \frac{2nI_{n-1} + \frac{1}{2^{n+1}}}{2n + 1}.
$$
(26)

Based on these, we obtain

$$
\begin{aligned}
C_n &= nc + \int_{a_2}^{b_2} x \, dP_n(x) \\
&= nc + \frac{b_2 - a_2}{2^{n+1}(2n+1)} + a_2 + (b_2 - a_2)I_n.
\end{aligned}
$$
(27)

Hence, the average cost when n agents are sent can be found recursively. Note that $I_0 = \frac{1}{2}$. Furthermore, the optimal number of agents and the minimum cost can then be determined.

C. Case 3: Quadratic

For a quadratic distribution, we have

$$
G(x) = 1 - \frac{(x - b_3)^2(2x + b_3 - 3a_3)}{(b_3 - a_3)^3}.
$$
(28)

This means that

$$
(1 - G(x))^n = \left(\frac{(x - b_3)^2(2x + b_3 - 3a_3)}{(b_3 - a_3)^3}\right)^n.
$$
(29)

The average cost can be found as follows:

$$
\begin{aligned}
C_n &= nc + \int_{a_3}^{b_3} x \, d(1 - (1 - G(x))^n) \\
&= nc - \int_{a_3}^{b_3} x \, d\left(\frac{(b_3 - x)^2(2x + b_3 - 3a_3)}{(b_3 - a_3)^3}\right)^n.
\end{aligned}
$$
(30)

Let $\frac{x - a_3}{b_3 - a_3} = t$. We then have $x = (b_3 - a_3)t + a_3$.

This implies that

$$
\begin{aligned}
C_n &= nc - \int_0^1 ((b_3 - a_3)t + a_3) \, d((1 - t)^2(2t + 1))^n \\
&= nc + a_3 + (b_3 - a_3)Z_{n,n}
\end{aligned}
$$
(31)

where $Z_{n,i} = \int_0^1 (1 - x)^{3n-i}(2x + 1)^i \, dx$.

It can be found that

$$
\begin{aligned}
Z_{n,i} &= \frac{-1}{3n - i + 1} \int_0^1 (2x + 1)^i \, d(1 - x)^{3n-i+1} \\
&= \frac{-1}{3n - i + 1} \left((2x + 1)^i(1 - x)^{3n-i+1}\Big|_0^1 \right. \\
&\quad \left. - \int_0^1 (1 - x)^{3n-i+1} \, d(2x + 1)^i \right)
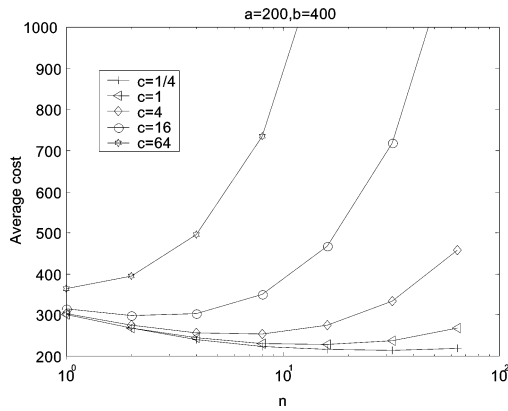\end{aligned}
$$

Fig. 9. Average cost when different numbers of agents are sent (uniform price distribution).

Fig. 10. Optimal number of agents to be sent when $a = 200$ (uniform price distribution).

$$= \frac{1}{3n - i + 1} + \frac{2i}{3n - i + 1} Z_{n,i-1}. \tag{32}$$

Similar to the linear distribution, we can obtain C_n recursively and, hence, the optimal n and the minimum cost. Note that $Z_{n,0} = \frac{1}{3n+1}$.

The detailed calculations for (16), (23)–(25), and (31) are shown in the Appendixes. It is also of interest to briefly discuss the business implications or issues. In general, the price of a product should be set based on a basic cost and a desired profit margin and different businesses may set different prices for the same product, depending on their requirements. Having said that, one would not expect the price of a product to greatly deviate from a market benchmark. This suggests that we can model prices with a price distribution function. The uniform price distribution is the simplest and easiest to use, particularly because it can give a close-form solution. Its limitation is that it can be used to model only simple products and provides only a rough estimate, in general. For most products, the best estimate should be provided by the normal price distribution, given the well-known central limit theorem, especially when the model involves a large number of merchants. Unfortunately, a normal price distribution is more computationally intensive; so, we make use of the linear price distribution and quadratic price distribution. Fortunately, based on the results in the paper, it is found that generally speaking all four distributions give similar results. Note that the price comparison model can be applied to any price distribution; so, these four price distributions are simply examples. Nevertheless, they should be useful for modeling many or even most products.

