Market-Based Models for Digital Signage Network Promotion Management

A Thesis

in

The Department

of

Concordia Institute for Information Systems Engineering (CIISE)

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Applied Science (Quality Systems Engineering) at

Concordia University

Montréal, Québec, Canada

December, 2012

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Kourosh Jamshidi-Gilani**

Entitled: **Market-Based Models for Digital Signage Network Promotion Scheduling**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science in
Quality Systems Engineering

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________ Chair
Dr. A. Schiffauerova

______________________ Examiner
Dr. Z. Tian

______________________ Examiner
Dr. M. Kazemi-Zanjani

______________________ Supervisor
Dr. C. Wang

Approved ______________________
Chair of Department or Graduate Program Director

__________ 20 _____ ______________________
Dr. Robin A. L. Drew, Dean
Faculty of Engineering and Computer Science

Abstract

Digital signage network (DSN) is capable of delivering customized content to designated screens in a real-time or near real-time manner, which provides tremendous potential for building dynamic demand stimulation tools. However current DSN media buying process is mainly carried out through manually conducted negotiation between the DSN operator and the advertisers. This practice does not capitalize the unique technology advantage offered by the newly emerged advertising medium. We propose automated DSN media buying models which allow advertisers to customize their promotion schedules in a highly responsive manner. Specifically, we design a direct revelation mechanism and an iterative bidding model for DSN promotion scheduling. We show that the direct revelation mechanism computes optimal solutions. We evaluate the revenue performance of the iterative bidding model through a computational study. The implementation of the iterative bidding mechanism is also described.

Keywords

Digital signage network, promotion management, auction-based scheduling, customization, Vendor Managed Inventory

Acknowledgements

This thesis wouldn’t be possible without the guidance and support of several individuals who had assisted me in the preparation and completion of this thesis.

First and foremost I would like to express my sincere gratitude to my supervisor, Dr. Chun Wang whose knowledge and patience have been invaluable to me. It was an honor for me in the past two years to work under his guidance as a master’s student and improve my academic knowledge.

Secondly, I would like to thank my colleagues and friends, Nima Eslamloo, Payman Salimi and Zoobin Jafari who have provided me with valuable suggestions and comments through this research.

I would also like to thank my family for their constant support and encouragement, without their support this thesis could not be done. Also a special thanks to my friends for their never ending supports.

Last but not least I have to thank our lab members who have shared all the joyful and unpleasant moments with me during this research. Also, a special thanks goes to the professors and staffs in CIISE for providing a delightful academic surroundings for me to complete this research.

Table of Content

List of Tables

List of Figures

Chapter 1

Introduction and Motivation

1.1 Background

With the advances in digital display and high bandwidth network technologies, digital signage network (DSN) has emerged to be a highly competitive advertising medium which enables organizations to target audiences in a wide range of settings with an unprecedented level of customization and timeliness. During the past decade, advertising industry has seen a significant development in advertisement delivery technologies. Typical examples include Internet ads, mobile phone ads, and DSN. While Internet and mobile phone ads require customers' personal devices as delivery media, DSN is seen as a promising public advertising channel for delivering ads to targeted group of customers. Companies, especially those in the retail and quick service restaurant (QSR) sectors, have realized the potential of DSN as a new medium for marketing and publicity. Some large retailers, chiefly Wal-Mart, have been very successful in their in-store DSNs (Hall, 2010). According to a new market research report from Global Industry Analysts Inc., the global DSN market should reach approximately $14 billion by 2017 (Global Industry Analysts, 2011) .

According to the definition of Wirespring Technologies, any kind of electronic device (usually Plasma or LCD display) which can display an advertisement or a message is called digital signage (An Introduction to Digital Signage, 2011). A digital signage network (DSN) usually connects many displays through a high bandwidth network and these displays are usually controlled by a central computer. DSN enables companies to customize their advertisements in real time. This unique technological advantage could provide several benefits for advertisers in retail and QSR industries. For example, (1) demographic information of targeted customers could be used to customize the advertisement for different times of the day; (2) DSN can be integrated to the inventory management system, which enables the dynamic adjustment of adverting dose to improve inventory management efficiency; (3) the effectiveness of advertisements can be assessed regularly through point-of-sale data and ineffective advertisements could be changed in a timely manner.

In general, the digital signage industry is driven by three economic models, namely *traditional* digital signage*, ad-funded* digital signage, and *ad-supported* digital signage (Gurley, 2010). In a traditional digital signage deployment, the venue owner owns and runs the DSN. This is a model used by corporations, hotels, universities, hospitals, banks, etc. In an ad-funded digital signage deployment, a third party provides a digital signage system to a venue at little or no cost to the venue owner. The third party then seeks to recoup the initial capital outlay plus on-going operational expenses by selling ads to advertisers. An ad-supported system is a hybrid of the ad-funded and traditional models. In an ad-supported system, the venue owner purchases the system, but then works to sell advertising to either off-set the cost of the system or to make money from the system. For ad-funded and ad-supported DSNs, advertising is the most important source of revenue. It is common that the selling of ads is managed by third party network operators. Some operators can integrate networks across many venues into a large scale DSN which is able to provide regional and national coverage. For example, the San Francisco based digital signage network solution provider Premier Retail Network (PRN) manages DSNs from large retailers such as Wal-Mart, Target, and Costco. PRN also manages DSNs from some of the most well-known QSR chains. These networks reside in over 5,700+ locations and deliver 181.1 million commercial viewers per month (PRN, 2012)

1.2 Challenges and Motivation

The responsiveness of advertising is crucial to many retail and QSR settings. Freshness is an important selling point for QSRs and the average shelf life of their items is short. Many QSRs have the policy of throwing unsold items away at the end of the day. In retail settings, managers are often under the pressure of selling perishable products before their expiration dates or emptying the shelves before a specific deadline to make room for new inventories. Traditional media such as weekly flyers, TV and radio cannot offer the needed responsiveness due to the long lead time required for content planning, producing and delivering. The establishment of the digital signage networks provides the infrastructure for building highly responsive advertising solutions. In the Vendor Managed Inventory (VMI) model, which has been adopted by major retailers around the world, vendors have access to up-to-date information regarding inventory levels of their products in the store. They strive to optimize the replenish schedule to reduce overall costs. When the actual demand fails to rise to the forecasted level or when vendors must push more products to the store to accommodate supply chain changes, in-store digital signage becomes an indispensable tool for boosting sales and keeping inventories at a desired level.

Despite the proliferation of DSNs and their enormous capability of reaching and engaging customers, existing DSN promotion management processes do not capitalize the unique technology advantages offered by the newly emerged digital medium. Unlike traditional broadcasting channels, DSN has the capability of delivering customized content to designated screens in a real-time or near real-time manner. This capability offers tremendous opportunities for building a dynamic demand stimulation tool which will considerably enhance retail and QSR supply chain efficiency. In current media buying practice, an advertiser and the DSN operator need to negotiate a contract well before a promotion campaign starts. Once the required promotion schedules and the associated payments are agreed upon, the advertiser's content will be added to the network's playlist and delivered to the screens according to the predefined schedules. This media buying process does not incorporate advertisers' supply chain changes in a timely manner. As a result, the opportunities of utilizing digital signage for high responsive demand stimulation remain under-exploited. In many cases, DSN media buying is also a First-Come-First-Served process, which does not optimize the allocation of screen time across advertisers.

1.3 Approach and Scope

In this thesis, we propose an automated digital signage advertising approach which optimizes advertisers' promotion schedules in response to the changes of their supply chain conditions. Specifically, we design economic-based mechanisms to dynamically distribute screen time to independent in-store brands. We note that by offering an automated promotion scheduling tool, our approach allows advertisers to adjust their promotion schedules "on the fly" according to the updated inputs from the market and the supply chain. Ultimately, we believe that, in today's increasingly virtual and volatile market, the real time or near real time integration of the store's digital signage network with vendors' inventory control strategies will considerably reduce supply chain costs and improve operational efficiency. At system level, the proposed approach balances the promotion requirements across all advertisers and achieves system-wide optimality.

In terms of the scope of this research, we focus on a typical VMI setting. In this type of environment, there can be hundreds of vendors competing with each other for the in-store advertising time. Vendors are assumed to be independent, profit-driven economic units, and they aim at optimizing their own objectives rather than the performance of the system as a whole. Therefore, digital signage promotion scheduling is a decentralized scheduling problem with self-interested agents (we model independent vendors as self-interested agents). This decentralized scheduling problem calls for economic-based models and techniques that take into account these agents' strategic behaviors. In addition, the scheduling process must have the capacity to accommodate dynamic changes in the supply chain as well. We assume an ad-funded or ad-supported DSN system which is managed by a network operator who tries to assign screen times to promotion contents from advertisers, such that the values of all vendors are maximized. A DSN has the capability to contain screens located in several venues, but in this thesis, we restrict our scope to a store- level single venue environment.

1.4 Outline of the Thesis

The rest of the thesis is structured as follows. Chapter 2 reviews related literature including inventory management policies through advertising, various auction models, and auction-based promotion management approaches. Chapter 3 discuss factors that affect customer response to the advertisements, which forms the base of our formulation of the scheduling promotion value presented in Chapter 4. Chapter 4 formulates the DSN promotion scheduling problem. Chapter 5 describes the auction-based DSN promotion scheduling framework. Chapter 6 describes our system implementation and verifies the performance of the proposed approach through a computational study. Chapter 7 concludes the thesis and discusses future research directions.

Chapter 2

Literature Review

This chapter reviews literatures related to the proposed DSN promotion scheduling approach. Since the objective of the proposed approach is to effectively integrate advertising with inventory management, we first survey some existing inventory management policies that involve advertising. We then describe different auction models which form the theoretical base of our auction-based promotion scheduling approach. Finally, we review existing digital signage promotion scheduling approaches and position our approach in the literature.

2.1 Inventory Control through Advertising

An effective marketing strategy is a key in supply chain management. Stores usually use different advertising media to promote their products and also control their inventory. We review several commonly used media in this subsection.

2.1.1 Flyer advertising

Retailers put a high amount of intention towards flyer advertisement. Generally half of the promotion budget of most large supply chain retailers is spent on the flyers printing (Bodapati & Anand, 1999). The flyer's success mostly depends on the volume of flyers, the printed site of the advertisement and the promotion of the flyers (Gijsbrechts, Campo, & Goossen, 2003). According to Gijsbrechts (2003), the type and size of discount and specials on the first page have a direct influence on customer traffic. Customers usually analyse the first page of the flyer and the rest of the flyer doesn’t get the same attention as the first page. Being static is another limitation of the flyers, particularly when the advertised are perishable products. Seasonal promotions are mostly used by franchise supply chains like Dominos, Tim Horton and Subway. They prepare flyers according to each season promotions and post them to their local customers. Also some large retail stores like Provigo and Costco prepare monthly or weekly flyers which help them to control their fresh product inventory. The efficiency of Flyers haven’t been studied that much although they are one of the key media in retail promotion. Flyers are usually successful when there is a huge price decrease on the advertised product.

2.1.2 Narrowcasting

An advertisement can be called a Narrowcast advertisement when customers are receiving the advertisements in a narrow range; a good example could be the shopping cart screens (Ghani, Cumby, Fano, & Krema, 2008). Individual customer shopping patterns can be built by these screens as they can forecast shopping lists through the combination of knowledge –based techniques with statistical and learning algorithms. With the help of the screens individual promotions was offered to the customers. Data mining was used in one of the suggested models to use historical data regarding customer purchases to build up an individual model for customer shopping behaviour prediction.

