Auctioning Vertically Integrated Online Services: Computational Approaches for Real-Time Allocation

RAVI BAPNA, PAULO GOES, AND ALOK GUPTA

RAVI BAPNA is an Associate Professor of Information Systems at the Carlson School of Management, University of Minnesota as well as the Executive Director of the Centre for Information Technology and the Networked Economy (CITNE) at the Indian School of Business. His research interests are in the areas of online auctions, e-market design, grid computing, and the economics of information systems. His research has been extensively published in a wide array of journals such as *Management Science, INFORMS Journal on Computing*, *Statistical Science, Information Systems Research, Journal of Retailing, MIS Quarterly, Decision Sciences, Communications of the ACM, Naval Research Logistics, Decision Support Systems,* and *European Journal of Operations Research.* He was recently awarded the Ackerman Scholar Award for outstanding research at the University of Connecticut. He contributes regularly to popular press outlets such as the *Financial Times*, *India Knowledge @ Wharton, Economic Times,* and *Business Today.*

PAULO GOES is the Salter Professor and Head of the MIS Department at Eller College of Management, University of Arizona. He received his Ph.D. from the University of Rochester. His research interests are in the areas of design and evaluation of models for e-business, emerging technologies, online auctions, database technology and systems, and technology infrastructure. His research has appeared in several journals, including *Management Science, Information Systems Research, Journal of Management Information Systems, MIS Quarterly, Operations Research, INFORMS Journal on Computing, IEEE Transactions on Communications,* and *IEEE Transactions on Computers.* Dr. Goes is senior editor of *Information Systems Research,* and is or has recently been associate editor of *Management Science, Journal of Management Information Systems*, *Decision Sciences, Production and Operations Management,* and *INFORMS Journal on Computing*. He cochaired the Workshop on Information Technology and Systems (WITS) in 2004 and the Statistical Challenges in E-Commerce Research Symposium (SCECR) in 2007. He was recently elected the WITS Organization President.

ALOK GUPTA is Carlson School Professor and Chairman of the Information and Decision Sciences (IDSc) Department at Carlson School of Management, University of Minnesota. His areas of specialization include process management, data communication, electronic commerce, design and evaluation of electronic mechanisms and processes, mathematical modeling of information systems, large-scale systems simulation, and economics of information systems. His research has been published in all top journals

Author names are in alphabetical order.

Journal of Management Information Systems / Winter 2008–9, Vol. 25, No. 3, pp. 65–97.

0742–1222 / 2009 $9.50 + 0.00.
DOI 10.2753/MIS0742-1222250302

in the field, including *Management Science, Information Systems Research, Journal of Management Information Systems, MIS Quarterly, Decision Sciences,* and *INFORMS Journal on Computing.* He was a recipient of the prestigious NSF CAREER Award for his research in online auctions. He serves on the editorial boards of several academic journals such as *Management Science*, *Information Systems Research, Journal of Management Information Systems,* and *DSS*. He also holds the position of publisher of *MIS Quarterly,* the top-rated journal in the field of MIS.

ABSTRACT: We develop three auction-based pricing and allocation solution methods for the case where a capacity-constrained online service provider offers multiple classes of unique, one-time services with differentiated quality. Consumers desire exactly one of the many service classes offered. We call such a setting a vertically integrated online services market. Examples of these services are webcasting of special events over the Internet, provision of video-on-demand, and allocation of grid computing resources. We model the pricing and allocation decision faced by firms in such a setting as a knapsack problem with an added preference elicitation dimension. We present a variety of computational solution approaches based on adaptations of the traditional greedy heuristic for knapsack problems. The solution approaches vary in efficacy depending on whether bidders are restricted to bid in one service class or allowed to bid in multiple service classes, as well as on the overall variability of the demand. In the case bidders can bid in multiple classes but are interested in consuming only one service class, a direct application of the heuristics developed for the single service case results in a nonfair allocation. We develop a novel data structure to eliminate the unfair allocation while maintaining the original computation complexity of the simpler setting. The paper contributes by presenting a menu of auction clearing mechanisms for selling vertically integrated online services.

KEY WORDS AND PHRASES: auction-based pricing, online services, service classes, service computing, service pricing.

WE CONSIDER THE PROBLEM OF PRICING and providing vertically integrated online services delivered through the Internet. A vertically integrated service setting arises when consumers may have interest in several similar services, differing perhaps only in their quality levels, but are interested in consuming only one service. An example of vertically integrated services is the interactive *webcasts* of concerts, high-profile interviews, and sporting events such as international soccer and cricket matches, offered in text, audio, or full audio/video in a variety of bit rates, or even interactive format. Typically, vertically integrated products are offered to differentiate and capture a wider market base by creating different quality versions of the same product [10, 20]. For such multimedia content-type services, we assume that the capacity provider is also the content owner and distributor. If that is not the case, we can maintain the structure of our results by simply assuming that the capacity provider factors in the appropriate licensing fees in his or her reserve price.

While immense strides have been made in developing and utilizing technical advancements of such technology, the analysis and understanding of dynamic pricing and allocation approaches required to facilitate emerging markets for vertically integrated digital services are still in nascent stages.

We consider the case where a consumer would like to purchase only one of multiple quality differentiated services that are offered based on his or her preferences and budget. Despite recent technological advances, the lack of proper mercantile processes and associated preference elicitation mechanisms still precludes multiple quality differentiated services being widely offered. Typically, a single (best-effort) level of service is the only option available for consumers even when the desired quality of service could potentially be delivered in a given service class. Further, the wide uncertainty in demand for unique one-time services coupled with the lack of economic incentives at the provider's end makes it difficult to justify procuring server capacity to serve all customers at the desired service level.

We propose an auction mechanism and develop a series of solution methodologies that allows a service provider to price and allocate multiple services, each at multiple quality levels, in a way that allocates the computational resources fairly to its customers.

We expect our approach to be applicable to a variety of other digital service markets. These include:

1. *Computational Grids and Peer-to-Peer (P2P) Networks*—Consider the case of a commercial storage grid operated by Kontiki.com. The grid operator utilizes hard drive storage on networked PCs owned by consumers. The total capacity available to sell is variable over time and the consumers may have "hard needs" for immediate storage as well as some "soft needs" based on anticipated usage. The flexibility in consumers' needs creates a vertically integrated market where total storage capacity available is constrained and the total demand at a given time establishes an appropriate clearing price. Similarly, consider a P2P bandwidth sharing market that can be facilitated by technologies such as BitTorrent (www.bittorrent.com). The BitTorrent framework uses a part of a consumer's upstream bandwidth in exchange for downloading shared content. An auction-based vertically integrated market may be facilitated in this environment when a consumer might be willing to share variable amount of upstream bandwidth depending on the price [3, 13]. Bapna et al. [1] discuss how the bids can be specified in such a setting. The downloaders can bid for different bundles of data rates or, in the case of Kontiki.com, different amounts of fixed and variable storage they are renting. The capacity constraint corresponds to the total amount of storage and bandwidth a sharer is willing to provide to the P2P network.
2. *Business-to-Business (B2B) Video-on-Demand (VOD) Content Shelf Space*—The issue of optimally managing the server capacity, or the "content shelf space" as it is referred to in the VOD industry, is also very important for firms that develop and deploy VOD servers in digital cable markets. Firms in these

markets serve as intermediaries between content providers, such as ESPN and Disney, and consumers in specialized, high-bandwidth digital cable markets. In today's environment, such firms have to consider what shows to carry in HDTV, with its variety of resolution levels and what to carry in SDTV format, and what to carry in both formats. This creates the vertically integrated service structure in this B2B market. The VOD server's capacity is the critical link in maintaining the desirable service quality level with broadband broadcast technologies.

3. *Asynchronous Transfer Mode (ATM) Networks*—In an ATM network, quality of service refers to specific traffic-handling parameters that are adhered to for a given circuit. Essentially, it allows each individual circuit and traffic type to receive the support it needs, hence allowing voice, video, and data traffic integration over a single network. Because of the inherent flexibility provided by the virtual circuit concept, managing an ATM network efficiently is a challenge because of the enormous number of choices available for setting the various operational parameters. Although time- and service-sensitive accounting and billing are more appealing to customers, carriers and service providers tend to provide "flat rate" ATM services, which typically lead to an inefficient allocation. Our work can be used to derive an "optimal" bandwidth allocation plan together with a pricing system, for real-time delivery of services such as videoconferencing and other digital streams.

Table 1 summarizes the two key components of the problem structure for the above-mentioned examples. Entries in this table exemplify vertically integrated services where a service provider offers multiple service quality choices. Users select one of these. Each row also specifies the corresponding resource constraint.

We develop a series of computational methods that can be used by providers offering vertically integrated service to achieve the following:

1. a *demand collection* mechanism that allows the customers (consumers or other firms) to bid for the services,
2. a *pricing* mechanism that determines the final prices at each service level,
3. a *capacity allocation* mechanism that allocates the necessary server resources to customers in each service level,
4. the determination of the *service mix* to be provided with guaranteed quality levels at the server side, and
5. an optimization mechanism to *maximize the total revenue* from the available service mix for a given server capacity.

We designate (1)–(5) as a real-time computational infrastructure for revenue-maximizing content providers. For analytical tractability, we assume that there are no network delays and last mile problems. For instance, content providers may have caching arrangements with companies such as Akamai to push their content to the edge of the network. Our pricing scheme allocates the available server capacity among various competing services requiring different quality metrics. Note that in an environment where quality of service is not important, a best-effort model suffices and pricing of

Table 1. Resource-Constrained Vertically Integrated Services Arise in a Variety of Settings

Setting	Nature of vertically integrated services	Resource constraint
Webcasting of live events	Different bit rate video streams, audio streams, or textual commentary	Streaming server's capacity to serve simultaneous connections
P2P file sharing	Variable upstream bandwidth a user is willing to share	P2P user's storage and bandwidth
Video-on-demand content shelf space	Distribution of HDTV, SDTV shows to be carried by syndicators for resale	Video-on-demand server streaming capacity
ATM networks	Voice, video, and data traffic services	Circuit capacity

services is not an important consideration. We also do not consider a time dimension to the demand. These will serve as natural extensions in the future.