Lastly, we present and discuss some analytical results as follows. As the uniform price distribution can provide some close-form expressions, it is of interest to analyze this distribution in detail. To simplify the presentation, the subscripts for a and b are dropped in Figs. 9–12. Fig. 9 shows the average cost when different numbers of agents are sent. In this case, $a = 200$ and $b = 400$. It can be seen that there exists an optimal number of agents to send in order to achieve the lowest cost. The optimal number of agents depends on c. As expected, as c decreases, more agents should be sent. For example, if $c = 4$, about six agents should be sent and the optimal average cost is about
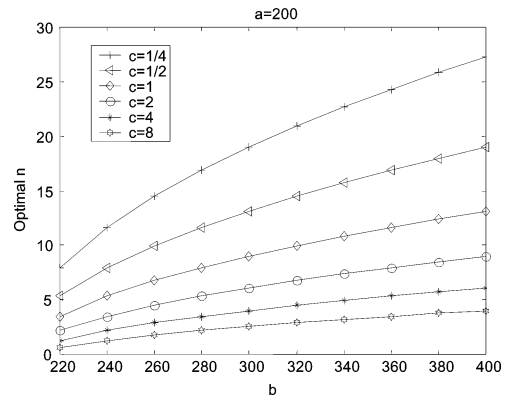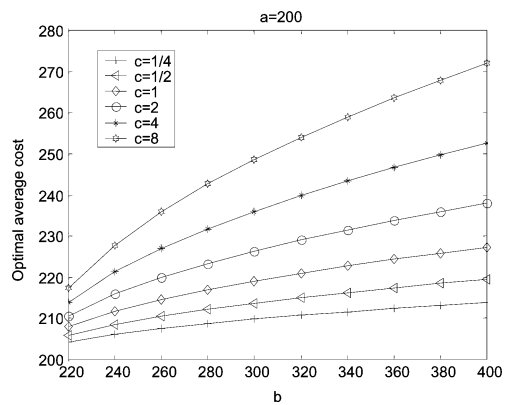

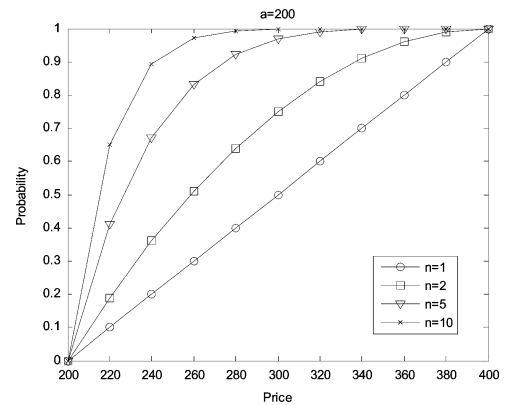Fig. 11. Optimal average cost when $a = 200$ (uniform price distribution).

Fig. 12. $P_n(x)$ when $a = 200$ (uniform price distribution).

$253. This means that, excluding the cost of sending the agents (i.e., $24), the best price found is about $(253 - 24) = 229, which is $29 above the lowest price of $200. Fig. 10 shows the optimal number of agents for different values of b when a is 200. As b increases, it is worth sending more agents. When c is smaller, the optimal value of n increases at a faster rate. Fig. 11 shows the optimal average cost when a is 200 and b varies between 220 and 400. It can be seen that when c is smaller, the optimal average cost is (relatively) less sensitive to changes in
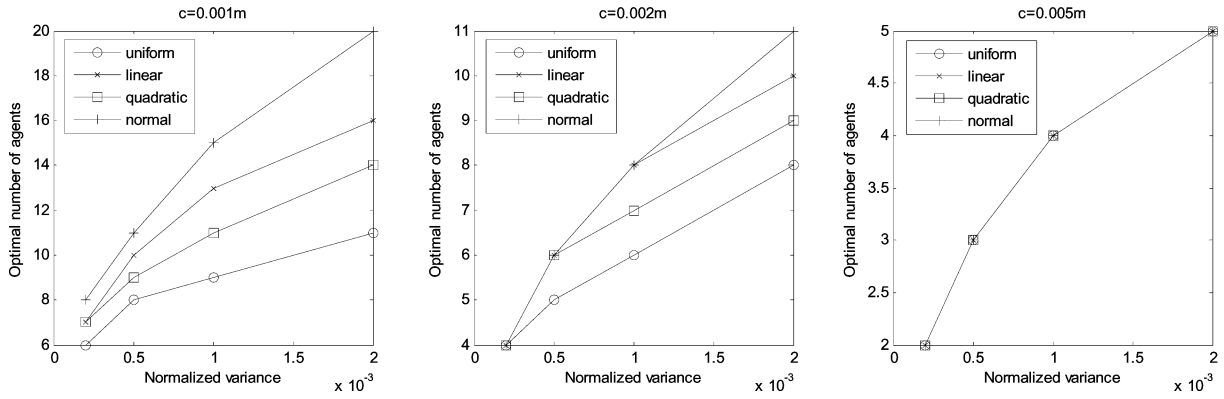
Fig. 13.   Optimal number of agents for different normalized variances.
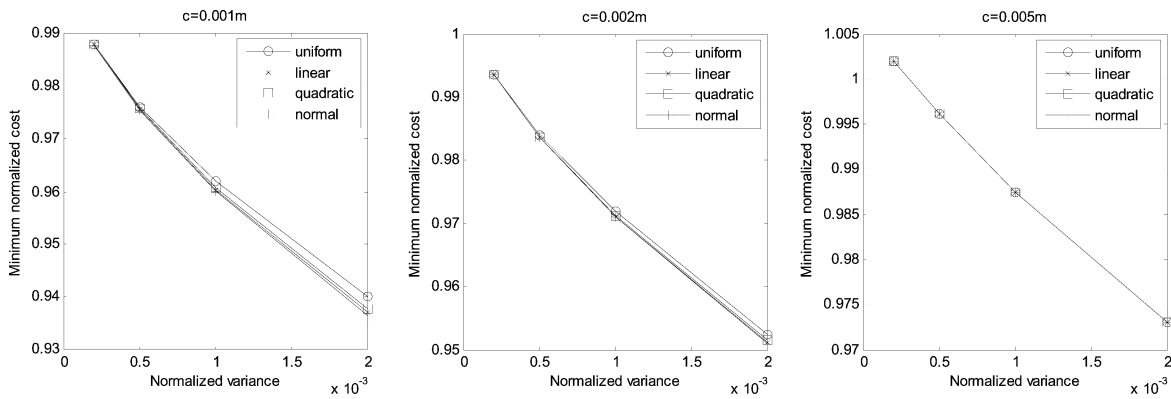