Another feature provided by the narrowcasting screens is the shopping list which narrows downs the product list for the customer. Also individual promotions and discounts can be offered through these screens. However, for large retailers, they should connect all the screens with their central inventory management system in real time which bring a lot of complication (Uta Juttner, Martin Christopher, Susan Baker, 2005).

2.1.3 E-grocery channels

Now days grocery shopping can be done through internet with the help of ecommerce technologies, this way of shopping is called e-grocery. Some new channels were introduced for e-grocery in recent years, one of them is a handheld device designed for shoppers to enable them ordering products when they are out of store (Bellamy, et al., 2001). This device had functioned like a smart phone; the store inventory was connected to the device and had enabled the customers to choose their desired goods through the device application. The connection was real time which was a great advantage of this device and through this real time connection the store could promote a product by messaging the customers. A limitation of the device is that it complicates user interface which was not suitable for older generations (Bellamy, et al., 2001). In E-grocery channel advertising, advertisements can also be sent to the screens of targeted customers through in-store wireless network (Arvind & Suresh, 2006). Factors like day, time slot, customer's age and sex were considered in the advertisement scheduling process.

2.1.4 Audio broadcasting

Another new method that has seen lots of developments during the recent years is audio broadcasting. This method usually competes with flyer promotions to attract customers (Williams & Diane, 2005). The history of audio/radio advertisement goes back to 50 years ago where retail stores had used it for advertising. In recent years, chain stores use separate radio stations for each of the stores, which enable them to target customers from different regions.

Retail audio networks are another audio broadcasting method. According to Arbitron's market research, 25 % of shoppers in United States think that retail audio advertising can affect their buying decisions (Williams & Diane, 2005). However with this method it is hard to allocate proper promotion schedules to play the ads because the store can only play a single ad at a time.

2.2 Auctions

Since we will propose an auction based DSN promotion scheduling method, we review some typical auction models in this subsection. In recent years auction theory has been applied to the design of a number of real-world markets. In particular, a considerable body of research has been devoted to designing auctions for resource allocation and scheduling problems. Several classes of auctions usually appear in scheduling literature, including single item auctions, Generalized Vickrey Auction, iterative bundle auctions, sequential, and simultaneous auctions. We review them as follows.

Single item auctions

Single unit auctions refer to traditional auction mechanisms such as the English, Dutch, First (or Second) price Sealed-Bid auctions. These auction types are useful for settings where there is a single unit of an item being bought /sold at a time. Single item auctions are trivial in terms of computation. However, they are used extensively in the real world. They have also been used to construct computational markets, such as “thermal market” for distributing air more equitably in a building a through auction-based system (Huberman & Clearwater, 1995), marketed-based control to computer systems (Ferguson, Nickolaou, Sairamesh, & Yemini, 1995), and computational resource scheduling in grid computing (Buyya, 2002). Since single item auctions do not address complementarities among resources, their application to scheduling problems are limited. Sometimes, they can be used as components in sequential or simultaneous auctions described later.

Generalized Vickrey Auction (GVA)

The GVA is a sealed bid auction. In theory, it can find efficient schedules for all decentralized scheduling problems, and it is incentive compatible. This GVA family of mechanisms has become the "Golden" standard in mechanism design. However, it requires each agent to evaluate all bundles of the items. Auctions directly implement GVA for G items quickly become informationally and computationally infeasible, with $2^{|G|}$ bundles to price and $O(|G|^{|G|})$ possible allocations. Therefore, GVA is likely to be practical for smaller problems. The computational complexities of GVA have been studied by various researchers (Fujishima, Yuzo, Leyton-Brown, & Shoham, 1999) (Rassenti, Stephen, Smith, & Bulfin, 1982) (Rothkopf, Michael, Aleksandar, & Harstad, 1998) (Sandholm T. , 2002). Some sophisticated algorithms have produced promising results (Sandholm, Suri, Gilpin,, & Levine, 2005). However, scheduling problems often result in large set of goods to be sold in the market (due to time line Discretization), which lead to bigger size problems, and in turn, inflict heavy burdens in terms of computation. Another limitation with GVA is the so called "lying auctioneer" problem (Sandholm T., 1999) which partially explains why Vickery auction is not widely used in practice, even though it has been proposed since 1960's.

Iterative bundle auction

GVA mechanism has also been implemented as iterative bundle auctions. Iterative bundle auctions are indirect implementations of GVA. This class of auction has practical significance because it addresses the computational and informational complexity of GVA by allowing agents to reveal their preference information as necessary as the auction proceeds, and agents are not required to submit (and compute) complete and exact information about their private valuations. Typical examples of iterative bundle auction include (Parkes & Ungar, 2001) (Parkes & Kalagnanam, 2005). A comprehensive survey for combinatorial auctions can be found in (deVries & Vohra, 2003).

The above mentioned combinatorial auctions are designed for general combinatorial allocation problems. Although they can be applied to scheduling problems, they do not explore the specific problems characteristics derived from the scheduling domain. There are few combinatorial auctions in the literature designed specifically for scheduling problems. In (Kutanoglu E., Wu S. D., 1999), iterative auctions are applied to the job shop scheduling problem. The focus is to investigate the links between combinatorial auctions and Lagrangean relaxation, and to design auctions based on the Lagrangean based decomposition. A "schedule selection game" is presented in (Kutanoglu and Wu, 2006) for collaborative production scheduling. The emphasis is on the incentive compatibility of the mechanism. In (Wellman, Walsh, Wurman, & MacKie-Mason, 2001), the properties of several auction protocols are investigated in the context of decentralized scheduling.

Sequential and simultaneous auctions

Sequential and simultaneous auctions price bundles as the sum price of the individual items. Sequential auctions suppose that the set of resources of interest are auctioned in sequence. Agents bid for resources in a specific, known order, and can choose how much (and whether) to bid for a resource depending on past successes, failures, prices, and so on. Sequential auctions are particularly useful in situations where setting up a combinatorial or simultaneous auctions are infeasible. Typical approaches using sequential auctions for resource allocation include (Boutilier, Dean, & Hanks, 1999)(Engelbrecht-Wiggans and Weber, 1983).

Simultaneous auctions sell multiple goods in separate markets simultaneously. Agents have to interact with simultaneous but distinct markets in order to obtain a combination of resources sufficient to accomplish their task. Real-world markets quite typically operate separately and concurrently despite significant interactions in preferences. A typical example is the series of FCC spectrum auctions (McAfee & McMillan, 1996). In (Reeves, Wellman, MacKie-Mason, & Osepayshvili, 2005) (Wellman, Walsh, Wurman, & MacKie-Mason, 2001), price prediction and bidding strategies for simultaneous auctions are studied in the setting of market-based scheduling. In (Parkes & Ungar, 2001) simultaneous auctions are designed for decentralized train scheduling problems. A review of the uses of economic theory in simultaneous auction design can be found in (Milgrom, 1999).

Sequential and simultaneous auctions tackle the complementarities over resources in the same spirit of general equilibrium theory. These auctions fail when there are no prices that support an efficient solution (the existence problem) and also when agents bid cautiously to avoid purchasing an incomplete bundle (the exposure problem). However, given that these auctions are more practical in terms of computation, they are two important models worth further studying.

2.3 Digital Signage Promotion and Auction-Based Scheduling Approaches

The scheduling approach that we aim to use in this thesis is the dynamic near real-time promotion scheduling which allows multilateral negotiation between advertisers and the digital signage operator. The previous study that is mostly relevant to our work is a virtual marketplace for advertising narrowcast over DSN (Harrison & Andrusiewicz, 2004) , In their paper, the authors introduce a partially automated intermediary, namely the Digital Signage Exchange, which enables the formation of a virtual marketplace for display time on the DSN and facilitates collaboration between buyers and the DSN. The focus is on the concise representation of customer's order and bilateral negotiation between the DSN and a customer. Different from their view, our emphasis is on the system-wide performance in terms of the values across all vendors. Furthermore, our approach achieves good performance in both competitive market and game theoretic settings. In our models, we also use economic models and pricing strategies which allow multilateral negotiation between vendors and the operator.

There are some other approaches in the market. One of them is the mechanism that selects the best ad in response to user requirement (Payne, David, Jennings, & Sharifi, 2006). This approach assumes that the identities of the customers can be discovered through their Bluetooth enabled devices and their historic purchase behaviour data can be retrieved. Auctions can be used to select the appropriate ads which would improve the advertising effectiveness. Another similar approach is to use screens on shopping carts for advertisement based on each customer's historic shopping pattern (Ghani, Cumby, Fano, & Krema, 2008). This method requires the use of individual consumer data, whereas our approach does not require individual customer's behaviour model.

There are several other auction based scheduling approaches like runway landing timeslots scheduling in airport (Rassenti, Stephen, Vernon, & Bulfin, 1982), machine processing time scheduling (Wellman, Walsh, Wurman, & MacKie-Mason, 2001) ,computation resources and network accessing time scheduling (Buyya, 2002),and scheduling of railroad tracks (Parkes & Ungar, 2001). In (Kutangolu and Wu, 1999), iterative auctions are applied to the job shop scheduling problem. The focus is to investigate the links between combinatorial auctions and Lagrangean relaxation, and to design auctions based on the Lagrangean based decomposition. A "schedule selection game" is presented in another paper from the same authors (Kutanoglu & Wu, 2006) , for collaborative production scheduling. The emphasis of the schedule selection game is on the incentive compatibility of the mechanism. At supply chain level, auction-based scheduling framework is applied to the due date management problem (Wang, Wang, Ghenniwa, & Shen, 2011) , the decisions of job selection, due date quotation, and pricing are made concurrently in this paper. Differently from the above mentioned auction-based scheduling applications, the proposed auction framework in this thesis is an iterative multiple homogeneous object auction which is specially designed for the domain of DSN promotion scheduling. The vendors' preferences are modeled based on an advertising response function proposed in the marketing research literature.

In addition to auctions, agent-based models are also applied to scheduling problems. These models are mainly found in agent-based manufacturing control literature (Xue, Sun, & Norrie, 2001) (Cavalieri, Garetti, Macchi, & Taisch, 2000). References and reviews of this line of research can be found in (Shen W., 2002) and (Shen, Wang, & Hao, 2006). Along with other economic based mechanisms, the contract net (Smith,

1980) and its variants have been commonly used in agent-based scheduling as a class of distributed decision making protocols. Differently from our game theoretic assumption, most of the agent-based manufacturing scheduling systems assume cooperative environments where strategic behaviors of agents are not considered.

Chapter 3 Advertising Response Assessment

How customer response to advertisements has been an important question in marketing research. In his experiential generalization of advertisement effectiveness (Clarke D. , 1976), Clarke observed that 90% influence of advertising on sales happened during the advertisement period, which indicates that on site advertising, such as DSN, can be effective. Jones (2006) also confirmed that, to have the highest impact, the advertisements should appear immediately before the purchase, which can be naturally achieved through DSN (Jones J. P., 2006). To assess the benefits of DSN, it is important to know how customers response to the advertisements delivered by DSN. In this chapter, we first describe the impact of technologies on different categories of customers. We then discuss factors that affect customer response in the DSN environment. We also provide an advertising response function which formally captures the relationship between the discussed advertising factors and sale uplifts.