We formulate such a resource allocation problem as a *knapsack* problem with two additional constraints. Users have values for the services offered and desire to be assigned a single service that consumes an exogenously specified amount of capacity. The total number of services, and the service mix, offered is constrained by the server's overall capacity. The first additional constraint imposes the structure of an *auction-based* pricing mechanism, while the second is an assignment constraint that ensures that a consumer is at most allocated one quality level for a given service. The structural impact of the auction constraint is that the contribution of an individual item to the knapsack is no longer static and independent of the other items. While knapsack problems have been applied to a wide variety of scenarios such as capital budgeting, cargo loading, and cutting stock, and a variety of efficient heuristics that perform well in practice are well known (see, e.g., [5, 15]), the two additional constraints of our problem restrict the straightforward application of these techniques.

Besides developing a novel market model for vertically integrated services, our computational infrastructure makes the following three major contributions. First, we modify the well-known greedy heuristics for the knapsack problem to account for the auction constraint for a nonvertically integrated service setting (where consumers are restricted to bidding in only one of the many quality-differentiated service classes). Second, we demonstrate that under different marginal valuation conditions, different knapsack heuristics perform better. Finally, we observe in the case of vertically integrated services (where consumers are allowed to bid in multiple quality–differentiated service classes, but have an interest in consuming no more than a single class) that a direct application of the aforementioned heuristics results in a nonfair allocation (e.g., an allocation where a bidder may be served in a less preferred class even when he or

she has a bid higher than a successful bidder in his or her preferred class). Moreover, ensuring that the assignment constraint is binding (i.e., consumer gets service in at most one quality) is computationally expensive. To overcome these challenges we construct a novel data structure that allows us to adapt the techniques we develop for the simpler nonvertically integrated case.

Background

THERE ARE THREE PRIMARY TYPES of economic resource allocation mechanisms—capacity allocation, posted price, and auctions and negotiations. *Capacity allocation mechanisms* usually are the most efficient mechanisms if the type of individual customers, and thus their needs, can be identified by the controlling entity. In general, *posted price mechanisms* can be considered as the mathematical dual of capacity allocation mechanisms [8]. Under this mechanism, the general distribution of customer types is known, but individual customer type is not identifiable. In other words, the aggregate demand curve is known. Even though posted price mechanisms can dynamically compute the prices based on changing demand (see, e.g., [9]), these mechanisms are more effective when a relatively long-term demand trend is available.

We use mechanisms based on *auctions and negotiations* because we are considering the allocation of resources for dynamic and unique one-time products and services and their associated demands. The demand for such products may not be assessable in advance and, thus, computing posted prices may be extremely difficult. While the theoretical properties of such mechanisms have been discussed extensively in Bapna et al. [1] as well as in Bikhchandani and Mamer [4], we develop tools and techniques to implement such mechanisms in real time.

Our work follows Pinker et al.'s [18] call for leveraging the computational power of online auctions to auction complex goods and services that would otherwise be sold inefficiently using a posted price mechanism. Our context of dynamic and unique one-time products and services fits us in the upper right quadrant of Pinker et al. [18, p. 1460, figure 1], namely, high coefficient of variation of consumers' valuations and rarity of goods.

Typically, ignorance of what price to post is a reason for negotiating or holding an auction. Rothkopf and Harstad [19] provide a behavioral reason for holding auctions by asserting that one of the critical reasons for the use of bidding is that the formality of the auction process provides legitimacy in a way that the other economic means cannot. Wang [22] compares auctions with posted prices in a simplified setting under the assumptions of the independent private values model. Her central result is that auctions are preferable if the marginal revenue curve is steep, or more precisely if the demand is inelastic. The nature of the Internet-based services under consideration suggests that individuals' valuations of webcast events are likely to be highly dispersed. For instance, a die-hard fan of a particular rock star or a sport such as cricket would have a different valuation for a particular webcast than a moderately interested individual, whose valuation in turn would be entirely different from a person who is nonchalant about the events under consideration. This distribution of valuations might be expected

to change drastically across time and events. For instance, the monetary valuation of a moderate fan—who does not want to interrupt a day at the office, preferring a textual ticker tape relaying running commentary—would not be equal to that of the die-hard fan for whom nothing but live audio and video would suffice. Wang [22] confirms the intuition that the more dispersed the value of an object or a service to the potential buyers, the more auctions are preferred.

Providers currently offer this kind of flexibility in service definition. The question is can we structure efficient customer-driven mechanisms that integrate the issues of pricing, quality, and service mix? The paper also contributes to the scant literature on auctions with variable supply. By this we mean that the total number of goods being auctioned is not known ex ante, but is determined as a part of the auction. Part of the reason for lack of research in this area perhaps is due to Lengwiler's [14] negative result regarding lack of incentive compatibility in auctions of variable supply. Bapna et al. [1] explore the theoretical properties of a variety of market formulations in the context of pricing and allocation of unique, one-time digital products in the form of data streams. These range from allocatively efficient generalized Vickrey auction (GVA) to a multiple Vickrey auction (MVA). While MVA is not incentive compatible, they show that it achieves bounded posterior regret [17] and can be solved in real time. Jones et al. [12] used simulation-based approaches to explore allocation in combinatorial settings. The interested reader is referred to deVries and Vohra [6] for a review of the combinatorial auction literature. We develop solution techniques that can be used to implement such a mechanism.

Model Formulation

Assumptions and Scope

For completeness, we first present a general model of allocating capacity under an uncertain, widely dispersed, and dynamic demand structure where the content provider's objective is to maximize its revenue. This model, derived from Bapna et al. [1], focuses on the theoretical properties of a variety of market formulations in the context of pricing and allocation of unique, one-time digital products in the form of data streams.

We assume that customers have values for services that are unknown to the provider. Further, we assume that the provider will use some price-setting mechanism and some customers may be excluded from receiving the service based on the prices (too low) they are willing to pay. We first examine the general model in detail and subsequently discuss specific price-setting mechanisms and their properties.

Let there be $i = 1, \ldots, I$ consumers in the market for $j = 1, \ldots, J$ different services offered by a provider. Let V_{ij} denote customer i's valuation for service j. Let C represent the total bandwidth, or capacity, and let C_j be the capacity consumed by service j. Finally, let x_{ij} represent the decision variables where $x_{ij} = 1$ if customer i receives service j, and $x_{ij} = 0$ otherwise.

The General Model

We assume that, given a price-setting mechanism and the negligible marginal cost of offering the type of digital services under consideration, a provider's objective is to maximize revenue. Let p_{ij} be the price charged to customer i for a particular service j, where p_{ij} is unknown and determined by the price-setting mechanism. The capacity allocation, revenue-maximization problem is formulated as

Maximize

$$z = \sum_i \sum_j x_{ij} p_{ij} \tag{1}$$

subject to

$$p_{ij} x_{ij} \le V_{ij} \qquad \forall i = 1,...,I,\ \forall j = 1,...,J \tag{2}$$

$$\sum_j x_{ij} \le 1 \qquad \forall i = 1,...,I \tag{3}$$

$$\sum_i \sum_j x_{ij} C_j \le C \tag{4}$$

$$x_{ij} = \{0,1\}. \tag{5}$$

Model Characteristics

Note that this is a special case of the general 0–1 multiple products knapsack problem with an additional constraint (2) that represents the participation constraint. Equation (2) ensures that if $x_{ij} = 1$, the price p_{ij} charged to a customer i for service j is less than or equal to his or her value for that service V_{ij}. Equation (3) ensures that the bidders get an allocation in at most one service class. Equation (4) is a typical Knapsack capacity constraint.

A *unique feature* of this model, which prevents the application of conventional optimization procedures, is that it requires a price-setting mechanism. Atypically, there are two unknown quantities x_{ij} and p_{ij} in the revenue-maximizing objective function represented by Equation (1). Furthermore, the customer valuations V_{ij} are private information. Therefore, to satisfy the participation constraint, the provider has to create a mechanism that reveals this information.

Given the one-time nature of the services and the associated unknown and widely dispersed demand for such services, a posted pricing mechanism may not be optimal [22]. Furthermore, because there is a need to reveal the customer's private valuations V_{ij}, an auction mechanism seems to be an appropriate choice for price setting. In an auction, a customer i will bid[1] $B_{ij} \le V_{ij}$, where B_{ij} is customer i's declared bid for service j.

Auction Mechanisms

In our search for an optimal auction mechanism, we restrict our attention to a special class of such mechanisms—*direct revelation mechanisms.* In direct revelation mechanisms, bidders are asked to announce their valuations directly and the seller commits him- or herself to using rules for allocating the object and charging the buyers. The direct revelation mechanisms ensure both that the buyers will be willing to participate and that each will find it in his or her interest to announce his or her true valuation. Nobel laureate William Vickrey [21] proposed one such mechanism. In his seminal work, Vickrey noted that when a second-price auction is used—that is, the high bidder wins but pays only the price of the second-highest bidder—each bidder has a dominant strategy of bidding his or her true valuation, $B_{ij} = V_{ij}$.

In the following section, we formulate the resource allocation and pricing problem based on the principles of second-price auctions [21]. The multi-unit version of the second-price auctions that we propose has the desirable property of setting a uniform price for a given service class and as such is perceived to be fair for a given service class encouraging truth-telling behavior.