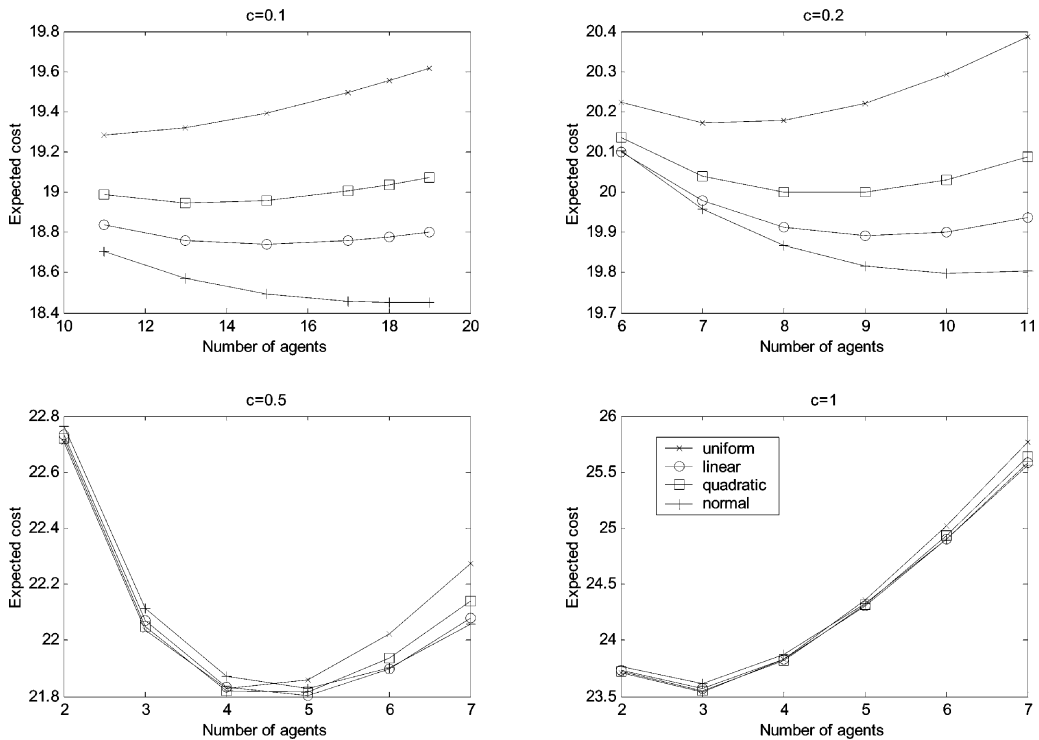Fig. 14.   Minimum normalized cost for different normalized variances.