3.1 The Impact of Technologies

In recent years there have been many new technological innovations inside the stores, such as digital signage, wireless devices and smart cards. These technologies have improved the shopping experience of customers and facilitated the purchasing process. Burke (2009) summarized the impact of technologies on different categories of customers (Burke & Raymond, 2009). In general, customers under 25 were more interested in having fun while they shop. This means they are more interested to new technologies like auctions. This group is also positive on factors that could increase the chance of finding a bargain like variable pricing policy. On the other hand, older consumers emphasize more on detailed product information and quality of service. They like more specific details of each product in large dimensions but they don't show high interest for interactive technologies. In terms of gender, male consumers show higher interest in using various types of technology through the shopping process whereas female costumers prefer printed documents for product information. Education also plays an important role in different consumer's decisions on technologies. For example, consumers with higher level of education are more comfortable with non-store channels such as internet whereas less educated consumers prefer receiving all the products directly from the retail store. In the following section, we discuss customer response to a particular technology DSN. However, we only consider a general category. We don't distinguish customers based on their age, gender and levels of education.

3.2 Customer Response in the DSN Environment

In addition to general advertising factors that apply to all channels, as an onsite delivery channel, DSN has some unique factors, such as screen location and promotion time. In this section, we discuss five major factors that affect customer response to the advertisement in the DSN environment, namely product category, promotion dose, content attractiveness, display location, and promotion time. Among them the first three are general; the last two are more specific to DSN.

3.2.1 Product category

The first factor being put in focus is product category. It is commonly understood that customers have different sensitivity to the advertisements of different types of products. Burke (2009) conducted a case study regarding the advertising sensitivity of different categories of products (Burke & Raymond, 2009). The study used the data from Tesco which is a chain retail store in UK. The data has been collected from 102 different advertising campaigns in 100 stores during the years between 2005 and 2007. Each store had at least 40 screens installed inside various zones of the store. Screens are used to show advertisements related to the product in each zone. The study shows that, given similar amount of advertisement for each category of products, different product categories show quite different advertising sensitivity. The following table taken from Burke (2009) gives detailed comparison of advertising sensitivity of product categories.

Product Category	***Average Sales Uplift***
Alcoholic Drinks	12.9%
Impulse Products (Gum, Candy ,Chocolate)	10%
Entertainment	9.2%
Grocery	7.1%
Household	3.2%
Health & Beauty	0.7%

Table 3-1: Product category advertisement effect on sales

As it can be seen in the table, food and entertainment products are more sensitive to advertisement whereas household and beauty products are less responsive to advertising.

3.2.2 Promotion dose

In his influential study, Jones investigated customer response to advertisements based on a brand's market share measured in purchasing occasions and not purchasing volumes (Jones J. P., 2006). He named this research system short-term advertising strength (STAS) which concludes three factors:

- **Baseline STAS:** Brand's market share in households with no prior advertisement.
- **Simulated STAS:** Brand's market share in households with prior advertisement.
- **STAS Differential:** This results from the difference between Baseline STAS and Simulated STAS; it is the measure of short term gain or loss.

Jones study shows that advertisement mostly doesn't influence the decision or need to buy a product but it can influence the choice of a brand, this is done by reminding the brand through continuous advertisement and also focusing the ads on prior brand experience. The study also indicates that as advertising exposure doses increase so do sales but at a diminishing rate from the beginning.

3.2.3 Attractiveness of the advertisement

Attractiveness, or the quality of the advertisement design, is also an important general factor that impacts customer response. In order to make an advertisement attractive, the demographic issues (race, income, age and etc.) of the targeted customers should be taken in to consideration (Harrison & Andrusiewicz, 2003). Harrison and Andrusiewicz (2003) used Ford's advertisement campaigns in Lake Tahoe (Nevada) and Miami (Florida) to illustrate this idea. The perceived quality and attractiveness of an advertisement changes according to the targeted customers, for example showing a snow covered mountain won't attract Miami Latin American people as they cannot relate them self to that kind of environment, but the same image could attract the people of Lake Tahoe as it is related to their demographics. In addition, the choice of the appropriate audio can also impact quality. The Ricky Martin song is a better solution for the people of Miami. However it could be the worst solution for people of Lake Tahoe as they would prefer country style music.

3.2.4 Promotion time

At which time the advertising media should be played is another factor that should be considered in DSN. Ideally, the media should be played when customer traffic is high and when customers are more interested in watching the media. To understand the impact of promotion time on the effectiveness of advertising, Video Mining Corp conducted a study by placing a finger touch digital signage showing paint products positioned in the main corridor of a shopping mall (Burke & Raymond, 2009). A video tracking system was used to record the number of shoppers who interacted with the display. The test lasted 3 months and around 100,000 customers were recorded during the testing period. The following table shows the result.

Time	***Customer Traffic***	***Customer Interaction***
Morning	20.2%	9.6 %
Afternoon	46.4%	12%
Evening	33.3%	14.6%

Table 3-2: Customer Response Based on Time

It is clear from the result that advertisements played in the afternoons and evenings attract more customer attention. Another study shows that customers are more task-oriented earlier on the weekdays and become more interested in browsing later in weekdays (afternoon and evening) and in weekends (Burke & Raymond, 2009). The same pattern is also confirmed by Jones (2006).

3.2.5 Promotion location

In addition to time, location is also an important factor that affects DSN advertising effectiveness. Screens placed in high traffic area certainly attract more customer attention. The following table summarizes the results of a location related customer traffic study conducted by Nielsen Media Research in 2007 (Gutierrez B. P., 2008). During a one week time window, 1,155,489 customers who entered the shop were tracked and the locations they visited were recorded.

The result shows that lobby zone has the highest customer traffic with 100% customer traffic. The result also shows that food and impulse products attract a higher amount of customers in comparison to health and pharmaceutical products which only has 9% of customer traffic.

Store Location	***Average Amount of Audience***
Lobby Zone	100%
Runway –Rear Wall	76%
Runway – Front Wall	64%
Product Zone	62%
Dairy Zone	59%
Meat & poultry zone	45%
Runway-Perimeter	42%
Frozen Food Zone	33%
Bakery Zone	24%
Pharmacy Zone	9%

Table 3-3: Location and audience amount

3.3 DSN Advertising Response Function

To formally describe the relationship between the above mentioned factors and the adverting effectiveness, in this section, we define a function which is a mapping from the *factors* in a promotion schedule to the sales uplift that the schedule generates. The function is usually called the *response function* of adverting in marketing research (Jones J. P., 2006). The general response functions in the literature only capture the three general factors: product category, dose, and advertisement quality. We first describe the general response function proposed in the literature. We then specify our DSN response function by extending the general response function.

While the general response function is usually assumed to be either S-shaped or concave in the literature (Khouja & Robbins, 2003), many researchers support the concave response function presented by Jones (Jones J. P., 2006). This function indicates that as exposure doses increase so do sales but at a diminishing rate from the beginning. Assume that $D_i(\tau\ )$ is a random variable representing the demand of vendor i's product. Because of diminishing returns of advertising, the expected value of $D_i(\tau\ )$ is a concave increasing function in the exposure dose of τ, which we assume to be given by

$$E\{D_i(\tau)\} = \mu_0 + \mu_0\beta_i(\tau)^{\alpha_i}$$

where μ_0 is the average demand without advertising; $\alpha_i(0 \le \alpha_i \le 1)$and β_i indicates the sensitivity of the product to advertising, which can be affected by factors such as the category of the product and seasonality; α_i determines the shape of the diminishing function, which is mainly related to the quality of content i. For any $\beta_i > 0$, the larger the value of α_i, the more effective is the advertising. A similar response function with advertising expenditure as input is also proposed in (Khouja & Robbins, 2003).

In the DSN environment, special factors, such as promotion time and location should also be considered when assessing the effectiveness of a promotion schedule. Given a promotion schedule that contains multiple promotion time periods and locations, constructing an accurate response function can be challenging. In this thesis, we assume that the impact of time and location can be converted to the impact to the dose. We add two impact factors, l_i for location and s_i for starting time, to model the impact of location and time to the promotion schedules effectiveness, where $0 \le l_i \le 1$, $0 \le s_i \le 1$, and τ is the dose at best location and best time. Including these two factors, the DSN response function can be written as:

$$E\{D_i(\tau)\} = \mu_0 + \mu_0\beta_i(l_i s_i \tau)^{\alpha_i}$$

Given a product to be promoted and the media content, μ_0, β_i, and α_i are constants which can be determined by historical data. In modern DSNs, the vast amount of real time point-of-sale data has been integrated into the digital signage management systems (Yackey, 2010). For an advertisement that has been running for a certain periods of time, it is reasonable to assume that the constants can be determined with good accuracy. In addition, the l_i and s_i can also be predicted by using the existing studies, such as Burke (2009) and historical data.

It is important to note that the response function developed here is a general template and may not reflect the unique characteristics of a DSN environment. To apply it to a specific DSN setting, the parameters and even the function structure need to be adjusted. In fact, we have already considered developing sales response models which take into consideration much more factors in our future research. In addition, in our promotion scheduling framework design, the response function is a module that is used to predict expected sales uplift given the content's exposure dose. Should more suitable models are developed for a DSN, the response function can always be replaced without altering other modules of the scheduling framework. Using the advertising response function, a vendor can predict the expected sales generated by the exposure dose of τ time units provided by a promotion schedule.

Chapter 4
The DSN Promotion Scheduling Problem

In this section, we define the DSN promotion scheduling problem. We assume an ad-funded or ad-supported setting where the DSN is managed by a network operator who allocates screen time to promotion contents from advertisers. While a DSN can contain screens located in multiple venues, to simplify the modeling, we restrict our attention to a store-level single venue environment.

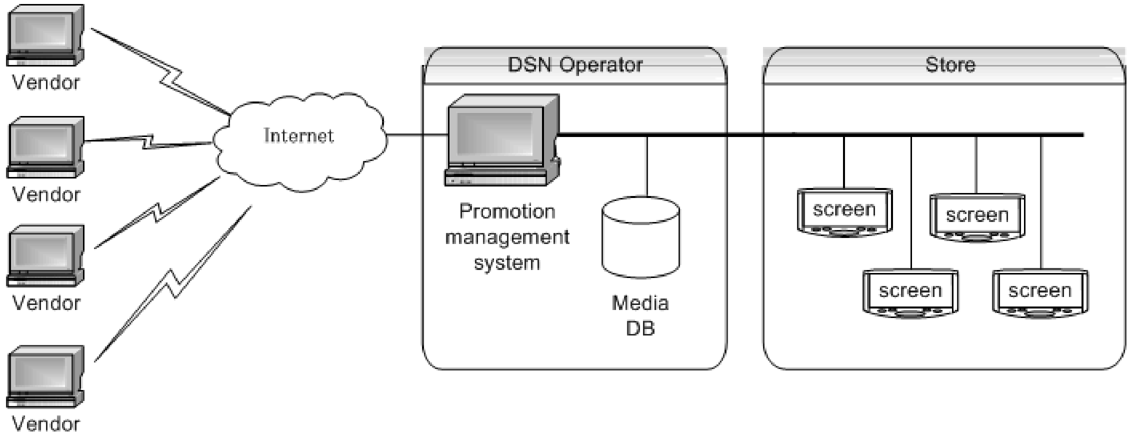


Figure 4-1: Single venue DSN promotion scheduling environment

4.1 The Single Venue DSN Promotion Scheduling Environment

The DSN promotion scheduling environment consists of a DSN, an operator, and a group of advertisers (we call them vendors in the rest of the thesis[1]). As shown in Figure 4-1, all screens in the DSN are located in a store. These screens are connected to the operator's promotion management system through a high speed network. They are also intelligent screens meaning that they are controlled by embedded computers which can store the promotion contents streamed to them and play the contents based on the predefined promotion schedule.