Second-Price Auction as a Price-Setting Mechanism

Here we consider a multi-unit analog of the second-price sealed-bid Vickrey auction for single items (MVA) [21]. The Vickrey auction adopts a uniform pricing scheme in which each accepted customer is charged a price equal to the value of the highest rejected customer for a particular service j. Let V_j represent an ordered list of values for service j such that $V_{1j} \geq V_{2j} \ldots \geq V_{Ij}$. Further, let B_j represent an ordered list of bids for service j such that $B_{1j} \geq B_{2j} \ldots \geq B_{Ij}$.

Under the MVA pricing mechanism, the general model (1–4) can be formulated as

Maximize

$$z = \sum_i \sum_j x_{ij} p_j \tag{1a}$$

subject to

$$p_j = \min_i \left[x_{ij} B_{i+1,j} \,\middle|\, x_{ij} > 0 \right] \tag{2a}$$

$$\sum_j x_{ij} \leq 1 \qquad \forall i = 1,\ldots,I \tag{3}$$

$$\sum_i \sum_j x_{ij} C_j \leq C \tag{4}$$

$$x_{ij} = \{0,1\}. \tag{5}$$

The objective function is the same as in the general case except that we drop the subscript i from p_{ij} because MVA is a uniform pricing mechanism. The participation constraint (2a) ensures that the price accepted for a particular service class is equal to the largest rejected bid for that class—hence the term $B_{i+1,j}$. This equality introduces dependencies between the bids of different individuals in the form of positive (one-sided) externalities. Each new lower value customer who is accepted into a particular service class lowers the price for all previously accepted customers of that class.

Nonvertically Integrated Services Solutions

IN THIS SECTION, WE EXAMINE THE BASE CASE and assume that a consumer bids in only one service class or, alternatively, if a consumer bids in multiple classes, he or she is willing to buy in all the classes simultaneously. As mentioned earlier, such a situation exists where different service classes reflect different products or customers are explicitly asked to bid in only one class. In a webcast application, this would be the case when a consumer chooses one of the classes (with a specified bit rate, for example) and bids only at that level. Mathematically, assuming that consumers bid in only one class implies that the assignment constraint (3) of the MVA model is ignored.

The analysis in this section will serve as a stepping-stone to the analysis of the next section, in which the consumer can simultaneously bid at different service levels and the system appropriately allocates the service to the consumer in only one of the levels. In a webcast application, this means that the consumer can specify a portfolio of bids for different bit rates and options of text and audio, but the system will guarantee that the consumer only gets one of the options in the fairest possible way.

Enumeration Procedure with Minimal Cardinality Bundle

In the case of a single-service environment, a simple *scan* procedure along with minimal cardinality bundling as described in Bapna et al. [1][2] can be used to construct an algorithm to optimally solve all cases. Table 2 illustrates an example of such an approach.

Intuitively, because the bids in a given class are sorted in nonincreasing order, accepting an additional consumer's bid may reduce the price for everyone under uniform-pricing approach. This may result in reducing the revenue with an additional customer because the additional gains from the customer may be lower than the loss due to lower prices for other consumers whose bids were already in the acceptance list. The minimal cardinality bundle ensures that when such an instance occurs, the bid that reduces the overall revenue is bundled with other bids in the sorted list until there is positive revenue gain, if feasible. Then the bids in a bundle are either considered all together or none at all. For example, in Table 2, it is not in the provider's best interest to offer service to all customers between 1 and 8. The service provider's revenue will go down by \$1 from \$9 to \$8 if he or she accepts bidder 2 in addition to bidder 1. Alternatively, consider the \$15 revenue the service provider gets from accepting the 5 high bidders. Accepting bidder 6 will result in revenue dropping to \$12, and accepting

Table 2. Example of Scan Procedure with Minimum Cardinality Bundling

Customer	1	2	3	4	5	6	7	8	9	10
Bids ($)	11	9	4	4	3.5	3	2	2	2	1
Revenue	9	8	12	14	15	12	14	16	9	
Marginal revenue	9	**–1**	**3**	2	1	**–3**	**–1**	**1**	–7	
Bundle MR	9	**2**		2	1	**0.33**				

Notes: Observe in the marginal revenue row that accepting the top two bids instead of the top bid alone causes a $1 drop in revenue, and hence the motivation for bundling bids 2 and 3 together to ensure no drop in revenue as we accept an increasing number of bids.

bidder 7 will yield total revenue of only $14. It is only after the service provider accepts bidder 8 can he or she improve upon accepting just the top 5. In other words, customer 2 will receive service if and only if customer 3 receives the service and customer 7 will receive the service if and only if customer 8 receives the service. As shown by Bapna et al. [1], a minimal cardinality bundle (i.e., where the customers are bundled in such a way that together their marginal revenue contribution is positive) allows the use of a simple *scan* procedure to decide revenue-maximizing allocation.

However, when capacity has to be allocated among several different services, the *knapsack* structure of the problem becomes apparent, and so does the issue of fairness[3] *between* service classes. It is trivial to see that the fairness is guaranteed within a class under the assumption that the bidder bids in at most one service class and Vickrey prices. The knapsack structure of the multiple-service case indicates that a decision maker has to decide which services to offer and how many customers receive each of the services at what price. Before we discuss the solution techniques, it is useful to review the preprocessing steps—that is, collecting, sorting, and bundling of the bids in each service class. The idea behind applying these preprocessing steps is that if it is not optimal to consider a bid in a certain service class (by virtue of its causing a decrease in revenue) when that class is, hypothetically, the only service class, then that bid will not be considered when multiple service classes are being offered. Because the computational costs of solving the knapsack problem are dependent on the number of bids, preprocessing helps reduce such costs. Thus, an essential step is to apply the simple *scan* procedure along with minimal cardinality bundling of Bapna et al. [1] to each individual service class to obtain a revised, truncated row bid matrix. The combined effect of bundling and truncation gives us a ranked $\tilde{i} \times j$ matrix that contains "bundles" derived from the individual bids, where $\tilde{i}$ represents the truncated set of bids containing only those bids that can potentially have a positive marginal impact on total revenue.

It is well known that the knapsack problem is NP (nondeterministic polynomial-time)-hard. However, in our framework, it is reasonable to expect that the number of consumers bidding will far outnumber the number of different service classes that a provider offers—that is, $I >> J$. Combining this with the special structure of the bid (knapsack elements) values in each service class allows us to construct some fast solution techniques, including a polynomial optimal solution technique.

Using the special value dependence structure due to MVA, we can construct an enumerative polynomial mechanism for optimally computing the allocation and prices. However, the computational costs of such a technique may be unacceptable for real-time applications in electronic markets. Therefore, we further explore heuristics in order to provide a decision maker with a portfolio of fast solution techniques that trade off accuracy and computational time.

Heuristic Solution Techniques

We begin by considering the most popular approach to knapsack problems—namely, the *greedy algorithm*—which is based on choosing the elements in nonincreasing value-to-weight (bids-to-capacity) ratio [15]. The heuristic is intuitively appealing

since it essentially prioritizes elements such that the elements that provide the highest value at the lowest cost are included first. Technically, the heuristic exploits the fact that the solution $\bar{x}$ of the continuous relaxation of the problem has only one fractional variable. To obtain a feasible solution, we just set this fractional variable to 0. It is well known that in the worst case, this procedure can be arbitrarily bad. For instance, suppose we have two objects, *a* (say, a text-based baseball score subscription) with a value and weight of 1 and *b* (say, a video-streaming baseball service) with a value and weight of 100 and a knapsack capacity of 100. Since the V/C ratios are both 1.0, the greedy algorithm might arbitrarily choose object *a* resulting in a greedy solution with a knapsack value of z' (the knapsack value by applying a heuristic) that is 100 times worse than the optimal. We can reduce this worst-case performance ratio to 1/2 if we choose the greater of z' and $z(\bar{x} = \{b\})$, the value considering the critical object alone. This worst-case scenario would arise if both objects were equally valued.

Under our formulation, we have to operationalize the value-to-weight ratio of bids that are not independent of each other, unlike in traditional knapsack settings. For example, with MVA, each accepted lower bid lowers the price for all preceding higher bids, but could aggregate to higher revenue. Because the impact of each bid is not solely dependent on the bid itself, we develop an alternative version of the traditional knapsack value-to-weight (V/C) ratio–based greedy heuristic. The alternate heuristic considers the *marginal revenue* (MR) contribution of each bundle, and uses the MR-to-weight (MR/C) ratio, in contrast to the traditional value-to-weight ratio.

An interesting property of the well-documented V/C ratio–based greedy heuristic is that it guarantees fairness between classes. By design, the heuristic picks up the highest V/C value among all of the candidate classes, and cannot allocate to a lower-valued bidder in a class, while not allocating to a higher one. In contrast, the MR/C procedure that we develop does not guarantee a fair allocation because the marginal value depends on the difference in price that an additional bid imposes and the number of bids that were higher than the current bid under consideration. In other words, it is possible that for a given set of bids in two different classes, we may have a situation where $(V_1/C_1) > (V_2/C_2)$ but $(MR_1/C_1) < (MR_2/C_2)$. In such a situation, it is guaranteed that revenue generated by using the MR/C technique will be equal to or higher than the revenue being generated by the V/C technique. However, it is easy to see that MR/C may not be fair; for example, in the aforementioned case, if the capacity is exhausted after including the second bid (using MR/C), then the first bid does not get in even though the bid itself specified higher willingness to pay for every unit of capacity consumed. We explain the nuances of using MR/C (see [15] for V/C), and then perform a comparative analysis of these two versions of the greedy heuristic in the next subsection. Our objective is to see under what conditions of demand distributions should a seller adopt one of these two versions.