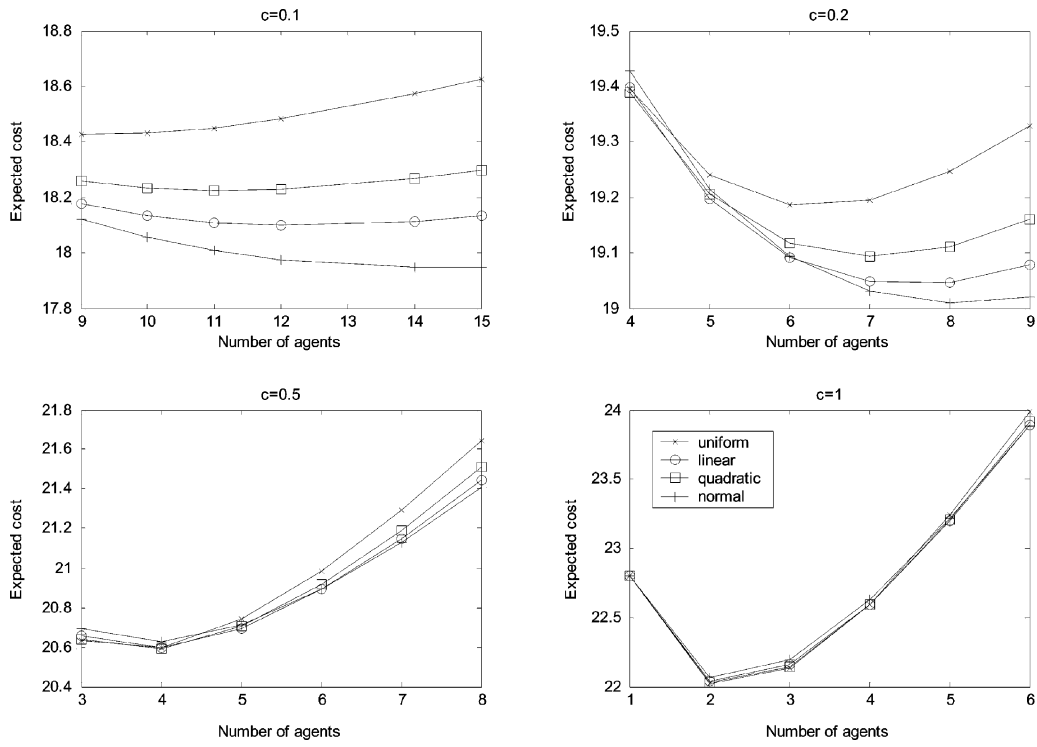Fig. 15.   Expected cost of purchasing the book *Harry Potter and the Order of the Phoenix* (prices based on the information at http://www.bestwebbuys.com).

Fig. 16.   Expected cost of purchasing the book *The Future of Freedom: Illiberal Democracy at Home and Abroad* (prices based on the information at http://www.bestwebbuys.com).
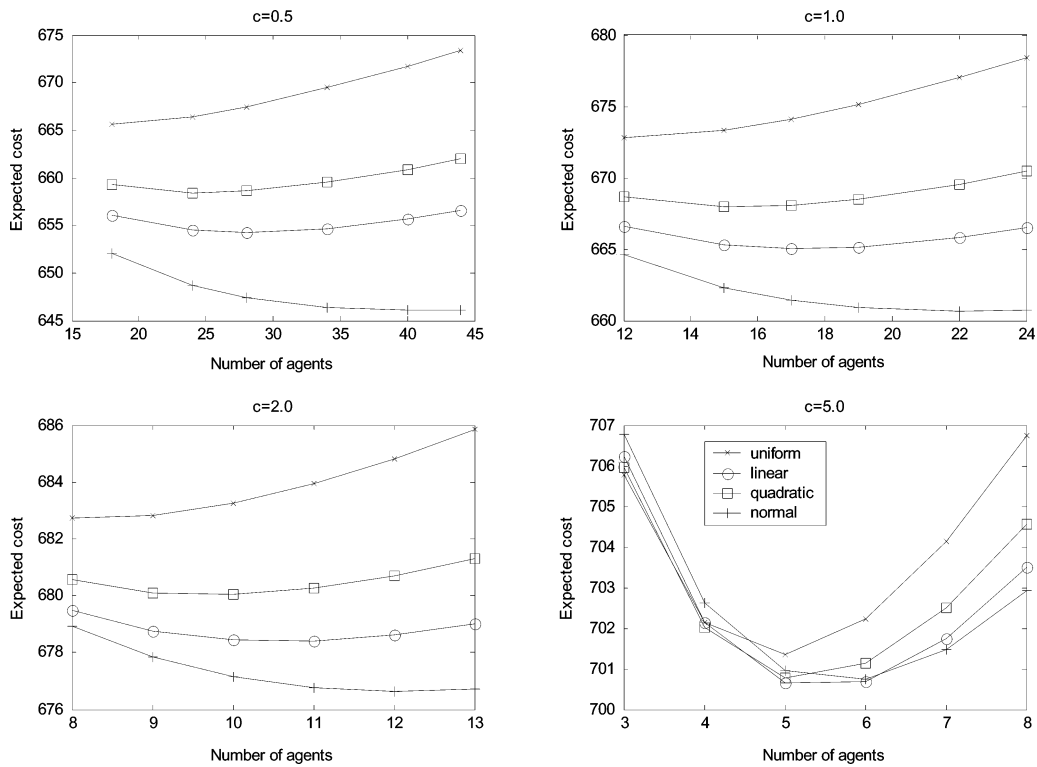