Vendors are also connected to the promotion management system through the Internet. A vendor has access to real time information regarding the inventory levels and sales data of their products in the store. This is particular the case in VMI (Vendor Managed Inventory) (Waller, Johnson, & Davis, 1999) which has been widely adopted by large retailers. In VMI, vendors, rather than retailers, are responsible for monitoring the inventory levels of their products and replenish the inventory as needed. They strive to optimize the replenish schedule to reduce overall costs. When the actual demand fails to rise to the forecasted level or when vendors must push more products to the store to accommodate supply chain changes, they have a strong incentive to use highly

[1] In the case of the DSNs owned by large retailers, screen time is typically sold to companies that do business with them (e.g. product suppliers) and both the retailer and advertiser benefit from the success of the ads. Without loss of generality, we assume that advertisers are product suppliers in this thesis. We call them vendors, which is consistent with the terminology used in the widely adopted VMI (Vendor Managed Inventory) retail model.

responsive demand stimulation tools, such as DSN, for boosting sales and keeping inventories at a desired level. Suppose that the store's opening hours of a day are divided into several promotion windows, within which the operator schedules media contents from vendors. The scheduling process is run window by window, that is, for a particular promotion window, the customization process has to be run within the window before it. In this way, vendors have the opportunity to update their promotion requirements for a window before it starts to accommodate dynamic changes in their respective supply chains. The width of a window can be adjusted based on responsiveness requirements of vendors.

4.2 Formulation of the DSN Promotion Scheduling Problem

The task of the promotion scheduling is to allocate limited screen time to the content of vendors such that the operator's business objective can be achieved. In this thesis, we define the network operator's business objective as to maximize the overall benefits that vendors receive. In microeconomic terms, this is to maximize the social welfare (Mas-Colell, Whinston, & R., 1995) of all vendors. The objective of maximizing social welfare is commonly used in the microeconomics and mechanism design literature as a criterion for measuring resource allocation efficiency. It is suitable for the settings such as VMI where vendors have repeated transactions and long term relationship with the operator. To quantify the business objective, we introduce a definition called the *promotion value* of a schedule which is defined as the measure of the benefits that a vendor receives from the sales uplift generated by a promotion schedule. These benefits can be, for example, increased profits, reduced wastes, and reduced inventory holding costs.

4.3 Promotion Value

The advertisement response function in Section 3.3 provides the sales uplift based on a specific dose of advertisement. Given the sales uplift generated by a schedule, we assume that a vendor always has a mechanism to assign a *promotion value* to it. This value is the price that the advertiser is willing to pay given its supply chain situations, such as inventory levels and scheduled transportation, delivery and promotion. This mechanism is usually vendor dependent and may involve experts' subjective judgments. We do not identify the specific forms of the mechanism in this thesis. In general this mechanism should take vendor's current supply chain circumstances, such as holding costs, product expiration date, and demand forecasts into consideration.

4.4 Problem formulation

Consider a simplified store level DSN promotion allocation problem with a set of n vendors, denoted N, who want to promote their products on a set of m digital screens. Given a promotion window which consists of w time units (for example, w minutes or seconds), the operator wishes to allocate the mw time units to vendors to maximize the sum of promotion values across all vendors. The time dimension is not an issue in our simplified model and we just deal with the allocation problem but this can be considered in future work. An allocation is called a promotion schedule which can be represented by a coverage vector $\mathbf{x} = (x_1, \dots, x_n\ )$, where $x_i,\ \ 1 \le i \le n$, is the number of time units assigned to vendor i. Let $v_i(\tau)$ denote the promotion value of vendor i for advertising within a number of τ time units. The promotion scheduling problem (PSP) can be expressed using the following integer programming model.

$$max \sum_{i=1}^{n} v_i(x_i) \qquad (1)$$

subject to

$\sum_{i=1}^{n} x_i \leq mw$, (2)

$x_i \in \{0,1,2,\dots,lmt_i\}$, $1 \leq i \leq n$, (3)

where lmt_i is the maximum number of time units that vendor i can purchase, called *vendor limit*. The purpose of setting lmt_i is to prevent a single or small group of vendors from dominating the screens. lmt_i is set by the operator. The set of constraints (2) ensures that the overall time units allocated to vendors do not exceed the available time units. Constraints (3) ensure that the number of time units obtained by any vendor is an integer and also restricted by their upper limit. The constraints of the model are linear. However, the objective function is not linear since the promotion value functions $v_i(x_i)$ are not linear in the number of time units assigned to vendors.

Chapter 5
The DSN Promotion Scheduling Framework

DSN promotion scheduling is a decision making process which allocates screen time to vendors based on their advertising demands. The allocation can be naturally implemented using a market mechanism in which the operator is the seller, the brand advertisers are buyers, and the screen time units are items to be sold in the market. In a typical DSN environment, vendors are independent, profit-driven economic units. They may have conflicting and competing promotion schedule requirements. They also possess private information relevant to their needs (such as the promotion value they place on a particular schedule). Vendors could be direct or potential competitors. They are reluctant to reveal their sensitive supply chain information. To recognize this independence, we treat them as self-interested agents that aim at optimizing their own objectives rather than the performance of the system as a whole. Designing a DSN scheduling framework for self-interested vendors calls for mechanism design techniques that take into account the vendors' strategic behaviors. We design DSN scheduling models for the game theoretic settings in which vendors will behave strategically to optimize their objectives. We first present a direct revelation mechanism (DRM) (Mas-Colell, Whinston, & R., 1995) for promotion scheduling. In the DRM model, vendors are only given one opportunity to submit their promotion valuation to the operator. We then extend the DRM model by designing an iterative procedure which allows vendors to gradually submit their valuation on a need to reveal basis.

5.1 The DRM Model

The DRM model is an interactive procedure between the vendors and the operator, which allows vendors only one opportunity to submit their promotion valuation. We first describe the overall procedure. We then propose algorithms for computing vendors' coverage vector and payments.

5.1.1 The procedure

The procedure contains the following steps:

Step 0: **Initialization**. Each vendor computes $v_i(\tau)$ for $1 \leq \tau \leq lmt_i$ using the response function and vendor specific mechanism for computing promotion values.

Step 1: **Submission**. All vendors submit their complete promotion valuation to the operator. For vendor i, the complete valuation can be represented by a vector of size lmt_i, each element of the vector storing the promotion value of a specific dose. Instead of submitting a promotion value vector, vendor i can of course submit their response function and promotion value estimation mechanism, so the operator can calculate the vendor's promotion value for any exposure dose. However, submitting the response function and promotion value estimation mechanism exposes more private information. Therefore, vendors may be reluctant to do that. In addition, value estimation mechanisms are vendor specific and often involve expert judgment, which is difficult to be formulated and transferred to the operator. In our DSN scheduling framework, we assume that vendors submit their promotion value vectors.

Step 2: Computing coverage vector and payments. Taking all vendors' promotion value vectors as input, the operator computes customized doses for each vendor by solving the PSP model. An algorithm for computing the PSP model is described in the next subsection.

Step 3: Notification and deployment. The operator notifies vendors the awarded coverage vectors. After the notifications are confirmed by the vendors, the operator constructs a play list for each of the vendors based on the awarded coverage vectors.

5.1.2 Algorithm for computing coverage vectors

We propose an algorithm for computing coverage vectors for vendors using their promotion value vectors as inputs. This algorithm makes use of the concept of vendor's *marginal value* of obtaining a time unit. For a promotion schedule with exposure dose τ, for vendor i, the vendor's marginal value of obtaining the τth time unit is defined as $v_i(\tau) - v_i(\tau - 1)$, denoted $\tilde{v}_i(\tau)$ which is the value increment brought by the τth time unit given the existing exposure dose $\tau - 1$. Using the value of $\tilde{v}_i(\tau)$, $1 \leq \tau \leq lmt_i$, we can construct marginal value vectors of vendors (Table 5-5 in Section 5.2.2 gives an example of vendors' marginal value vectors). Marginal value vectors are used in the proposed algorithm for computing coverage vectors. The algorithm includes the following steps:

Step 1: Computing marginal values. For each vendor, compute marginal values for $1 \leq \tau \leq lmt_i$ based on their promotion value vector using the formula $\tilde{v}_i(\tau) = v_i(\tau) - v_i(\tau - 1)$.

Step 2: Sorting. Sort the marginal values of all vendors in a decreasing order.

Step 3: Computing coverage vector. Take the first mw marginal values as winning marginal values. Break the ties randomly. Count the number of winning marginal values that belongs to each of the vendors. Assign each vendor the number of time units according to the number of their winning marginal values.

The algorithm solves PSP optimally as stated in the following proposition.

Proposition 1 *The proposed algorithm for computing coverage vector* solves PSP *optimally.*

*Proof***.** Given the definition of the marginal value $\tilde{v}_i(\tau) = v_i(\tau) - v_i(\tau - 1)$, we have that $v_i(\tau) = \tilde{v}_i(\tau) + v_i(\tau - 1) = \sum_{k=1}^{\tau} \tilde{v}_i(k)$. Suppose that $\hat{\mathbf{x}} = (\hat{x}_1, \dots, \hat{x}_n)$ is the coverage vector returned from the algorithm. The overall value of $\hat{\mathbf{x}}$ is $\sum_{i=1}^{n} v_i(\hat{x}_i)$ which can also be represented by marginal values in the form of $\sum_{i=1}^{n} \sum_{k=1}^{\hat{x}_i} \tilde{v}_i(k)$ which is the sum of selected marginal values of vendors. Since the algorithm selects the largest mw marginal values across all vendors, we know that $\sum_{i=1}^{n} v_i(\hat{x}_i)$ is the largest value can be achieved with the mw time units in the promotion window. It follows that $\hat{\mathbf{x}} = (\hat{x}_1, \dots, \hat{x}_n)$ solves PSP optimally.