The Iterative Greedy Approach

In this subsection, we focus on providing details of the MR-greedy and a forward-moving approach that improves the performance of MR-greedy. We also outline the necessary preprocessing steps that are necessary to avoid incorporating the bids that

provide nonpositive marginal revenues in the solution. As we will see below, both the preprocessing steps and the forward-moving approach to improve the heuristic performance are also applicable to the more traditional V/C-greedy procedure that we adapt to the uniform pricing environment.

Let us refer back to Table 2. A typical *greedy* procedure in a typical knapsack setting would ignore customer 2's bid and move on to customer 3 if marginal revenue was the value metric. However, the MVA structure imposes an additional ordering constraint that forbids us from doing that. To smooth the effect of nonmonotonic MR values of individual bids, Bapna et al. [1] use bundles of individual bids, thereby providing a positive MR contribution whenever possible. The value of creating bundles is amplified in the multiple service cases while applying greedy or greedy-like procedures to solve the knapsack problem.

It is instructive to pause and differentiate between a general knapsack greedy algorithm and the implementation of greedy under our formulation. Let a *current* element be defined as the smallest indexed element in a service class that has not been included in the knapsack. Then the greedy procedure in our formulation selects the largest marginal revenue from the *current* elements in *all* the service classes. This is, again, due to the dependency of bids in a service class and the resulting restriction that a higher indexed element in a service class cannot be included unless all of the lower indexed elements have been included. Therefore, as opposed to the greedy implementation in a general knapsack problem where all the elements are considered in nonincreasing value-to-weight ratio, in our framework, the highest available MR-to-capacity ratios in all of the service classes are considered in each step. In other words, in each step, there are a maximum of J elements from which the highest MR-to-capacity is included in the knapsack.

Let $\bar{x}_g$ denote the solution obtained from *procedure greedy.* The question arises, "can we improve on *procedure greedy?*" While the worst-case scenario is well known for a general knapsack problem, it is unlikely to occur in our scenario where, by design, each service requirement is much smaller than the knapsack size. In addition, since the value of items already in the knapsack is changing as more service requests are added in a nonlinear manner, the *tough cases* where a greedy procedure will not perform well have different characteristics. In the next section, we illustrate these worst-cases both analytically and numerically. Based on the characteristics of these worst-cases, we provide an improvement of marginal revenue–based greedy (MR-greedy) technique. A pseudocode for MR-greedy is provided in Figure 1.

Worst-Case Scenarios

Without loss of generality, suppose there are two service classes, each with an individual capacity requirement of 1. The *tough cases* will occur if the bid pattern presented in Panel A of Table 3 is observed for any $\delta > 2\varepsilon$, where δ, ε are infinitesimal fractions and K is a large integer. The associated MRs that the procedure greedy would utilize are presented in Table 3, Panel B.

In this case, if we have N units of capacity, then the optimal solution is $NK/2$ whereas the *greedy* solution is $K + (N - 1)\delta$, which tends toward K as $\delta \to 0$. Thus the MR-

```
procedure greedy ( capacity, ĩ )
begin
    x̄ = ∅      // initialize vector representing solution of accepted bids to NULL
    ĩ = ∅      // initialize vector representing number of bidders accepted in each
               class to NULL
    do
        begin
            find the bid bundle that has the highest value to weight ratio
            x̄ = update_solution( current_bundle )
            update_ĩ
            update_capacity_consumed
        end
    while capacity_available
end
```

Figure 1. Pseudocode for Procedure *Greedy*

greedy bound is 2/*N*. Intuitively, the worst case occurs when the sorted bids have a structure where marginal revenue is small for a few bids at the beginning of the list in a given class (as compared to other classes) followed by a large number of bids with high marginal revenue. Such a case may happen when, for example, there is a sudden drop in a bid for a given class and then all the other bids are the same (as represented in Table 3, Panel A). The drop in the bid creates small marginal revenue initially since the price for everyone who is currently winning drops due to the uniform Vickrey pricing rule. Note that the marginal revenue may be zero in the worst case; that is, no other bids may be considered in that particular class as long as there are bids with positive marginal revenues in other classes. However, if the rest of the bids in the class are the same as the "offending" bid, the subsequent marginal revenues will be equal to those bids themselves, which may be quite high (because price will no longer be changing). The structure of the worst case provides an intuitive solution to improve on the basic greedy heuristic that is described in the next subsection.

Improving the Basic Greedy Solution

What if we were to utilize a procedure where after running the greedy, we modify the greedy solution by including one more element in each class than the *greedy solution* and then run *greedy* with the remaining capacity? Let us call this procedure *greedy_plus_1*. Note that this addresses the problem of getting stuck in a given class by "peeking" ahead. Intuitively, a *greedy_plus_n* procedure can be derived from the worst-case behavior of greedy approaches to the knapsack problem described above. In such cases, *n* is the number of items that are included beyond the *best* solution

Table 3. Worst-Case Scenario Analysis

Panel A: Tough case bids

	Bids					
Class	1	2	3	4	5	
1	K	K	$K/2 + \varepsilon$	$K/2$	$K/2$	…
2	δ	δ	δ	δ	δ	…

Panel B: MRs for tough case

	Marginal revenues				
Class	1	2	3	4	
1	K	**2ε**	$K/2 - 2\varepsilon$	$K/2$	…
2	δ	δ	δ	δ	…

Notes: The bidding pattern observed in Panel A, will lead to the set of marginal revenues depicted in Table 3b. Because $\delta > 2\varepsilon$, the boldface figure in Panel B represents a hump that acts as a barrier for simple greedy procedure. This suggests that in the best case the greedy procedure can only lead to a revenue that is some tiny fraction greater than K. On the other hand a *greedy_plus_1* algorithm will overcome the "hump" and lead to a revenue of $3/2K$ if the capacity was 3 or more generally $NK/2$ if the capacity is N.

obtained so far. The last item(s) may also have been arbitrarily left out in preference for an equivalent *MR/weight* ratio bid having a higher value. It is straightforward to show that the worst-case bound of this procedure improves to $3/N$.[4] Note that, although for the greedy approach, the worst-case bound is still infinitely bad, for the *greedy_plus_n* approach, our tough cases are the worst cases and, hence, represent worst-case bounds. Figure 2 provides the pseudo-code for the *greedy_plus_n* approach.

The *greedy_plus_n* procedure essentially iterates through the various service classes, fixing the number of bids accepted for class j, and then reapplying *greedy* with the reduced capacity to the remaining classes ($\tilde{j} - j$), until there is no improvement. Proposition 1 presents its worst-case time complexity:

> *Proposition 1: In the worst case, greedy_plus_n takes $O(I^3)$ time, where I is the total number of consumers who place a bid.*

Proof: First, note that for a given value of n the worst case for *greedy_plus_n*, from the perspective of complexity, occurs when the solution is to include all bidders; however, each application of *greedy_plus_n* improves the solution by including exactly one more element. In other words, each iteration adds at least two more bidders to the best solution: one by design[5] and one to improve the existing solution. Therefore, a maximum of $I/2$ iterations of *greedy* are required—that is, for a given n, *greedy_plus_n* takes $O(I^2)$ time since *greedy* takes $O(I)$ time.

```
procedure greedy_plus_n(capacity)
begin
    x̄ = x̄_g // Begin with the greedy solution
    x̄_temp = ∅
    do
        for ctr = 1 to j̃ // iterate through all service classes
        begin
            x̄_temp = x̄_{i+1,ctr} // force the next bundle in the current class to the
                solution
            capacity = compute_remaining_capacity(x̄ , ctr) //update capacity
            x̄_temp = greedy(capacity, j̃ - ctr) //solve greedy with remaining
                capacity and remaining classes
            if z(x̄_temp) > z(x̄)  //if new solution better than existing solution,
                update solution
            begin
                improvement=1
                x̄ = x̄_temp
            end
        end
    while improvement  //iterate until there is improvement
```

Figure 2. Pseudocode for Procedure *Greedy_plus_n*

Next, the maximum number of iterations due to incrementing n are of the order of I. Therefore, in the worst case, the overall algorithm is guaranteed to stop in $O(I^3)$ time. Q.E.D.

To illustrate the solution using *greedy_plus_n*, consider the following example:

Numerical Example: The provider offers three different kinds of service classes—A, B, and C. Assume, for expositional simplicity, all classes to have an equal weight of unity. The total available capacity is nine units. Table 4 displays the complete solution process, including starting (sorted) bids, MRs, ranked and bundled $\tilde{i} \times j$ matrix with MRs, *greedy* solution, and the solutions achieved by the application on *greedy_plus_1*. In Table 4, the notation a^b is used to denote a bundle with value-to-weight ratio a, and cardinality b. The boldface figures in the table indicate the consumers chosen by *greedy* and the italicized boldface figures indicate the elements that are fixed during the application of *greedy_plus_1*.