Fig. 17.   Expected cost of purchasing the digital camera *Canon Powershot G5* (prices based on the information at http://www.bestwebbuys.com).
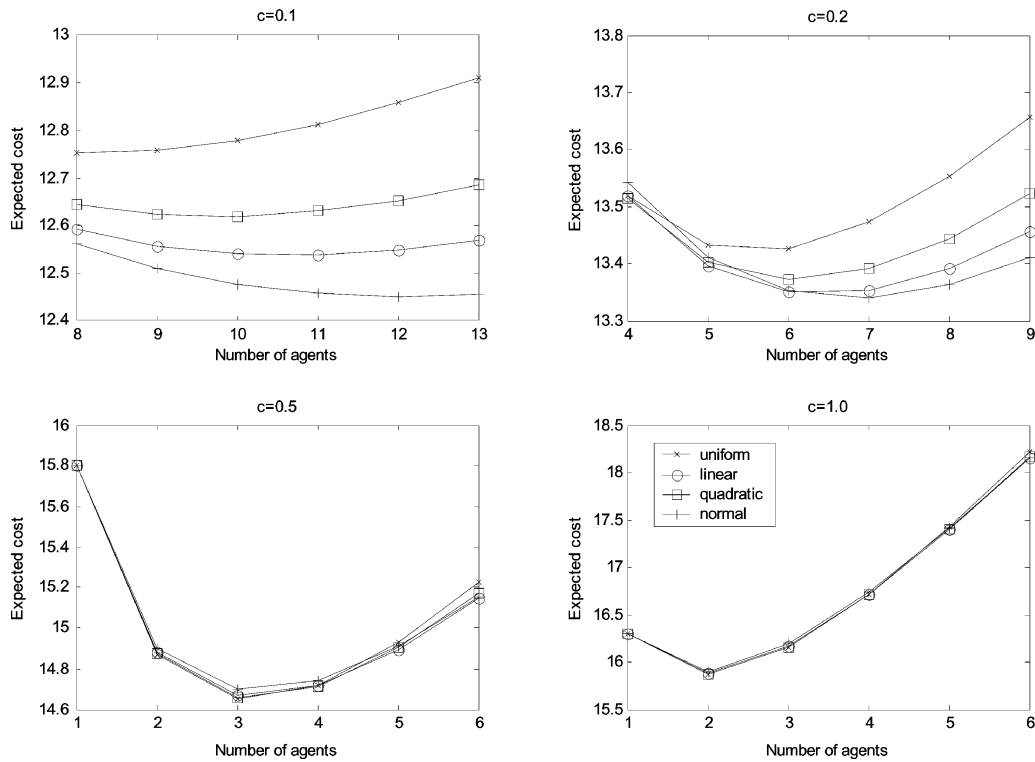
Fig. 18.   Expected cost of purchasing the audio CD *Mission to Mars (2000)* (prices based on the information at http://www.bestwebbuys.com).

Goods	Mean price ($)	Variance ($)
Book 1: "Harry Potter and the Order of the Phoenix"	24.0578	4.0684
Book 2: "The Future of Freedom: Illiberal Democracy at Home and Abroad"	21.8020	3.0846
Digital Camera: Canon Powershot G5	734.0788	49.9906
Audio CD: "Mission to Mars (2000)"	15.3038	2.4873

b. It is also of interest to see how $P_n(x)$ changes as n increases. As shown in Fig. 12, it can be seen that as more agents are sent, the curve moves upward, which indicates that there is a higher probability of obtaining a lower price.

Next, we compare the analytical results when different price distributions are used. Fig. 13 shows the optimal number of agents to be sent for different variances. As an example, we set $m = 100$ in Figs. 13 and 14. Furthermore, all costs/prices and variances are normalized with respect to m and m^2, respectively. We have actually obtained another set of results with $m = 1000$. It was found that the results are the same as those presented later (i.e., the normalized values are the same or do not depend on m). As expected, it can be seen from Fig. 13 that the optimal number of agents increases as the variance increases. The optimal number of agents as calculated by the normal price distribution is the largest. Among the three price distributions,