5.1.3 A worked example

In this section an example will be illustrated to show how the DRM Model works, in this example we have three vendors (n), four screens (m) and three time units (w). So the following will be our summary of the facts:

w=3 m=4 n=3 max $\sum_{i=1}^{n} v_{i\,(x_i)}$

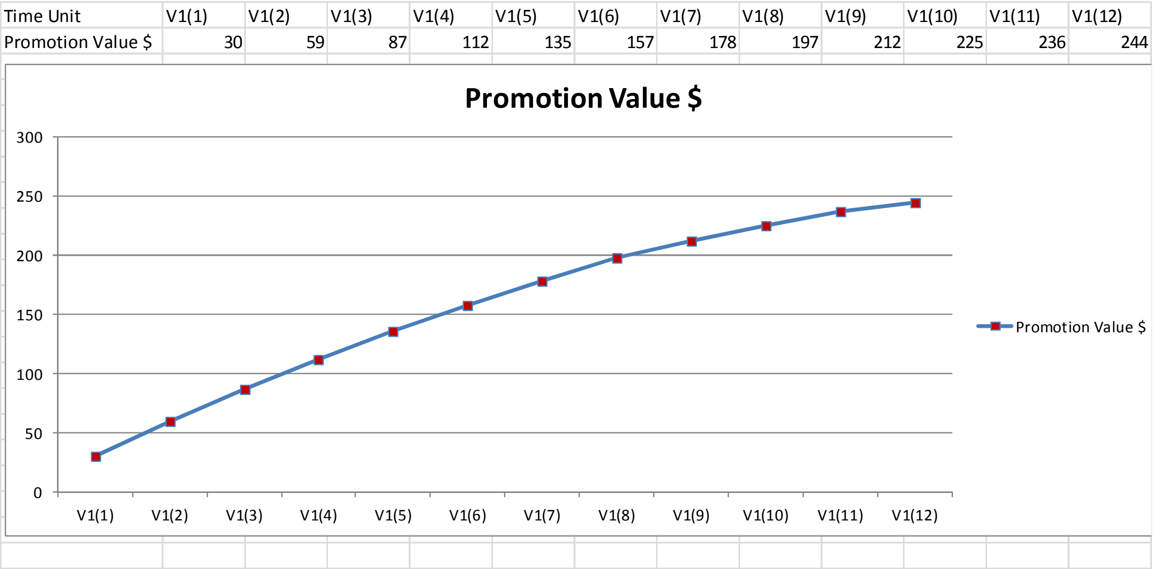
mw= 3 * 4= 12 available time units

$\hat{v}(\tau) = v(\tau) - v(\tau - 1)$ Marginal value

So according to the information we have there are twelve available time units which can be assigned to three different vendors, this requires the vendors to submit their complete promotion valuation to the operator.

Step1 and 2 (Submission and coverage vector):

In this step vendors propose twelve promotion values for the twelve available time units.



Time Unit	V1(1)	V1(2)	V1(3)	V1(4)	V1(5)	V1(6)	V1(7)	V1(8)	V1(9)	V1(10)	V1(11)	V1(12)
Promotion Value $	30	59	87	112	135	157	178	197	212	225	236	244

Figure 5-1: Vendor 1 Promotion Valuation

Time Unit	V2(1)	V2(2)	V2(3)	V2(4)	V2(5)	V2(6)	V2(7)	V2(8)	V2(9)	V2(10)	V2(11)	V2(12)
Promotion Value	34	69	101	131	156	176	191	204	215	223	228	230

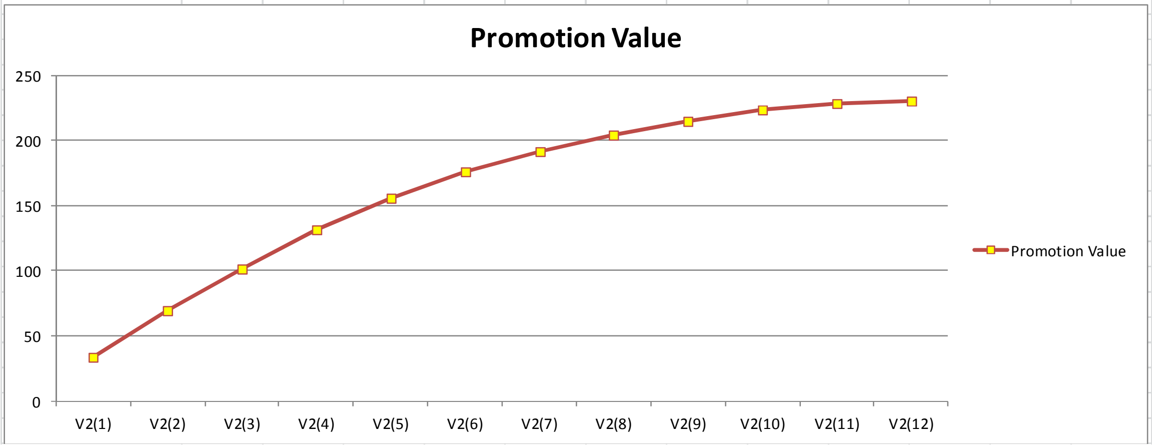


Figure 5-2: Vendor 2 Promotion Valuation

Time Unit	V3(1)	V3(2)	V3(3)	V3(4)	V3(5)	V3(6)	V3(7)	V3(8)	V3(9)	V3(10)	V3(11)	V3(12)
Promotion Value	33	59	83	105	126	146	165	182	197	211	223	234

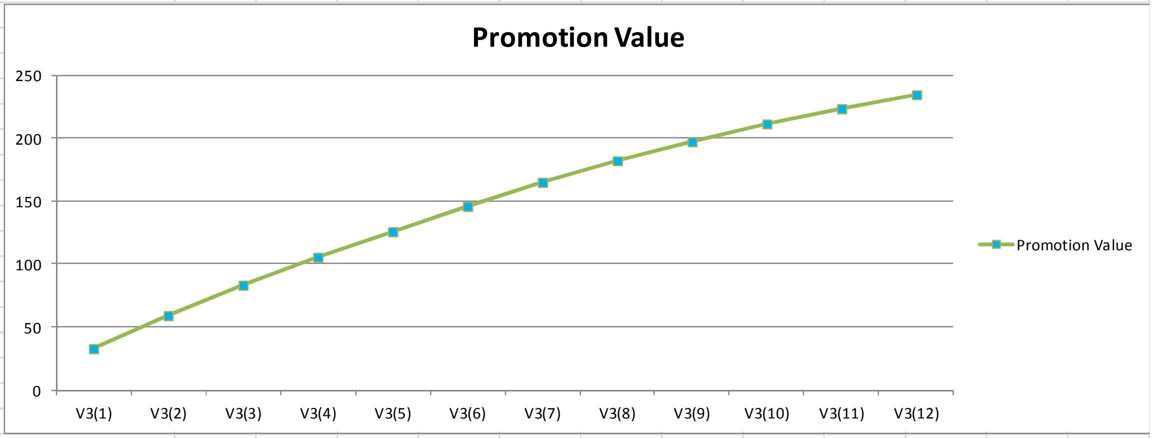


Figure 5-3: Vendor 3 Promotion Valuation

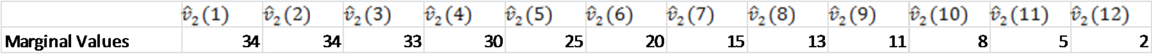
After getting the complete promotion value of each vendor we should compute the marginal values of each vendor, the following tables show the complete marginal values:

Vendor (1)

	$\hat{v}_1(1)$	$\hat{v}_1(2)$	$\hat{v}_1(3)$	$\hat{v}_1(4)$	$\hat{v}_1(5)$	$\hat{v}_1(6)$	$\hat{v}_1(7)$	$\hat{v}_1(8)$	$\hat{v}_1(9)$	$\hat{v}_1(10)$	$\hat{v}_1(11)$	$\hat{v}_1(12)$
Marginal Value	**30**	**29**	**28**	**25**	**23**	**22**	**21**	**19**	**15**	**13**	**11**	**8**

Table 5-1: Vendor 1 Marginal Value

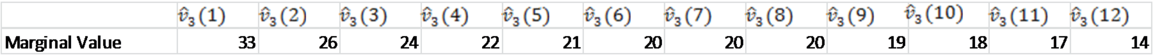
Vendor (2)



	$\hat{v}_2(1)$	$\hat{v}_2(2)$	$\hat{v}_2(3)$	$\hat{v}_2(4)$	$\hat{v}_2(5)$	$\hat{v}_2(6)$	$\hat{v}_2(7)$	$\hat{v}_2(8)$	$\hat{v}_2(9)$	$\hat{v}_2(10)$	$\hat{v}_2(11)$	$\hat{v}_2(12)$
Marginal Values	**34**	**34**	**33**	**30**	**25**	**20**	**15**	**13**	**11**	**8**	**5**	**2**

Table 5-2: Vendor 2 Marginal Value

Vendor (3)

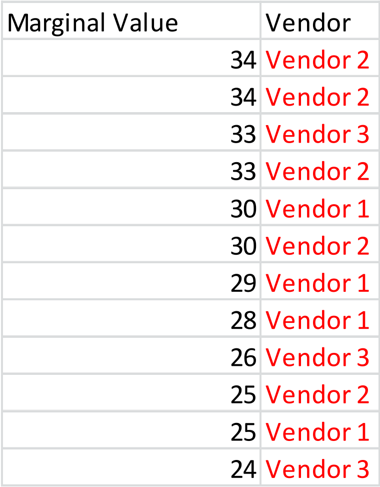


	$\hat{v}_3(1)$	$\hat{v}_3(2)$	$\hat{v}_3(3)$	$\hat{v}_3(4)$	$\hat{v}_3(5)$	$\hat{v}_3(6)$	$\hat{v}_3(7)$	$\hat{v}_3(8)$	$\hat{v}_3(9)$	$\hat{v}_3(10)$	$\hat{v}_3(11)$	$\hat{v}_3(12)$
Marginal Value	**33**	**26**	**24**	**22**	**21**	**20**	**20**	**20**	**19**	**18**	**17**	**14**

Table 5-3: Vendor 3 Marginal Value

The next step is to sort all these marginal values in a decreasing order and the first twelve marginal values are the winners.

Winners (Sorted):



Marginal Value	Vendor
34	Vendor 2
34	Vendor 2
33	Vendor 3
33	Vendor 2
30	Vendor 1
30	Vendor 2
29	Vendor 1
28	Vendor 1
26	Vendor 3
25	Vendor 2
25	Vendor 1
24	Vendor 3

Table 5-4: Winners -Sorted Marginal Values

5.2 Iterative Bidding Model

The iterative bidding model allows vendors to gradually submit their valuation on a need to reveal basis. Specifically, the mechanism is implemented using a multiple-object ascending auction. We first present the ascending auction protocol. We then describe two payment schemes which are suitable for *competitive market* settings where vendors are "price takers" and *game theoretic* settings where vendors will behave strategically to influence the price of time units. The auction prices time unit. It does not distinguish time units of different screens in a store.

5.2.1 Ascending auction

Starting with a reserve price, the operator announces a current price p of a time unit. Vendors report back the quantity demanded at p. We assume that vendors follow a myopic best-response bidding strategy (this strategy will be justified later in this subsection by the application of "clinched" payment scheme). According to the best-response bidding strategy, vendors maximize their utilities given the current price. That is, the quantity demanded by vendor i at price p is $q_i(p)$ such that $\tilde{v}_i\big(q_i(p)\big) > p$. In words, a vendor will demand a time unit if its marginal promotion value is greater than the price they pay for it. If the aggregate demand from all vendors is greater than the available supply, that is$\sum_{i=1}^{n} q_i(p) > mw$, the operator raises the price by an increment ε. The process continues until the market clears, that is the aggregate demand of all vendors is less than or equal to the available mw time units at market clearing price p^*. If $mw = \sum_{i=1}^{n} q_i(p^*)$, the supply matches the aggregate demand exactly at the clearing price p^*. In this case, all selected mw marginal values are greater than p^* and all unselected marginal values are less than or equal to p^*. In other words, the operator selects the largest mw marginal values across all vendors. It follows that if supply matches demand exactly at the clearing price, the auction solves PCP optimally. On the other hand, if $\sum_{i=1}^{n} q_i(p^*) < mw$ at p^*, there are $mw - \sum_{i=1}^{n} q_i(p^*)$ free time units left at termination. Given the iterative bidding structure, the only way to reduce the number of free time units is to decrease ε. However, for large scale problems, ε cannot be too small since a smaller ε leads to higher number of rounds of bidding, which increases computation overhead. Therefore, the issue of free time units needs to be addressed by an *additional allocation procedure*. We randomly assign the free time units to the demand dropped when price increases from $p^* - \varepsilon$ to p^*. The implementation of the additional allocation procedure and the analysis of its impact will be a future work.