To start with, 4 bundles are created: 1 for service A, 2 for service B, and 1 for service C. The application of *greedy* results in choosing 1 customer for service A, 3 for service

Table 4. The Solution Process

Problem data: capacity = 9; number of services = 3; capacity consumed = 1 by each element in all services									
Sorted bids									
A	12.0	9.0	4.5	3.0	2.5	2.5	2.5	2.5	1.0
B	6.0	6.0	4.0	3.0	2.0	1.9	1.9	1.5	1.5
C	4.0	3.0	2.0	1.5	1.1	1.1	1.1	1.1	1.1
Marginal revenues									
A	9.0	0	0	1.0	2.5	2.5	2.5	–9.5	
B	6.0	2.0	1.0	–1.0	0.5	1.9	–0.9	0.6	
C	3.0	1.0	0.5	–0.1	1.0	1.1	1.1	1.1	
Bundled marginal revenues									
A	9.0	0.333^3	2.5	2.5	2.5				
B	6.0	2.0	1.0	0.25^2	1.9	0.3^2			
C	3.0	1.0	0.5	0.5^2	1.1	1.1	1.1		
Greedy solution (boldface figures) = 23.5									
A	**9.0**	0.333^3	2.5	2.5	2.5				
B	**6.0**	**2.0**	**1.0**	0.25^2	1.9	0.3^2			
C	**3.0**	**1.0**	**0.5**	$\mathbf{0.5^2}$	1.1	1.1	1.1		

Greedy_plus_1 (on "A" diagonal, boldface figures indicate the fixed elements) = 26.5*

A	***9.0***	***0.333***^{3}	**2.5**	**2.5**	**2.5**		
B	**6.0**	2.0	1.0	0.25^{2}	1.9	0.3^{2}	
C	**3.0**	1.0	0.5	0.5^{2}	1.1	1.1	1.1

Greedy_plus_1 (on "B" diagonal, boldface figures indicate the fixed elements) = 24.4

A	**9.0**	0.333^{3}	2.5	2.5	2.5		
B	***6.0***	***2.0***	***1.0***	***0.25***^{2}	**1.9**	0.3^{2}	
C	**3.0**	**1.0**	0.5	0.5^{2}	1.1	1.1	1.1

Greedy_plus_1 (on "C" diagonal, boldface figures indicate the fixed elements) = 23.6

A	**9.0**	0.333^{3}	2.5	2.5	2.5		
B	**6.0**	**2.0**	1.0	0.25^{2}	1.9	0.3^{2}	
C	***3.0***	***1.0***	***0.5***	***0.5***^{2}	***1.1***	1.1	1.1

Notes: Bundle sizes are denoted as superscripts. * Optimal solution.

B, and 5 for service C with the picking order of A1-B1-C1-B2-B3-C2-C3-C4-C5. This yields the total revenue of 23.5. In our implementation, heuristic *greedy_plus_n* is applied to all of the service classes based on the solution obtained in the previous step. Therefore, in this case, *greedy_plus_1* is applied to each service class one-by-one. First, we take the number of customers chosen in service class A by the *greedy,* add one additional customer, and reduce the available capacity by the space taken up by these elements. Then, *greedy* is applied to rest of the customers with the adjusted capacity for the knapsack. The same process is repeated for service classes B and C. In this case, customers 1 and 2 in service class A are chosen first. Since customer 2 is a bundle of 3 customers, the total available capacity is reduced to 5 units. When we apply *greedy* for rest of the customers, the solution is to choose 7 customers for service class A and 1 each from service classes B and C with the *greedy* picking order of B1-C1-A5-A6-A7. The resulting revenue is 26.5—that is, higher than *greedy.* Next we apply *greedy_plus_1* to service class B, again starting from the *greedy* solution. The revenue from this is also greater than *greedy.* Finally we apply *greedy_plus_1* to class C, which also results in higher revenue than *greedy.* However, the best solution from overall application of *greedy_plus_1* is obtained when we apply it to class A. Therefore, at this stage, that solution is designated best solution so far for successive application of *greedy_plus_1* or higher order of *n*.

In our example, the solution obtained by application of *greedy_plus_1* to service class A is the optimal solution; however, in general, that may not be the case. Let the current solution obtained after the application of *greedy_plus_1* be denoted by $\bar{x}_c$. Because of the nonmonotonic nature of the marginal revenue curves for each of the service classes, we cannot be sure that it is sufficient to eliminate the optimality gap by including just one extra item into a service class. Horowitz and Sahni [11] describe a *forward move* consisting of inserting the largest possible set of new consecutive items into the current solution. When such a forward move is exhausted, the current solution obtained is compared with the best solution so far and a choice is made whether to make further forward moves or to backtrack. Using a similar approach, we further exploit the special structure of our problem, to create procedure *greedy_plus_n*, which is essentially the same as *greedy_plus_1*, with $\bar{x}_{temp} = \bar{x}_{i+k,ctr}$ in Figure 2.

In general, because optimality of a given solution cannot be verified, several stopping criteria can be used. Furthermore, several other observations can be used to minimize computational load. While discussing the implementation details is beyond the scope of this paper, we present some observations.

First, note that if a fixed customer set at any point is a subset of a previously obtained solution, then fixing those elements and applying *greedy* for the rest of the capacity will duplicate the previously obtained solution. Therefore, such steps can be skipped. For example, if we tried to do another iteration of *greedy_plus_1* on the solution obtained above, we will start with service class B (because all customers of service class A have already been chosen). Based on the current solution, we will fix the first 2 customers in class B and apply *greedy* for the rest of the customers with a capacity of 7. The resulting solution will exactly be the same as obtained by the application of

greedy to all the customers and capacity. This is no surprise because B1 and B2 were part of the original *greedy* solution and hence choosing them and applying *greedy* to the rest of the customers and capacity reproduces the result. Thus, keeping a record of best solutions chosen at different stages of computation can reduce the number of steps in *greedy_plus_n* substantially.

In terms of stopping criteria, we chose a minimalist approach where if the application of *greedy_plus_k+1* does not produce a better result, we stop. In other words, suppose the best solution so far was obtained by applying an iteration of *greedy_plus_1*. We then apply another iteration of *greedy_plus_1* with the new solution; if that does not improve the solution, we next apply *greedy_plus_2* where 2 additional customers are chosen in each class. If *greedy_plus_2* does not produce a better result, we stop; otherwise we continue.

Optimality Gap

In this subsection, we present some results for the performance of *greedy* and *greedy_plus_n* as compared to the enumerative polynomial optimal solution technique mentioned in the subsection "Enumeration Procedure with Minimal Cardinality Bundle." First, we present the performance of these heuristics as a function of problem size. The experiments were conducted with the following set of parameters. We tested with three service classes ($J = 3$). The number of initial bids in each class was kept the same, e.g., 20, 40, ..., in each class ($I = 20, 40, ...$). Capacity ratio C_1 (e.g., video): C_2 (e.g., audio): C_3 (e.g., text) was fixed to be 4:2:1, with total capacity kept at five times the initial number of bids in a class. For example, if $I = 20$, then $C = 100$. Bidders' valuations are drawn from a uniform distribution. Our usage of the uniform distribution for the bidder's valuation is consistent with the emerging experimental standard in combinatorial auction research, the Combinatorial Auctions Test Suite (CATS) database (see www.cs.ubc.ca/~kevinlb/CATS/). In addition, our settings are similar to other work that is designing futuristic markets. For instance, almost identical experimental settings for a knapsack Vickrey formulation, but in the context of sequential auctions of capacity, can be found in Ng et al. [16].

The reported results are the mean of five replications with each setting with the same input parameters for corresponding runs with different solution procedures. Figure 3 graphically illustrates the improvement by applying the *greedy_plus_n* as compared to the *greedy* solution. In most cases, *greedy_plus_n* improves the *greedy* solution substantially. Overall, the average optimality gap using the *greedy* was about 2 percent as compared to 0.5 percent with *greedy_plus_n*.

Therefore, on average, *greedy_plus_n* produced 400 percent better performance with only a 50 percent increase in computational time. The worst-case performance of *greedy* during the 250 experiments was a 9.6 percent optimality gap versus 1.86 percent for *greedy_plus_n.* Given the real-time computation requirements of electronic markets, the amount of computational time required is an important factor. Figure 4 compares the performance of the three approaches with respect to computational time as problem size increases.

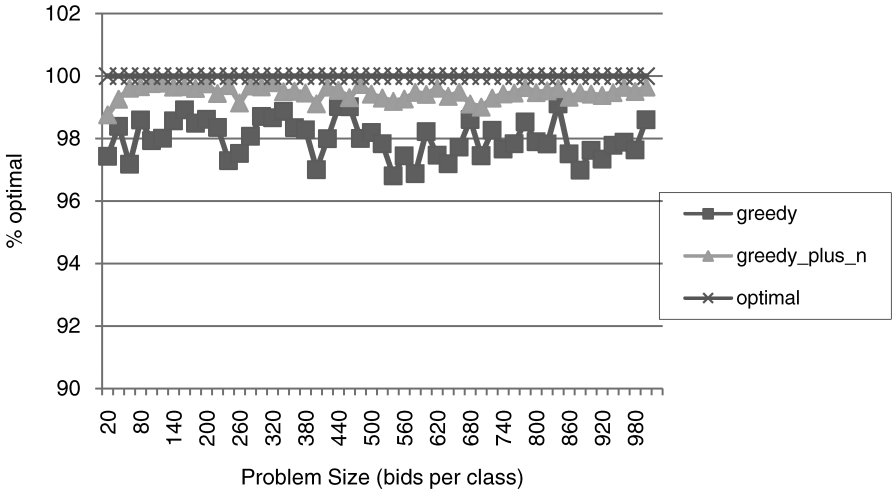

Figure 3. Greedy Versus *Greedy_plus_n*

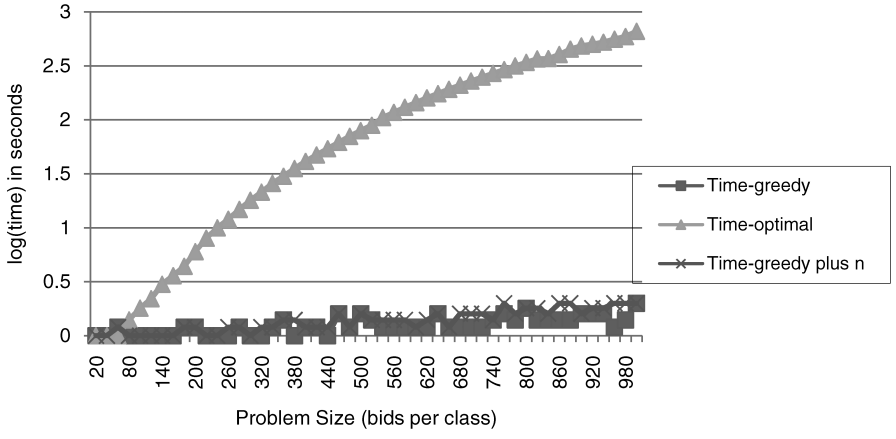

Figure 4. Comparison of Computational Times

For large problems, even the pseudo-polynomial optimal computation can take many minutes on a typical PC. For instance, a 3,000-bid problem took almost 11 minutes to solve optimally as opposed to 1.5 seconds using the *greedy_plus_n* heuristic.