the optimal number of agents as computed by linear distribution is closest to that calculated by normal distribution. However, the difference is not large unless the variance is large. Also, if the cost of sending an agent increases, all of the four distributions give almost the same number of agents. Fig. 14 shows the minimum normalized cost (i.e., with respect to the mean price) when the variance is varied. In this case, the minimum normalized cost is computed using the normal price distribution based on the optimal number of agents as determined by the respective price distribution. The figure indicates that the average cost decreases as the variance increases. This is because if the variance is large, it is more likely to find a lower price by sending out more agents. It is interesting to find that all of the four distributions give nearly the same average cost, especially if the cost of sending an agent is high. We have also conducted further analysis by using some real price information as obtained from Best Web Buys (www.bestwebbuys.com). Fig. 15 shows the expected cost of purchasing the book *Harry Potter and the Order of the Phoenix* when different numbers of agents are sent and the four distributions are used. Note that all of the four distributions have the same mean and variance based on the price information obtained from Best Web Buys (see Table V). In this case, the expected cost is computed based on the respective price distribution. It can be seen that the normal price distribution gives the lowest cost. However when the cost of sending an agent increases, the expected costs of all of the distributions are similar. Figs. 16–18 show the results for buying another book, a digital camera, and a compact disc. The results are generally quite similar. In summary, the proposed model can be employed

to determine the best number of agents to be sent for the purpose of comparing prices. Also, for performing the computation, the uniform, linear, and quadratic price distributions can provide a good estimation of the normal price distribution, in general. In particular, the linear price distribution gives the closest estimate.

V. CONCLUSION

A mobile agent-based system has been presented for facilitating consumer-oriented e-commerce or m-commerce. A distributed approach is employed through which merchants can set up special servers to serve mobile agents at their own sites. Using a WAP/Web interface (e.g., a mobile phone), a consumer can provide the buying requirements to a proxy server. This proxy server then generates mobile agents to conduct the necessary shopping tasks. A basic buying protocol has been presented to support the consumer buying process. In particular, a mathematical model has been developed to evaluate multiple decision factors so as to find the retailer that can best fulfill a consumer's requirements. A prototype system has been developed to demonstrate a sales transaction under an m-commerce scenario. We have also formulated an analytical model to calculate the optimal number of agents to be sent when comparing the prices of a product. A closed-form solution was obtained for the case with a uniform price distribution. Furthermore, we have presented and discussed some analytical results based on four different price distributions and some real price information.

APPENDIX I

Detailed derivation of (16) is as follows:

$$C_n = n \times c - \int_{a_1}^{b_1} p \, d \left(1 - G(p)\right)^n$$

$$= n \times c + \int_{a_1}^{b_1} p \, d \left(1 - \left(\frac{b_1 - p}{b_1 - a_1}\right)^n\right)$$

$$= n \times c + p \times \left(1 - \left(\frac{b_1 - p}{b_1 - a_1}\right)^n\right)\Big|_{a_1}^{b_1}$$

$$- \int_{a_1}^{b_1} \left(1 - \left(\frac{b_1 - p}{b_1 - a_1}\right)^n\right) dp$$

$$= n \times c + b_1 + (b_1 - a_1)$$

$$\times \left(-1 - \frac{1}{n+1} \times \left(\frac{b_1 - p}{b_1 - a_1}\right)^{n+1}\Big|_{a_1}^{b_1}\right)$$

$$= n \times c + b_1 - \frac{n}{n+1} \times (b_1 - a_1). \tag{33}$$

APPENDIX II

Detailed derivation of (23) is as follows:

$$\int_{\frac{a_2+b_2}{2}}^{b_2} x \, d \left(\frac{2(b_2-x)^2}{(b_2-a_2)^2}\right)^n$$

$$= 2^n \int_{\frac{a_2+b_2}{2}}^{b_2} x \, d \left(\frac{b_2-x}{b_2-a_2}\right)^{2n}$$

$$= 2^n x \left(\frac{b_2-x}{b_2-a_2}\right)^{2n}\Big|_{\frac{a_2+b_2}{2}}^{b_2} - 2^n \int_{\frac{a_2+b_2}{2}}^{b_2} \left(\frac{b_2-x}{b_2-a_2}\right)^{2n} dx$$

$$= 2^n \left(0 - \frac{a_2+b_2}{2}\left(\frac{1}{2}\right)^{2n}\right)$$

$$+ 2^n (b_2-a_2)\frac{1}{2n+1}\left(0 - \left(\frac{1}{2}\right)^{2n+1}\right)$$

$$= -\frac{a_2+b_2}{2^{n+1}} - \frac{b_2-a_2}{(2n+1)\,2^{n+1}}$$

$$= -\frac{2(n+1)b_2 + 2na_2}{2^{n+1}(2n+1)}. \tag{34}$$

APPENDIX III

Detailed derivation of (24) is as follows:

$$\int_{a_2}^{\frac{a_2+b_2}{2}} x \, d \left(1 - \frac{2(x-a_2)^2}{(b_2-a_2)^2}\right)^n$$

$$= \int_0^{\frac{1}{2}} ((b_2-a_2)t + a_2)\, d \left(1 - 2t^2\right)^n$$

$$= (b_2-a_2)\int_0^{\frac{1}{2}} t \, d \left(1 - 2t^2\right)^n + \left(\frac{a_2}{2^n} - a_2\right)$$