5.2.2 Vendors' payments

At the termination of auction, the operator charges vendors for the time units assigned to them. A straightforward payment scheme is to require vendors to pay their final demands evaluated at the final clearing price. We refer to this payment scheme as *uniform payments* as vendors pay time units at the uniform clearing price. In the case that $mw > \sum_{i=1}^{n} q_i(p^*)$, some vendors can be assigned extra time units after termination. For these time units, vendors are awarded ε discount below the clearing price.

Time Units	Vendor A	Vendor B
1	$99.80	$100
2	$83.20	$87.10
3	$77.90	$82.70
4	$74.70	$80.10
5	$72.40	$78.10
6	$70.60	$76.60
7	$69.10	$75.40
8	$67.90	$74.30
9	$66.80	$73.40
10	$65.90	$72.70

Table 5-5: Vendors' marginal values for the ten time units

Price	Vendor A	Vendor B
$65	Demand=10, Clinched=0 Cumulative payment=0	Demand=10, Clinched=0 Cumulative payment=0
$65.90	Demand=9, Clinched=0 Cumulative payment=0	Demand=10, Clinched=1 Cumulative payment=65.9
$66.80	Demand=8, Clinched=0 Cumulative payment=0	Demand=10, Clinched=2 Cumulative payment=132.7
$67.90	Demand=7, Clinched=0 Cumulative payment=0	Demand=10, Clinched=3 Cumulative payment=200.6
$69.10	Demand=6, Clinched=0 Cumulative payment=0	Demand=10, Clinched=4 Cumulative payment=269.7
$70.60	Demand=5, Clinched=0 Cumulative payment=0	Demand=10, Clinched=5 Cumulative payment=340.3
$72.40	Demand=4, Clinched=0 Cumulative payment=0	Demand=10, Clinched=6 Cumulative payment=412.7
$72.70	Demand=4, Clinched=1 Cumulative payment=72.7	Demand=9, Clinched=6 Cumulative payment=412.7
$73.40	Demand=4, Clinched=2 Cumulative payment=146.1	Demand=8, Clinched=6 Cumulative payment=412.7
$74.30	Demand=4, Clinched=3 Cumulative payment=220.4	Demand=7, Clinched=6 Cumulative payment=412.7
$74.70	Demand=3, Clinched=3 Cumulative payment=220.4	Demand=7, Clinched=7 Cumulative payment= 487.4

Table 5-6: Summary of the changes of demand, clinched time units, and cumulative payments of vendors as the price advances

In competitive market settings where there are a large number of competing vendors and no vendors dominate the market, uniform payments motivate vendors to adopt myopic bidding strategy. They truthfully report their quantity demanded at a price since, in a competitive market setting, the impact of an individual's strategic behavior on the market price is negligible. However, in game theoretic settings where there are a small number of competing vendors or a small group of vendors dominate the market, uniform payments are not sufficient in terms of motivating vendors to adopt myopic bidding strategy. As pointed out in (Ausubel L. M., 2004), dominant vendors have strong incentive to reduce demand in order to depress the price. For the game theoretic settings, we adopt a "clinched" payment scheme which was originally proposed by Ausubel (2004). The payment scheme computes VCG payments in general multiple-object ascending auctions, which makes the auctions strategic proof. Therefore, in our DSN promotion scheduling, the "clinched" payment motivates the vendors to adopt the myopic bidding strategy. In the following, we explain how this payment scheme is applied to DSN promotion scheduling through an illustrative example.

Consider a simplified DSN setting with two vendors, denoted A and B, ten available time units to be sold. Vendors have diminishing marginal values over the number of time units, which are shown in Table 5-5. We adopt the "clinching" rule to credit time units to vendors along the bidding process. Table 5-6 summaries the changes of demand, clinched time units, and cumulative payments of vendors as the price advances. Suppose that the auction begins with the operator announcing a price of $65. Vendors A and B, adopting myopic bidding strategy according to the marginal values of Table 1, would response with demands of 10 and 10 respectively. Since the aggregate demand of 20 exceeds the available supply of 10, the operator subsequently adjusts the price by adding a price increment $\varepsilon = 0.1$. The aggregate demand is still 20 with the updated price. The operator increases the price continuously. When price reaches $65.9, vendor A reduces their quantity demanded from 10 to 9. At this point, the aggregate demand of 19 continues to exceed the available supply of 10. However, from Vendor B's perspective, the competing demands from other vendors (only Vendor A in this example) are 9 which is less than the available supply of 10. If Vendor A bids monotonically, Vendor B is now mathematically guaranteed to win at least one time unit. In the language of Ausubel (2004), Vendor B has "clinched" winning one time unit. The auction then awards Vendor B one time unit at the clinching price of $65.9. Vendor A reduces their quantity demanded to 8 at $66.8 and Vendor B clinches to win another time unit at $66.8. Since there is still excess demand, the price must rise further. Vendor A continues to drop their demand at $67.9, $69.1, $70.6, and $72.4. At $72.7, Vendor B starts dropping their demand to 9 and Vendor A clinches to win one time unit. The price continues to rise. At $74.7, the market clears. Vendor A, who had already clinched time units at $72.7, $73.4 and $74.3, wins 3 time units and pays the cumulative payment of $220.4. Vendor B, who had already clinched time units at $65.9, $66.8, $67.9, $69.1, $70.6, $72.4 and $74.7, wins 7 time units and pays the cumulative payment of $487.4. The overall value of the assignment is $840.9. Observe that, for each vendor, the quantity demanded at the final price equals the sum of all time units credited along the way. Since many of the credits occurred at earlier prices, however, vendors' payments do not generally equal their final demands evaluated at the final price. Rather, the vendors' payments are related to those from the VCG mechanism (Ausubel L. M., 2004), justifying the myopic bidding strategy assumed previously.

5.2.3 A Worked Example with Four Vendors

This example is different from the previous example because a different approach is used for the promotion scheduling; this approach is named as Iterative bidding model. This model happens in an ascending auction environment which means there is a start price at the beginning of the auction and each round there should be at least a standard price increment till the demand matches the supply.

In our example we have an auctioneer who wants to sell 6 time units to four available vendors who are willing to bid for these time units, there is a certain limit for each vendor they can't get more than 4 time units, In order to give the time units to the vendors who value it the most an iterative auction is needed, The following table and diagram show the marginal values of each vendor for different amount of time units.

	Vendor A	Vendor B	Vendor C	Vendor D
Marginal Value $ (1 time unit)	45	57	51	60
Marginal Value $ (2time units)	25	35	42	49
Marginal Value $ (3time units)	20	28	30	29
Marginal Value $ (4time units)	17	12	21	14

Table 5-7: Worked Example with four vendors

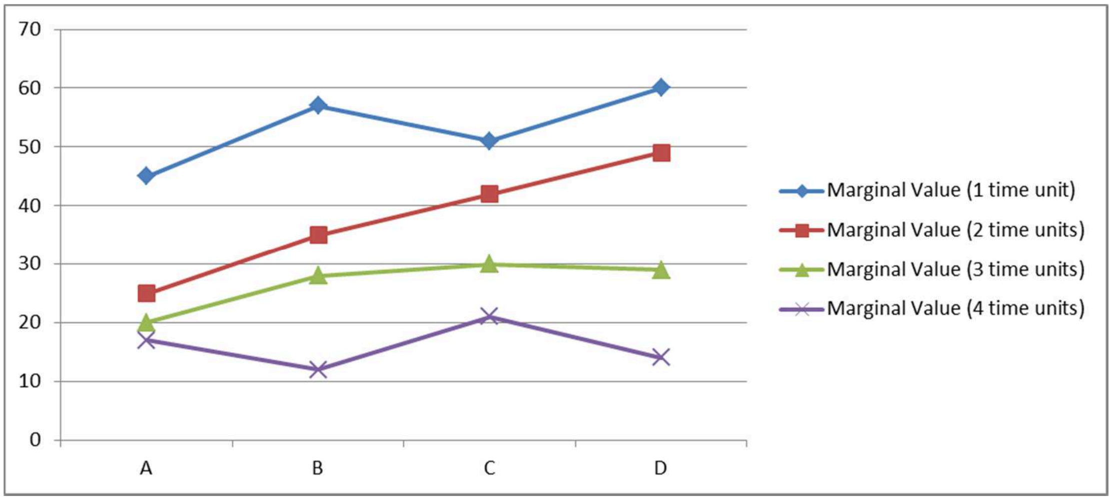


Figure 5-4: Worked example marginal values

The next step is to perform the auction rounds to determine the winning vendors. The price increment standard for this auction is 5 $.

Vendors	4
Price Increment limit (∈)	5$
Available Time Units	6
Time units per vendor limit	4

Round 1

The starting price for this auction is 20$, this means all the vendors who are willing to pay less or equal to 20 $ for their time units will get out of the bidding for that specific time unit, this table illustrates the situation that we are going to face in the first round of auction :

Price	A	B	C	D
20$	2	3	4	3

Figure 5-5: Round 1

The demand is 2+3+4+3=12 but the supply is 6 so the auction should continue to the next round, as it can be seen in the table vendors A, B and D get out of their bidding for the 4 units because their marginal value was less than 20$, also vendor A loses his bidding for three time units as it is equal to 20$.

Round 2

The price for next round is 25$ and the table below shows the new bidding situation for all vendors:

Price	A	B	C	D
25$	1	3	3	3

Figure 5-6: Round 2

As it can be seen in the table Vendor A gets out of his biding for 2 and 3 time units because they were less than 25$, also the vendor C get out of his bidding for the 4 time units, The demand is right now at ten time units but our supply is at six, that means the auction should continue to the third round to determine the winners.

Round 3

In this round the auction price is increased with 5 $ which means the price is at 30$, the following situation would happen to our bidders table:

Price	A	B	C	D
30$	1	2	2	2

Figure 5-7: Round 3

As it can be seen in the table bidders B, C, D each loose another time unit because of the ascending auction but they all also get a clinched winning time unit at 30 $ because in the ascending auction environment the summation of the rest of the bidders in comparison to them equals five (1+2+2) while the supply is six, this means they are all going to win one time unit for sure.

The market is still not cleared yet because we have seven demands at this point while the supply is six so the auction must move to the fourth round.

Round 4

In this round another 5$ increment is applied to the price which brings the auction price to 35$, the result of bidders is summarized in the below table:

Price	A	B	C	D
35$	1	1	2	2

Figure 5-8: Round 4

As the table shows bidder B losses his two time units bidding because it was 35$, so at this point all the other vendors get their time units because the demand matches the supply which means the market is cleared and no further auction rounds is needed.

5.3 Auction Protocol Implementation

Figure 5-9 describes the interaction between the operator and the vendor under the proposed auction protocol. Once a new advertising window is ready for sale, the operator notifies the registered vendors by sending the start-of-auction message. The operator then announces the first round unit price (also called reserve unit price). After receiving the announced price, vendors compute the number of units they demand based on their individual promotion valuations and response with the quantities demanded. Based on the quantities returned from vendors, The operator computes the screen units credited to each of the vendors at the current price and verifies whether the supply and demand are balanced. If not, the operator will increase the unit price and notifies vendors with the updated price and the number of credited units. Once the market is cleared at a certain round, the operator will terminate the auction by sending the final schedule and payments messages to vendors.