MR/C- Versus V/C-Greedy Heuristic

Continuing our heuristic analysis as per Chellappa and Kumar [5] and Martello and Toth [15], we examine a second strain of greedy heuristics based on an adaptation of the more traditional V/C procedure. Before running this procedure in our environment, we need to apply the same bundling preprocessing step to avoid selecting bids that add

Table 5. Bundling with V/C Solution Process

Bundled bids							
A	12.0	5.5^{3}	2.5	2.5	2.5		
B	6.0	6.0	4.0	2.5^{2}	1.9	1.5^{2}	
C	4.0	3.0	2.0	1.3^{2}	1.1	1.1	1.1
Greedy solution (boldface figures) = 21.5							
A	**9.0**	**0.333**^{3}	2.5	2.5	2.5		
B	**6.0**	**2.0**	**1.0**	0.25^{2}	1.9	0.3^{2}	
C	**3.0**	**1.0**	0.5	0.5^{2}	1.1	1.1	1.1

Note: Bundle sizes are denoted as superscripts.

nonpositive marginal revenues to the knapsack, due to the uniform pricing feature. The procedure then selects the bids in order of their V/C ratio. To illustrate, we use the same example shown in Table 5. The top panel shows the bundled bids which serve as inputs to the V/C heuristic. The bottom panel shows the resulting selection of bids upon application of the heuristic. As with the MR strain, here we can also apply the forward-moving approach *greedy_plus_n* to improve the solutions.

We test the relative performance of the two versions of the greedy heuristics under varying distributions of the demand. In the context of online services such as text or audio or video content streaming, we would expect that newer or niche services would have higher variability in demand. We expect that as variability in the demand increases, the pure revenue-driven MR/C approach should outperform the V/C approach and the seller would be better off using an MR-based greedy heuristic. As the market matures and the variability reduces, the seller would be better off using the V/C-based greedy heuristic. Intuitively, we would expect higher variability to cause higher nonmonotonicity in the MR curve, and consequently the MR bundling would have a bigger impact in smoothening that out.

Figure 5 presents results from simulations where we test the impact of demand variability. The *y*-axis is the difference in revenue in percentage terms between MR/C and V/C algorithms keeping everything else constant. We simulate the demand variability in a single class by drawing bids from distributions with a fixed mean value but different variances. The relative valuations in different classes are chosen to depict three different conditions:

1. *Constant marginal value of capacity*—This is the case where the mean bid value to capacity ratio of each class is held constant. The performance in this case is depicted by the line labeled "constant" in Figure 5.
2. *Increasing marginal value of capacity*—Although rare, this may happen where higher-capacity services deliver significantly higher value for customers. This is simulated by assigning the lowest mean bid value-to-capacity ratio for the smallest class and the highest mean bid value-to-capacity ratio for the largest

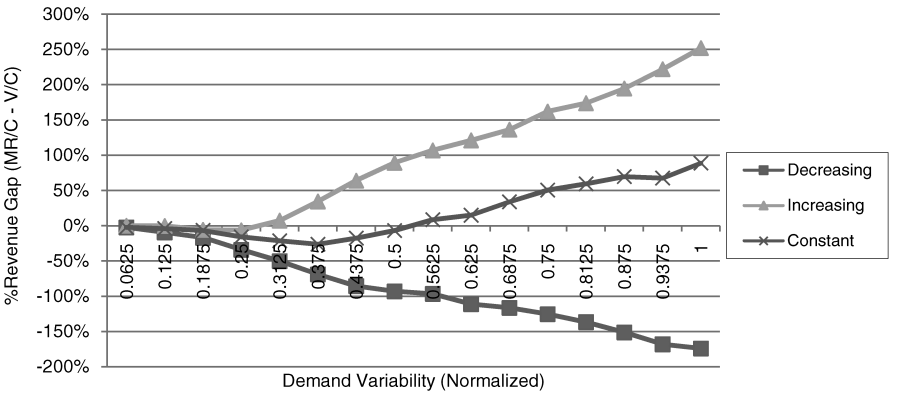

Figure 5. MR/C Versus V/C Greedy Under Varying Demand Variability and Marginal Value of Capacity

class. The performance in this case is depicted by the line labeled "increasing" in Figure 5.

3. *Decreasing marginal value of capacity*—This is the most common economic scenario. Here the customers have higher valuations for higher classes but not in proportion to the increased resource requirement. This is simulated by assigning the lowest mean bid value-to-capacity ratio for the largest class and the highest mean bid value-to-capacity ratio for the smallest class. The performance in this case is depicted by the line labeled "decreasing" in Figure 5.

The results indicate that under constant marginal valuations between the classes (constant V/C ratio), the MR-based heuristic is initially outperformed by the V/C heuristic until a crossover point is reached, beyond which the enhanced variability adversely affects the V/C-based greedy. Also, note that in case of diminishing marginal valuations between the service classes (case 3), V/C dominates MR.

Vertically Integrated Services Solutions

In this section, we relax the assumption that a consumer places a bid in only one service class. If the provider is selling multiple versions of the same service, such as text only, audio, audio and video at different bit rates, it is reasonable to assume that while a consumer may ultimately consume only one service, he or she may have, and indeed reveal, valuations for more than one service. We assume that the consumers' marginal valuations with respect to capacity requirement of a service are either monotonically nonincreasing or monotonically nondecreasing per unit of capacity. The first is a classical case where marginal value of a higher service class is decreasing, for example, with increasingly higher levels of audio quality. The latter is the case where higher service classes constitute the addition of services in two or more classes providing positive externalities. This situation results in total valuation for a higher level of service being higher than the addition of individual lower-level services. We

do not consider the case where consumer marginal valuations are nonmonotonic in nature. Such multimodal distribution of the valuations would be rare and insignificant occurrences. Furthermore, if these are unimodal (for example, first increasing and then decreasing), then they could be modeled and solved as convex combination of the above two cases. Recall that this relaxation enforces the assignment constraint (3) of our MVA formulation.

The presence of the assignment constraint increases the complexity of the problem significantly because a direct application of the heuristics is not feasible. The complexity arises when a consumer who has already temporarily received allocation in a given service class becomes eligible to be considered in a higher class. In such a case, the prior allocation needs to be recomputed, removing the concerned bidder's bid from the lower class. Formally, considering a consumer for service in class $\hat{j} \neq j$ when the customer is currently winning in class j involves backtracking in class j to remove that bidder from consideration in that class. That in turn may change the current level of the marginal (price-setting) bids in class j, with the possibility that resulting allocation may be infeasible, revenue reducing, or both. Worse, the forward move involving class $\hat{j}$ may prove to be nonrewarding at the current or subsequent stage and the bid in class may indeed be back in play. From the point of view of the solution techniques, depending on the valuation structure of the bidders, this may or may not involve rebundling in the affected classes.

The Case of Monotonically Nondecreasing Per-Unit Quality Valuations

Of the two valuations structures under consideration, this represents the case when consumers have positive marginal valuations of the multiple service classes. This is an indirect manifestation of positive externalities in such markets. Recall that despite bids in multiple service classes made by the consumer, he or she is interested in consuming a single service. For convenience, let us define the "margins of capacity" as that capacity in the allocation process where some classes of services may not be available due to lack of available capacity, however, allocation in other classes may be feasible due to smaller capacity requirement.

> *Proposition 2: It is Pareto efficient to consider only the bid in the highest service class made by a consumer bidding in multiple classes, given monotonically nondecreasing marginal valuations, except at the margins of capacity, when the next-lower-class feasible bid should be considered.*

Proof: Such a strategy is optimal from the seller's point of view since the bid in the highest service class also represents the highest per unit capacity bid made by the bidder under consideration. It is also optimal from the bidder's perspective since this bid maximizes the consumer's surplus. The only exception to this rule is the case when at the margins of capacity, if a higher-class (higher-capacity) bid is infeasible, then the next-lower-class feasible (nonwinning in any other class and within capacity) bid should be considered. Such a bid would only be accepted if it had a positive marginal

revenue contribution in that next lower class. The above argument extends until all successively lower-class bids are exhausted or infeasible. Q.E.D.

Note that the policy of considering a bidder's highest-class bid, except at the margins of capacity, has the added benefit that our entire solution approach developed for the nonvertically integrated services directly carries over. In particular, there is no need for rebundling since a non-highest-class bid will only be considered at the margins of capacity.

The Case of Monotonically Nonincreasing Per-Unit Valuations

This scenario arises when the provider's extra effort in offering a higher-quality service is not proportionally matched in valuations by the consumer. For instance, in order to jump from audio only to video may require more than doubling the capacity requirement; however, customers may have valuations of less than double the amount for audio. Mathematically, this case can be represented by the situation when $\forall$ classes j, j', where $c_j \geq c_{j'} \Rightarrow B_j \geq B_{j'}$, but $(B_j/c_j) \leq (B_{j'}/c_{j'})$. Such a bid structure creates a significantly trickier problem. In this case, assignment constraint (3) of our formulation becomes binding, and we have to consider *all the bids* made by a consumer but *allocate at most one. In addition, from a mechanism design perspective, we want to ensure a fair and efficient allocation.* By efficient we mean that the buyers get the highest service level that their preferences allow them. By fair we mean that no one with a less revealed valuation can get a service if a higher revealed valuation person has been denied.