$$= (b_2-a_2)\left(\frac{1}{2^{n+1}} - I_n\right) + \left(\frac{a_2}{2^n} - a_2\right)$$

$$= \frac{a_2+b_2}{2^{n+1}} - a_2 - (b_2-a_2)I_n. \tag{35}$$

APPENDIX IV

Detailed derivation of (25) is as follows:

$$\frac{1}{2^{n+1}} - I_n = \int_0^{\frac{1}{2}} x \, d \left(1 - 2x^2\right)^n$$

$$= n \int_0^{\frac{1}{2}} x \left(1 - 2x^2\right)^{n-1}(-4x)\, dx$$

$$= -4n \int_0^{\frac{1}{2}} x^2 (1 - 2x^2)^{n-1} dx$$

$$= 2n \int_0^{\frac{1}{2}} \left((1 - 2x^2)^n - (1 - 2x^2)^{n-1}\right) dx$$

$$= 2nI_n - 2nI_{n-1}. \tag{36}$$

APPENDIX V

Detailed derivation of (31) is as follows:

$$C_n = nc - \int_0^1 ((b_3 - a_3)t + a_3)\, d((t-1)^2(2t+1))^n$$

$$= nc - a_3((t-1)^2(2t+1))^n \Big|_0^1$$

$$- (b_3 - a_3) \int_0^1 t \, d((1-t)^2(2t+1))^n$$

$$= nc + a_3 - (b_3 - a_3)t(1-t)^{2n}(2t+1)^n \big|_0^1$$

$$+ (b_3 - a_3) \int_0^1 (1-t)^{2n}(2t+1)^n \, dt$$

$$= nc + a_3 + (b_3 - a_3)Z_{n,n}. \tag{37}$$

ACKNOWLEDGMENT

REFERENCES

[1] R. Kalakota and A. B. Whinston, *Electronic Commerce: A Manager's Guide*. Reading, MA: Addison-Wesley/Longman, 1997.

[2] H. Chan, R. Lee, T. Dillon, and E. Chang, *E-Commerce: Fundamentals and Applications*. Chichester, U.K.: Wiley, 2001.

[3] U. Varshney and R. Vetter, "Mobile commerce: Frameworks, applications and networking support for mobile commerce," *ACM/Kluwer J. Mobile Netw. Appl.*, pp. 185–198, Jun. 2002.

[4] R. Kalakota and M. Robinson, *M-business: The Race to Mobility*. New York: McGraw-Hill, 2002.

[5] D. B. Lange and M. Oshima, *Programming and Deploying Java Mobile Agents with Aglets*. Reading, MA: Addison-Wesley/Longman, 1998.

[6] P. Dasgupta, N. Narasimhan, L. E. Moser, and P. M. Melliar-Smith, "MAgNET: Mobile agents for networked electronic trading," *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 4, pp. 509–525, Jul.–Aug. 1999.

[7] M. He, N. R. Jennings, and H.-F. Leung, "On agent-mediated electronic commerce," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 4, pp. 985–1003, Jul.–Aug. 2003.

[8] A. G. Moukas, G. Zacharia, R. H. Guttman, and P. Maes, "Agent-mediated electronic commerce: An MIT Media Laboratory perspective," *Int. J. Electron. Com.*, vol. 4, no. 3, pp. 5–22, Spring 2000.

[9] R. H. Guttman, A. G. Moukas, and P. Maes, "Agent-mediated electronic commerce: A survey," *Knowl. Eng. Rev.*, vol. 13, no. 2, pp. 147–159, 1998.

[10] H. C. B. Chan, B. Lam, and R. S. T. Lee, "MAGICS shopper," in *Proc. IC 2001*, Jun. 2001, vol. 2, pp. 1062–1068.

[11] H. C. B. Chan, C. K. H. Chin, and B. Lam, "price comparison agents for MAGICS," in *Proc. PACRIM 2001*, Aug. 2001, vol. 2, pp. 744–747.

[12] M. J. Etzel, B. J. Walker, and W. J. Stanton, *Marketing*, 12th ed. New York: McGraw-Hill/Irwin, 2001.

[13] A. Chavez and P. Maes, "Kasbah: An agent marketplace for buying and selling goods," in *Proc. 1st Int. Conf. Pract. Appl. Intell. Agents MultiAgent Technol.*, London, Apr. 1996, pp. 75–90.

[14] T. Sandholm and Q. Huai, "Nomad: Mobile agent system for an Internet-based auction house," *IEEE Internet Comput.*, vol. 4, no. 2, pp. 80–86, Mar.–Apr. 2000.