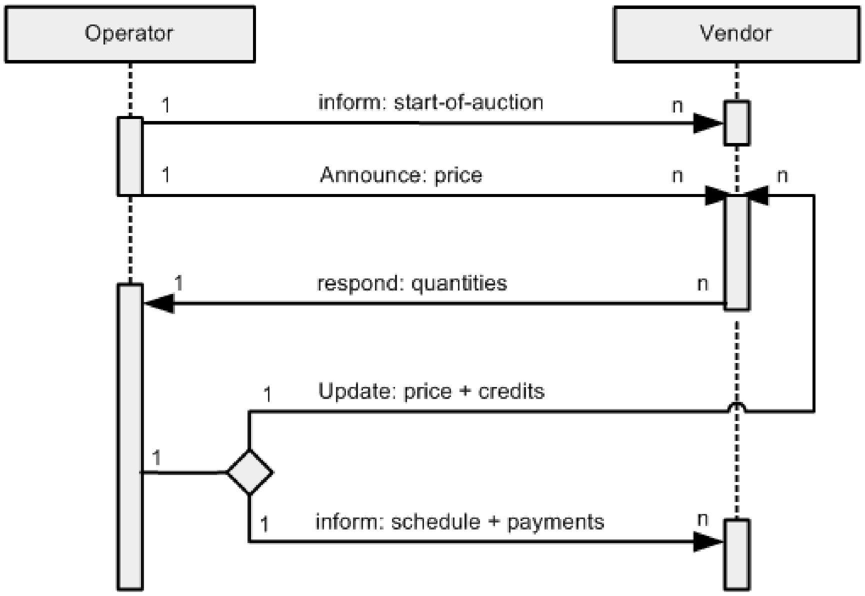


Figure 5-9: Auction Protocol Implementation

Chapter 6
Implementation and Verification

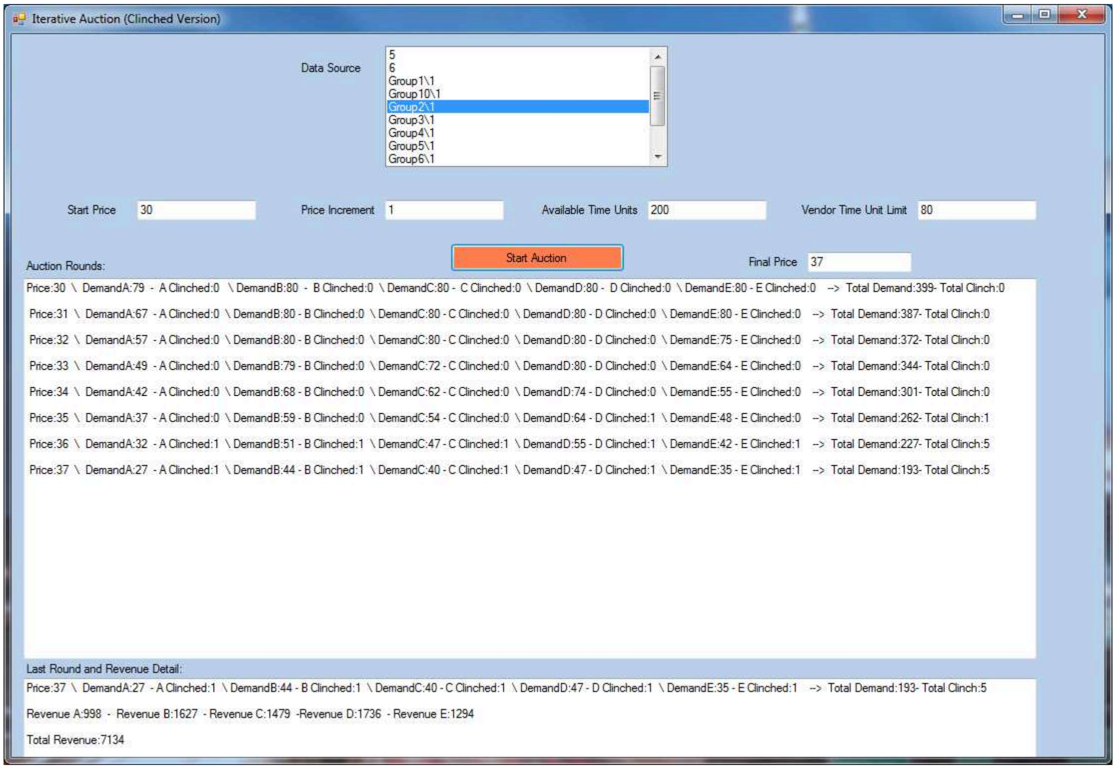
In the PSP formulation presented previously, the number of time units that can be assigned to a vendor is restricted by their vendor limits. The vendor limit indicates the level of customization that vendors can possibly achieve in DSN promotion scheduling. A higher limit provides vendors with the flexibility to adjust their advertising exposure doses in a larger range. Vendors, of cause, do not have to purchase time units to the limit. However, a lower limit puts a restriction on vendors' customization capability. This section evaluates the revenue performance of the iterative bidding through a computational study.

6.1 System Implementation

We coded the iterative bidding model using Microsoft Visual Basic. The prototype consists of a data generator which randomly generates problem data, a database which stores the generated problem data, and an iterative bidding module equipped with both clinched and uniform payment scheme. A user interface for the bidding module is also implemented. Figure 6-1 is a screen shot of the user interface. All experiments were conducted on a PC with a 2.4 GHz CPU. For any of the instances, the solving time of iterative bidding is less than 60 seconds. Some sample solving times can be seen in the table 6-1. It seems that computation time is not a main constraint for the benchmark distributions used in our computational study.

Payment Model	Group	Start Price	Price increment	Available time units	Vendor limit	Solving Time
Uniform	1	10$	0.1	200	80	14 seconds
Uniform	1	10$	0.1	200	200	17 seconds
Clinched	1	10$	0.1	200	80	21 seconds
Clinched	1	10$	0.1	200	200	22 seconds
Uniform	10	10$	0.1	200	80	22 seconds
Uniform	10	10$	0.1	200	200	58 seconds
Clinched	10	10$	0.1	200	80	37 seconds
Clinched	10	10$	0.1	200	200	59 seconds

Table 6-1: Solving times



Data Source: 5, 6, Group1\1, Group10\1, Group2\1 (selected), Group3\1, Group4\1, Group5\1, Group6\1

Start Price: 30

Price Increment: 1

Available Time Units: 200

Vendor Time Unit Limit: 80

Final Price: 37

Auction Rounds:

```
Price:30 \ DemandA:79 - A Clinched:0 \ DemandB:80 - B Clinched:0 \ DemandC:80 - C Clinched:0 \ DemandD:80 - D Clinched:0 \ DemandE:80 - E Clinched:0 --> Total Demand:399- Total Clinch:0
Price:31 \ DemandA:67 - A Clinched:0 \ DemandB:80 - B Clinched:0 \ DemandC:80 - C Clinched:0 \ DemandD:80 - D Clinched:0 \ DemandE:80 - E Clinched:0 --> Total Demand:387- Total Clinch:0
Price:32 \ DemandA:57 - A Clinched:0 \ DemandB:80 - B Clinched:0 \ DemandC:80 - C Clinched:0 \ DemandD:80 - D Clinched:0 \ DemandE:75 - E Clinched:0 --> Total Demand:372- Total Clinch:0
Price:33 \ DemandA:49 - A Clinched:0 \ DemandB:79 - B Clinched:0 \ DemandC:72 - C Clinched:0 \ DemandD:80 - D Clinched:0 \ DemandE:64 - E Clinched:0 --> Total Demand:344- Total Clinch:0
Price:34 \ DemandA:42 - A Clinched:0 \ DemandB:68 - B Clinched:0 \ DemandC:62 - C Clinched:0 \ DemandD:74 - D Clinched:0 \ DemandE:55 - E Clinched:0 --> Total Demand:301- Total Clinch:0
Price:35 \ DemandA:37 - A Clinched:0 \ DemandB:59 - B Clinched:0 \ DemandC:54 - C Clinched:0 \ DemandD:64 - D Clinched:1 \ DemandE:48 - E Clinched:0 --> Total Demand:262- Total Clinch:1
Price:36 \ DemandA:32 - A Clinched:1 \ DemandB:51 - B Clinched:1 \ DemandC:47 - C Clinched:1 \ DemandD:55 - D Clinched:1 \ DemandE:42 - E Clinched:1 --> Total Demand:227- Total Clinch:5
Price:37 \ DemandA:27 - A Clinched:1 \ DemandB:44 - B Clinched:1 \ DemandC:40 - C Clinched:1 \ DemandD:47 - D Clinched:1 \ DemandE:35 - E Clinched:1 --> Total Demand:193- Total Clinch:5
```

Last Round and Revenue Detail:

```
Price:37 \ DemandA:27 - A Clinched:1 \ DemandB:44 - B Clinched:1 \ DemandC:40 - C Clinched:1 \ DemandD:47 - D Clinched:1 \ DemandE:35 - E Clinched:1 --> Total Demand:193- Total Clinch:5
Revenue A:998 - Revenue B:1627 - Revenue C:1479 -Revenue D:1736 - Revenue E:1294
Total Revenue:7134
```

Figure 6-1: Auction Form Screen Shot

6.2 Design of the testing problems

We assume a relatively smaller VMI setting which consists of 5 vendors and 200 available time units for scheduling within an advertising time window. Vendors in an instance are distinguished by their advertising response functions. For each vendor, their advertising response function is characterized by the value of *alpha* and *beta*. As mentioned previously, alpha indicates the effectiveness of the content and beta indicates the sensitivity of the product to advertising. Alphas are randomly drawn from a uniform distribution $U(\bar{\alpha} - \frac{rang(\alpha)}{2}, \bar{\alpha} + \frac{rang(\alpha)}{2})$, where $U(a, b)$ represents a uniform probability distribution between a and b; $\bar{\alpha}$is the predetermined mean of alpha and $rang(\alpha)$ is the range of the distribution for α. Similarly, betas are randomly drawn from a uniform distribution $U(\bar{\beta} - \frac{rang(\beta)}{2}, \bar{\beta} + \frac{rang(\beta)}{2})$, where $\bar{\beta}$is the predetermined mean of beta and $rang(\beta)$ is the range of the distribution for β. The range of *alpha* controls the distribution of advertisements' effectiveness. The range of *beta* controls the distribution of the products' sensitivity to advertising. The larger the ranges, the more diverse the vendors in terms of their advertising response functions. We generated 10 groups of problem instances with increasing level of diversities. For these groups, we fixed $\bar{\alpha}$ to 0.8 which is higher than the data presented in (Jones J. P., 2006). This is justifiable since DSN has higher ad recall rate than that of traditional media (PRN, 2012). We also fixed $\bar{\beta}$ to 1.0, which is consistent with the data presented in (Jones J. P., 2006). μ_0 is assumed to be 100 for all vendors. Promotion values are equal to the sales uplift. Details of problem configurations are summarized in Table 6-2.