An Efficient Optimal Procedure with Fair Allocation

Fairness is a very significant issue when bidders have a choice of bidding in multiple service classes with Vickrey pricing in each service class. If fairness is not enforced under revenue maximization, a bidder bidding in a single class M could get preference over a consumer who bid higher for class M but has bid in multiple classes with a higher bid/capacity ratio in a lower class. This would constitute an allocation that will give rise to negative consumer sentiments along the dimensions of envy and unfairness. Since such an allocation will weaken any incentives to bid in multiple classes and will make the mechanism's performance unpredictable, we create an envy-free mechanism. When the V/C rankings are used to pick the allocation ordering, we guarantee that bidders will get the highest service level that their preferences allow them. Such an allocation is fair in the sense that no one with a lower revealed valuation can get a service if a higher revealed valuation person has not gotten it.

This fairness implementation comes at a cost. Table 6 presents an example where a revenue-maximizing allocation is such that John's allocation is not envy-free. Even though he had higher valuation than Mary in the high class, he ended up getting the low class in the revenue-maximizing allocation. However, the envy-free allocation presented in the table results in lower revenue that can be considered as the *cost of fairness.*

Table 6. Fair Allocation and Pricing Is Significantly Different from Revenue-Maximizing Allocation

Class	2	1
Capacity	1	2
Bidder	Bids	
John	6	9
Mary	5	6
X	3	
Y		4.5

	Fair solution		Revenue-maximizing solution	
Class	2	1	2	1
John		9	6	
Mary		6		6
Price		4.5	5	4.5
Quantity		2	1	1
Revenue		$9.00		$9.50
Cost of fairness	$0.50			

Note: The table shows that under the fair allocation, John and Mary win Class 1 and pay $4.5 each, whereas under the revenue-maximizing allocation, John wins class 2 and pays $5, whereas Mary wins class 1 and pays $4.5.

We develop a fair solution procedure that is a simple extension of the optimal enumerative procedure as discussed in the "Enumeration Procedure with Minimal Cardinality Bundle" subsection, with added bookkeeping. We exhaustively consider every combination of each bidder in each class as the price-setting bidder, and enforce the fairness constraint by separately calculating the revenue of a given allocation using a downward sweep from the highest service level to the lowest. A bidder once allocated in the higher class is subsequently removed from consideration in the lower classes. This does not increase the computational complexity of the procedure.

In general, to measure the cost of fairness of this enumerative procedure, we have to compare its results with an optimal revenue-maximizing allocation. For this purpose, we use a brute force enumerative approach, which obviously suffers from combinatorial explosion as the number of bidders grows. This brute force revenue-maximization procedure in itself has no value to the seller. It violates fairness, and is only valuable to measure the cost of fairness. It does, however, limit the range of computational experiments we can conduct. A brute force enumeration to calculate the revenue-maximizing allocation with 14 bidders in 3 classes requires 4^{14} iterations (bidders could win in any one class or not at all), assuming each bidder bids in each class. Our analysis, conducted using 111 randomly generated auctions, showed that

the revenue-maximizing allocation yielded, on average, 4 percent (standard deviation of 16 percent) more revenue than the optimal fair allocation.

A Greedy Heuristic for Fair Allocations: Creating Pseudoclasses

Aiming at solving the allocation problem in real time, we now consider the development of greedy heuristics based on the V/C ratio that also guarantee fair allocations. We exploit the special structure of the problem to address the issue of allocation in different classes by a greedy procedure, even in the presence of the assignment constraint. The key property that allows us to efficiently solve this problem lies in the realization that the underlying commodity that is being allocated is the same for all classes—that is, the capacity—and it can be arbitrarily divided across classes.

We create a novel decomposition that involves generating pseudoclasses for all of the higher classes beyond the base class for any vertically integrated classes. The capacity requirement for a pseudoclass is the difference of capacity requirement between the corresponding higher class and its base class. For instance, if there were three original service classes L, M, and N, with capacity requirements of C_L, C_M, and C_N, we would create ${}^3C_2$ additional pseudoclasses, say P_{LM}, P_{MN}, P_{LN}, with the capacity requirements of C_M-C_L, C_N-C_M, and C_N-C_L, respectively.

Intuitively, pseudoclasses allow us to retain the allocated bids in lower-capacity classes when a bidder becomes eligible to be considered in higher-capacity classes without having to remove the bids in lower-capacity classes and recompute the allocation. In other words, no reallocation needs to be performed because the natural fairness of the V/C heuristic is maintained within a given class. For all the other classes (with higher capacity requirements), we put the bidder in appropriate pseudoclasses. Since the V/C ratio in a given pseudoclass is kept the same as it would have been in the original higher capacity class, a particular bid is considered in exactly the same sequence as it would have been with no pseudoclasses. However, having pseudoclasses allows us to allocate only the additional capacity required for a bidder with vertically integrated preferences and does not require backtracking to remove the original allocation. To our knowledge, no one has considered exploiting the underlying nature of the product to solve the problem in this manner and this data structure constitutes an important novel contribution of this research.

A bidder bidding in all three original classes (say B_L, B_M, and B_N) would have entries in the base class L with consideration ratio of (B_L/C_L), no entries in classes M, N, and entries in classes P_{LM}, P_{MN} with consideration ratios of (B_M/C_M) and (B_N/C_N), respectively. Another bidder bidding in, say, only classes L and N would have entries in base class L and class P_{LN}. This decomposition of the problem allows direct application of the V/C-based greedy algorithm for resource allocation because of the property described in Proposition 3.

> *Proposition 3: Let a vertically integrated set of services be depicted by $j \equiv \{1,2, ..., J\}$, and let there exist a set of ${}^JC_2$ pseudoclasses created as the difference of requirements for each pair of classes, depicted as $P \equiv \{P_{MN}, M < N; M, N \in j\}$.*

Then, if allocations are made from smaller classes to larger classes using a V/C greedy procedure, and a customer is allocated capacity in pseudoclass P_{MN}, the customer must have allocation either in class M or another pseudoclass P_{LM}, where L < M.

Proof: Note that, for the case under discussion $B_L \leq B_M \leq B_N$ and $(B_L/C_L) \geq (B_M/C_M) \geq (B_N/C_N)$ $\forall$ classes L < M < N.

Now, consider the pseudoclass P_{MN}. The customer's consideration order is decided in this class by the ratio (B_N/C_N). If the customer is being allocated capacity in this pseudoclass then the allocation based on (B_M/C_M) had to be considered earlier because $(B_M/C_M) \geq (B_N/C_N)$. This implies that if the customer's base class was M (i.e., he or she did not bid in any class lower than M) then he or she must have been allocated in class M, or if the base class was lower than M, then he or she must have an allocation in some other pseudoclass P_{LM} with the consideration ratio of (B_M/C_M). Q.E.D.

As per Proposition 3, we treat the pseudoclasses as any other class, except for the fact that bidders winning in the pseudoclass must have a supporting bid allocated in a base class. Note that if the available capacity is less than the capacity for the base class but greater than the capacity requirement of some pseudoclasses, infeasible allocations may result. However, this is easily taken care of by deleting the series of bids of consumers whose base class and pseudoclasses are not feasible as available capacity reduces. While the V/C-greedy approach does not require the explicit knowledge of whether a consumer is winning in the lower classes or not because of Proposition 3, it is convenient to keep track of this for the computation of revenue-maximizing results.

We thus create an enhanced data structure that allows us to keep explicit account of the status, winning or nonwinning, of each bidder in each of the classes he or she bids. Recall that in the simpler one bid per bidder case, all we needed to know was the marginal price–setting bid in each class, and all bids greater than or equal to that bid in the class were also part of the solution (with ties at the margin of capacity resolved by lottery). However, for vertically integrated services, we need an expanded data structure for our preprocessing and bundling procedures of the form *expandedBundle {valueWeightRatio, memberBidders{bidderNo, inBundleSolution}, inSolution}*. The initial ordering of bids in each service class is designed to keep track of the identity of the bidder (*bidderNo*). Moreover, each bundle definition now tracks not just the cardinality of the bundle, along with the value-to-weight ratio, but also the composition of the bundle.

Optimality Gap of the Revised Greedy Approach with Pseudoclasses

Overall, the fairness requirement affected the development of the greedy heuristic for the vertically integrated case in two ways. First, we were forced to consider only the V/C strain of the heuristic. Second, fairness restrained us from making forward

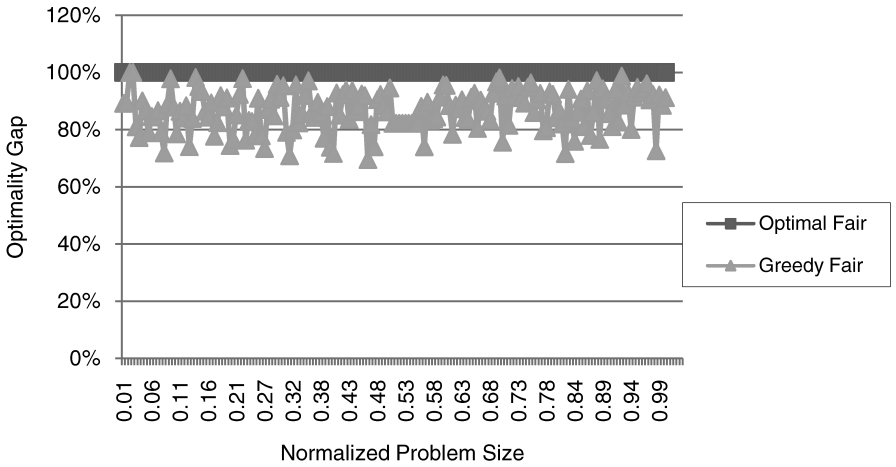

Figure 6. Optimality Gap for Vertically Integrated Services with *Expanded Greedy* Heuristic

search moves of the form presented in *greedy_plus_n* of the fourth section. Despite these limitations, the numerical experiments we conducted show that the heuristic generally performs very well when applied to cases where bidders bid in multiple classes to ultimately win in only one service class, and where they have diminishing marginal valuations.