[15] J. M. Andreoli, F. Pacull, and R. Pareshi, "XPect: A framework for electronic commerce," *IEEE Internet Comput.*, vol. 1, no. 4, pp. 40–48, Jul.–Aug. 1997.

[16] C. M. Jonker and J. Treur, "An agent architecture for multi-attribute negotiation," in *Proc. IJCAI 2001*, pp. 1195–1201.

[17] J. Ueyama and E. R. M. Madeira, "An automated negotiation model for electronic commerce," in *Proc. 5th Int. Symp. Auton. Decentralized Syst.*, 2001, pp. 29–36.

[18] M. Chung and V. Honavar, "A negotiation model in agent-mediated electronic commerce," in *Proc. Int. Symp. Multimedia Softw. Eng.*, 2000, pp. 403–410.

Hui Chen was born in China in 1977. He received the B.S. and M.S. degrees in computer science and technology from Nanjing University, Nanjing, China, in 1999 and 2002, respectively. He is currently working toward the Ph.D. degree in electrical and computer engineering at the University of British Columbia, Vancouver, BC, Canada.

From 2000 to 2003, he was a Research Assistant in the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. His current research interests include modeling and cross-layer optimization of packet-switched third-generation wireless networks.
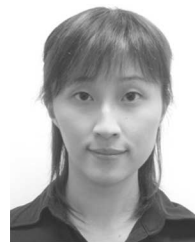
Perry P. Y. Lam received the B.Sc. (Hons.) degree in information technology in 2002 and the M.Phil. degree in 2005, both from The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.

Her current research interests include electronic commerce and Internet technologies.

Henry C. B. Chan (M'95) received the B.A. and M.A. degrees from the University of Cambridge, Cambridge, U.K., and the Ph.D. degree from the University of British Columbia, Vancouver, BC, Canada.

He is currently an Associate Professor in the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. From 1988 to 1993, he was with Hong Kong Telecommunications Ltd., From 1997 to 1998, he worked with BC TEL Advanced Communications, Canada. His current research interests include networking/communications, wireless networks, Internet technologies, and electronic commerce. He is the author/coauthor of more than 50 journal/conference papers and is the coauthor of a textbook on electronic commerce.

Dr. Chan is a member of the Association for Computing Machinery and the Insitution of Engineering and Technology.

Tharam S. Dillon (M'83–SM'87–F'98) received the B.E. (Hons) and Ph.D. degrees from Monash University, Australia.

He is currently a Distinguished Research Professor with the Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, Perth, WA, Australia. His current research interests include Web semantics, ontologies, Internet computing, e-commerce, hybrid neurosymbolic systems, neural nets, software engineering, database systems, and computer networks. He is the author or coauthor of more than 650 papers published in international and national journals and is the author of five books. He is the Editor-in-Chief of the *International Journal of Computer Systems Science and Engineering* as well as *Engineering Intelligent Systems*. He is the Co-Editor of the *Journal of Electric Power and Energy Systems*.

Jiannong Cao (M'93–SM'05) received the B.Sc. degree in computer science from Nanjing University, Nanjing, China, in 1982, and the M.Sc. and Ph.D. degrees in computer science from Washington State University, Pullman, in 1986 and 1990, respectively.

He is currently a Professor in the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. He is also the Director of the Internet and Mobile Computing Laboratory in the department. Prior to this, he was with James Cook University, Queensland, Australia, the University of Adelaide, Adelaide, Australia, and the City University of Hong Kong, Kowloon, Hong Kong. His current research interests include parallel and distributed computing, networking, mobile and wireless computing, fault tolerance, and distributed software architecture. He is the author or coauthor of more than 180 papers published in international journals.

Prof. Cao is a Senior Member of the China Computer Federation, the IEEE Computer Society, the IEEE Communications Society, and the Association for Computing Machinery. He has served as a member of editorial boards of several international journals, a reviewer for international journals/conference proceedings, and also as an organizing committee member for several international conferences.

Raymond S. T. Lee (M'98) received the B.Sc. degree from The Hong Kong University, Hong Kong, in 1989, and the M.Sc. and Ph.D. degrees from The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, in 1997 and 2000, respectively.

From 1989 to 1993, he was a Meteorological Scientist with the Hong Kong Government in the Hong Kong Observatory, where he was engaged in weather forecasting and developing meteorological telecommunication information systems. Between 1998 and 2006, he was with The Hong Kong Polytechnic University. He has also been a Management Information Systems Manager and a System Consultant in Hong Kong. He is currently the President of IATOPIA Technology Ltd., Kowloon, which aims at the provision of intelligent business solutions. He is also the Founder and the Director of the IATOPIA Research Laboratory. His current research interests include intelligent agent technology, artificial intelligence, agent ontology, chaotic neural networks, neural oscillators, pattern recognition, visual perception and visual psychology, and weather simulation and forecasting.

Dr. Lee is a member of the Association for Computing Machinery.