We simulated 175 auction instances. A maximum bid per unit capacity was set at 7 and for all bidders we tossed a coin to determine whether they bid in a class or not. Valuations were generated for 3 classes, with bids drawn uniformly and satisfying $B_L \leq B_M \leq B_N$ and $(B_L/C_L) \geq (B_M/C_M) \geq (B_N/C_N)$ $\forall$ classes $L < M < N$, where B_L, B_M, and B_N represent the bids in the three classes and C_L, C_M, and C_N the capacities. The respective capacities for the classes were 2, 3, 5. This led to 3 pseudoclasses C_{23}, C_{35}, C_{25} with weights of 1, 2, 3, respectively.

We examine the optimality gap between the *greedy* technique and the enumerative optimal fair allocation described above. Figure 6 graphically illustrates this gap as a function of the problem size.

Overall, the average optimality gap using the *greedy* was about 13 percent with a standard deviation of 6.9 percent. The worst-case performance of *greedy* during the experiments was a 30 percent optimality gap. Observe that the optimality gap is trendless and indicates that the performance of the heuristic is not adversely affected by the problem size.

Conclusions and Directions of Future Research

THIS PAPER ADDRESSES AN IMPORTANT PROBLEM of developing a computational infrastructure for preference elicitation, resource allocation, and pricing in the context of vertically integrated services offered on the Internet. Examples of such services include the selling of webcast or bandwidth streams and computing resources to a set

of buyers who have heterogeneous preferences for the various quality levels, but wish to consume a single service class. We design an asset allocation mechanism modeled after second-price auctions that jointly solves the pricing and allocation problems in a capacity-constrained environment. The resulting formulation for the latter case has characteristics of a nonlinear knapsack problem as well as a quadratic assignment problem—both NP-hard problems.

The paper also contributes to the scant literature on auctions with variable supply. By this we mean that the total number of goods being auctioned is not known ex ante, but is determined as a part of the auction. Part of the reason for lack of research in this area perhaps is due to Lengwiler's [14] negative result regarding lack of incentive compatibility in auctions of variable supply. We show that there are other ways to limit the manipulability of the mechanism. In particular, by deriving conditions required to achieve posterior regret-freeness, as well as by ensuring fairness in allocation, we show that sellers will find it in their interest to use a uniform pricing scheme, such as the MVA.

Another significant contribution of this work is to combine an innovative formulation (considering marginal revenue for knapsack) with a data structure that allows us to use techniques such as *greedy_plus_n* for this complex combinatorial allocation problem. The combinatorial aspect arises in the case of vertically integrated services where a bidder bids in multiple classes but wants to be served in only one. In vertically integrated services, uniform price mechanism alone is not sufficient for fair allocation, and we evaluate the cost of providing fair allocation, without which the market itself may break down. Mathematically, it is equivalent to adding an assignment constraint to a knapsack problem, making the problem even more complex. We develop an innovative data structure that decomposes the service classes by creating pseudo–service classes, and the traditional value-to-cost ratio heuristic for knapsack problems can then be employed directly to provide good solutions to the problem. Future work in this area will look at the revenue-maximizing and efficiency properties of alternative auction mechanisms as well as continuous versus call-based clearing schemes. In addition, demand and capacity here may have a time dimension, as in the grid computing setting of Bapna et al [2]. This, coupled with the consideration of network delays, will serve as a natural extension of the current work.

Acknowledgments: Alok Gupta's research is supported by NSF CAREER grant no. IIS-0301239 and NSF grant no. IIS-0219825 but does not necessarily reflect the views of the NSF. Partial support for this research was also provided by TECI (Treibick Electronic Commerce Initiative) and CIDRIS (Center for Internet Data Research and Intelligence Services), OPIM, School of Business, University of Connecticut.

NOTES

1. Recall, while many, but not all, of the types of services described have a time dimension we assume a call auction setting and do not consider the time dimension in the current work.

2. A minimal cardinality bundle is a bundle that starts at the first customer index where the customer's marginal revenue became less than or equal to zero and ends at the first customer

index where the marginal revenue becomes positive. Note that such bundles can be created as a part of data preprocessing.

3. We operationalize fairness based on Foley's [7] characterization of an envy-free allocation. A fair allocation scheme is defined as one in which there is *no instance wherein, for the purposes of revenue maximization, a consumer with a lower revealed valuation for a given service can get allocated while there exists another, with a higher revealed valuation, for the same service who is denied.*

4. Assume that the valuations have the following pattern for two classes:

Class/Bids	1	2	3	4	5	
1	K	K	$K/2$	$K/3$	$K/3$	...
2	δ	δ	δ	δ	δ	...

The marginal valuations will be

Class/MR	1	2	3	4	
1	K	0	0	$K/3$	...
2	δ	δ	δ	δ	...

The optimal revenue here is NK/3. A *greedy* and *greedy_plus_1* heuristic will both produce the revenue of $K + \delta(N - 1)$. For arbitrarily small δ (i.e., $\delta \rightarrow 0$), we get the revenue of K. Therefore, the worst-case bound is $K/(NK/3) = 3/N$. Note that the worst case for greedy occurs if we replace the valuation of $K/3$ above by $K/2$ and is $2/N$. However, *greedy_plus_1* will produce the optimal revenue in that case.

5. As would be the case with *greedy_plus_1*, that is, when $n = 1$.

References

1. Bapna, R.; Goes, P.; and Gupta, A. Pricing and allocation for quality differentiated online services. *Management Science, 51,* 7 (July 2005), 1141–1150.

2. Bapna, R.; Das, S.; Garfinkel, R.; and Staellert, J. Market design for grid computing. *INFORMS Journal of Computing, 20,* 1 (Winter 2008), 100–111.

3. Bhargava, H.K., and Sundaresan, S. Computing as utility: Managing availability, commitment, and pricing through contingent bid auctions. *Journal of Management Information Systems, 21,* 2 (Fall 2004), 201–227.

4. Bikhchandani, S., and Mamer, J.W. Competitive equilibrium in an exchange economy with indivisibilities. *Journal of Economic Theory, 74,* 2 (June 1997), 385–413.

5. Chellappa, R.K., and Kumar, K.R. Examining the role of "free" product-augmenting online services in pricing and customer retention strategies. *Journal of Management Information Systems, 22,* 1 (Summer 2005), 355–377.

6. deVries, S., and Vohra, R. Combinatorial auctions: A survey. *INFORMS Journal of Computing, 15,* 3 (Summer 2003), 284–309.

7. Foley, D. Resource allocation and the public sector. *Yale Economic Essays, 7,* 1 (1967), 45–98.

8. Greenwood, J., and McAfee, R.P. Externalities and asymmetric information. *Quarterly Journal of Economics, 106,* 1 (February 1991), 103–121.

9. Gupta, A.; Stahl, D.O.; and Whinston, A.B. A stochastic equilibrium model of Internet pricing. *Journal of Economics Dynamics and Control, 21,* 4–5 (May 1997), 697–722.

10. Hansell, S. Google moves to sell space for video spots on network of Web sites. *New York Times* (May 23, 2006) (available at www.nytimes.com/2006/05/23/business/media/23adco.html?ex=1306036800&en=0765a0e63df17ce8&ei=5090&partner=rssuserland&emc=rss).

11. Horowitz, E., and Sahni, S. Computing partitions with applications to the knapsack problem. *Journal of ACM, 21,* 2 (April 1974), 277–292.

12. Jones, J.L.; Easley, R.F.; and Koehler, G.J. Market segmentation within consolidated e-markets: A generalized combinatorial auction approach. *Journal of Management Information Systems, 23,* 1 (Summer 2006), 161–182.

13. Lazar, A.A., and Semret, N. Design and analysis of the progressive second price auction for network bandwidth sharing. Technical Report 487–98–21, Center for Telecommunication Research, Columbia University, New York, 1998 (available at http://comet.columbia.edu/~nemo/telecomsys.pdf).

14. Lengwiler, Y. The multiple unit auction with variable supply. Federal Reserve Board Finance and Economics Discussion Series (FEDS) 28, Washington, DC (June 1998) (available at www.federalreserve.gov/pubs/feds/1998/199828/199828pap.pdf).

15. Martello, S., and Toth, P. *Knapsack Problems: Algorithms and Computer Implementations.* Hoboken, NJ: John Wiley & Sons, 1989.

16. Ng, C.; Parkes, D.; and Seltzer, M. Virtual worlds: Fast and strategy proof auctions for dynamic resource allocation. Paper presented at the Fourth ACM Conference on Electronic Commerce, San Diego, June 9–12, 2003.

17. O'Hara, M. *Market Microstructure Theory.* London: Blackwell, 1995.

18. Pinker, E.; Seidmann, A.; and Vakrat, Y. Managing online auctions: Current business and research issues. *Management Science, 49,* 11 (November 2003), 1457–1484.

19. Rothkopf, M.H., and Harstad, R.M. Modeling competitive bidding: A critical essay. *Management Science, 40,* 3 (March 1994), 364–384.

20. Shapiro, C., and Varian, H. *Information Rules: A Strategic Guide to the Network Economy.* Boston: Harvard Business School Press, 1999.

21. Vickrey, W. Counter-speculation, auctions, and competitive sealed tenders. *Journal of Finance, 16,* 1 (March 1961), 8–37.

22. Wang, R. Auctions versus posted-price selling. *American Economic Review, 84,* 4 (September 1993), 838–851.