Group	Mean of alpha	Alpha Range	Mean of beta	Beta Range
1	0.8	0.02	1.0	0.10
2	0.8	0.04	1.0	0.20
3	0.8	0.06	1.0	0.30
4	0.8	0.08	1.0	0.40
5	0.8	0.10	1.0	0.50
6	0.8	0.12	1.0	0.60
7	0.8	0.14	1.0	0.70
8	0.8	0.16	1.0	0.80
9	0.8	0.18	1.0	0.90
10	0.8	0.20	1.0	1.00

Table 6-2: Configuration of problem instances: Group 1-10

An important objective of the proposed DSN promotion scheduling approach is to provide vendors with customized promotion schedules. To evaluate the customization performance of the proposed approach, problem instances with various levels of diversity is necessary since if vendors' requirements are identical or very similar, customization will not have significant impact on systems' performance. In the following subsection, we evaluate the performance of the iterative bidding model under various vendor limits.

6.3 Solution value under various vendor limits

In the case of the testing data designed for this thesis, we have 200 time units during a promotion window. If we set vendor limit to 40, there will be no competition between vendors. The iterative bidding model will allocate a fixed number of 40 time units to each vendor. If we go below 40, there will be time units left unused. Therefore, 40 is the minimal vendor limit the operator wants to assign. We solve the 10 groups of instances using the iterative model. Table 6-3 shows the solution value performance for all 10 groups of problems under 4 vendor limits: VL=50, VL=100, VL=150, VL=200. It is observed that for all 10 groups, a higher vendor limit, or in other words, higher customization level results in higher solution values, especially for high-diversity groups. For example, Group 10's average promotion value increases from 4264.8 to 10354.8 when the vendor limit increases from 50 to 200. This is because vendors in high-diversity groups have quite different promotion values on an exposure dose and, in this case, a customization procedure helps allocate the time units to the vendors who need them the most, thus improve overall value of solutions.

	VL=50	VL=100	VL=150	VL=200
G1	7172.2	7217	7217	7217
G2	6778	7065	7065	7065
G3	6188.2	7218	7218	7218
G4	5548.2	6735	6735	6735
G5	3639.4	6684.1	6684.1	6684.1
G6	5272.3	7319.4	8316.4	8543.7
G7	5663.9	9876.4	10087.2	10087.2
G8	5432.6	7476	8247.8	8717.2
G9	7249	9050	9690.2	10126
G10	4264.4	7907.5	9182	10354.8

Table 6-3: Promotion value performance

6.4 Revenue performance of iterative bidding

In the iterative bidding model, the clinched payment scheme motivates vendors to truthfully report their demand along the bidding process. However, these benefits are obtained with a cost of losing revenue. Under the clinched payment scheme, many of the credits occurred at earlier prices, therefore, vendors' payments do not generally equal their final demands evaluated at the final prices. In the following we evaluate the revenue performance of the iterative bidding under clinched payments through experiments.

Groups	VL=40%	VL=100%
G1	96.40%	96.40%
G2	96.30%	96.30%
G3	95.10%	95.90%
G4	94.80%	96.00%
G5	93.70%	95.30%
G6	89.80%	94.90%
G7	91.30%	93.60%
G8	84.40%	94.70%
G9	90.50%	91.30%
G10	88.40%	77.40%

Table 6-4: Ratio of Revenue

We use the same problem groups as we used in the vendor limit testing. We tested the revenue performance of clinched payment scheme in comparison with uniform payment scheme under two vendor limits: VL=80 (40%) and VL=200 (100%). Table 6-4 shows the ratio of the revenue under clinched payments to the revenue under uniform payments on all 10 groups of problem instances. It is observed that, for the low-diversity problem groups, the revenue performance of clinched payment is pretty good (above 90%) under both VL=40% and VL=100%. For high-diversity problems, the revenue performance of clinched payments can decrease up to 20% compared to that of the low-diversity problems. This is because vendors' values vary to a larger extent in high-diversity problems, which results in a higher chance for allocations to be clinched at earlier prices, thus depress the revenue. The revenue decrease observed here is the cost needed to prevent vendors from gamming the system.

Chapter 7
Conclusions

In today's increasingly virtual and volatile market, DSN promotion scheduling has to response to vendors' changing requirements in a timely manner and, at the same time, balance the customization requirements across all vendors. We propose automated scheduling models which concurrently customize vendors' promotion schedules and optimize overall system performance. The models are designed based on the direct revelation mechanism and iterative bidding with uniform and clinched payment schemes, which make them suitable for both competitive and game theoretic settings. In addition, the level of customization can be adjusted by assigning different vendor purchasing limits. We show that the proposed direct revelation mechanism computes optimal solutions to the DSN promotion scheduling problem. For the iterative bidding model, our experimental results show that the iterative bidding model exhibits good revenue performance.

Our work is ongoing in several areas, namely content customization and quality assessment, advanced advertising response models, and interactive DSN scheduling. Current multimedia technology supports dynamic content configuration. For example, content's language, text message, and background scenes can be configured on the fly. We are investigating how to predict the effectiveness of various content configurations given specific demographics of the audience and how to effectively incorporate this knowledge into promotion schedules. In terms of assessing a vendor's return from the purchased time units, a good adverting response model is critical to high quality estimation. In Section 3, we proposed a simple advertising response function which assumes that vendors are indifferent from the locations of the screens. However, casual observation and the literature (Gutierrez B. P., 2008) reveal that a particular location does not have the same exposure effectiveness for all vendors. While certain locations such as the entrance and the checkout counters can be desirable for most of the vendors, a packaged goods vendor (e.g. P&G) will prefer promoting their shampoo brands on the screens located in the grocery department rather than the electronics department. Since the exposure effectiveness of a location is vendor dependent, we plan to develop more practical advertising response models to reflect the location differences of vendors. Approaches based on data mining techniques will be used to build the correlation between the exposure doses at various locations and the resulted sales uplift. Finally, techniques that collect audience feedback will also be integrated into the automated promotion scheduling system. Along this direction, we will investigate the effectiveness assessment models that incorporate the audience attention information captured by the on-screen cameras based on our previous experiments on eye-tracking systems (Jin, Zeng, & Wang, 2010). In all these research activities, insights derived from extensive simulation and on-site experiments will play an essential role in guiding system design and implementation.

Appendix I

Iterative Auction Form (Clinched Win Version)

```
Imports System.Data.OleDb
Imports System.IO
Imports System.Data.Sql
Imports System.Data.SqlClient

Public Class Form1

    Private Sub BtnSt_Click(sender As System.Object, e As System.EventArgs)
Handles BtnSt.Click


        TextBox1.Clear()   ' Clears all the text boxes
        TxtFP.Clear()
        TxtRev.Clear()



        Dim S As Decimal = TxtS.Text    ' Gets the values of the text files in the
form
        Dim PI As Decimal = TxtPI.Text
        Dim Time As Integer = TxtT.Text
        Dim VTLA As Integer = TxtVTL.Text
        Dim VTLB As Integer = TxtVTL.Text
        Dim VTLC As Integer = TxtVTL.Text
        Dim VTLD As Integer = TxtVTL.Text
        Dim VTLE As Integer = TxtVTL.Text
        Dim Time2 As Integer
        'Gets the selected table name as String
        Dim GroupN As String = LstData.SelectedItem

        'Dim AR As String = TextBox1.Text
        Dim DACount As Integer = 0
        Dim DBCount As Integer = 0
        Dim DCCount As Integer = 0
        Dim DDCount As Integer = 0
        Dim DECount As Integer = 0
        Dim ClinchA As Integer
        Dim ClinchB As Integer
        Dim ClinchC As Integer
        Dim ClinchD As Integer
        Dim ClinchE As Integer
        Dim RevClA As Integer
        Dim RevClB As Integer
        Dim RevClC As Integer
        Dim RevClD As Integer
        Dim RevClE As Integer
        Dim RevClACount As Integer
        Dim RevClBCount As Integer
        Dim RevClCCount As Integer
        Dim RevClDCount As Integer
        Dim RevClECount As Integer



        Dim DT As Integer

        ' --------------------------------  First Auction Round

        ' Makes the connection to the database

        Dim Conn1 As New OleDbConnection("Provider=Microsoft.JET.OLEDB.4.0;" & _
              "Data Source=C:\\Users\\Kourosh\\Documents\\Auction.mdb;")
        Conn1.Open()

        Dim Trans1 As OleDb.OleDbTransaction
        Trans1 = Conn1.BeginTransaction


        Dim CmdA1 As New OleDb.OleDbCommand

        Dim CmdB1 As New OleDb.OleDbCommand

        Dim CmdC1 As New OleDb.OleDbCommand

        Dim CmdD1 As New OleDb.OleDbCommand

        Dim CmdE1 As New OleDb.OleDbCommand


' Produces the query statement for selecting the marginal values higher than the
start price

        CmdA1.CommandText = "SELECT  A FROM  " & GroupN & "  WHERE A > '" & S & "'
"

        CmdB1.CommandText = "SELECT  B FROM  " & GroupN & "  WHERE B > '" & S & "'
"

        CmdC1.CommandText = "SELECT  C FROM  " & GroupN & "  WHERE C > '" & S & "'
"

        CmdD1.CommandText = "SELECT  D FROM  " & GroupN & "  WHERE D > '" & S & "'
"

        CmdE1.CommandText = "SELECT  E FROM  " & GroupN & "  WHERE E > '" & S & "'
"

        CmdA1.Connection = Conn1

        CmdA1.Transaction = Trans1

        CmdB1.Connection = Conn1
        CmdB1.Transaction = Trans1

        CmdC1.Connection = Conn1

        CmdC1.Transaction = Trans1

        CmdD1.Connection = Conn1

        CmdD1.Transaction = Trans1

        CmdE1.Connection = Conn1

        CmdE1.Transaction = Trans1


        Dim MyReaderA1 As OleDb.OleDbDataReader

        MyReaderA1 = CmdA1.ExecuteReader

       ' Reads the data and checks the vendor time unit limit


        Do While MyReaderA1.Read And DACount < VTLA
            Dim DA As String = MyReaderA1.Item("A")

            DACount  =  DACount  +  1        '  Counts  the  amount  of  demand  read  for
vendor A

        Loop

        MyReaderA1.Close()


        Dim MyReaderB1 As OleDb.OleDbDataReader

        MyReaderB1 = CmdB1.ExecuteReader

        ' Reads the data and checks the vendor time unit limit

        Do While MyReaderB1.Read And DBCount < VTLB

      ' Counts the amount of demand read for vendor B

            Dim DB As String = MyReaderB1.Item("B")
            DBCount = DBCount + 1

        Loop

        MyReaderB1.Close()


        Dim MyReaderC1 As OleDb.OleDbDataReader

        MyReaderC1 = CmdC1.ExecuteReader
        Do While MyReaderC1.Read And DCCount < VTLC

            Dim DC As String = MyReaderC1.Item("C")
            DCCount = DCCount + 1

        Loop

        MyReaderC1.Close()


        Dim MyReaderD1 As OleDb.OleDbDataReader

        MyReaderD1 = CmdD1.ExecuteReader

        Do While MyReaderD1.Read And DDCount < VTLD

            Dim DD As String = MyReaderD1.Item("D")
            DDCount = DDCount + 1

        Loop

        MyReaderD1.Close()


        Dim MyReaderE1 As OleDb.OleDbDataReader

        MyReaderE1 = CmdE1.ExecuteReader

        Do While MyReaderE1.Read And DECount < VTLE

            Dim DE As String = MyReaderE1.Item("E")
            DECount = DECount + 1

        Loop

        MyReaderE1.Close()


        '-----------------------------------  Clinch Check for different Vendors


        If DACount > 1 And (DBCount + DCCount + DDCount + DECount) < Time Then
' Checks if A has any clinched wins

            ClinchA = 1
            RevClA = RevClA + (ClinchA * S)     ' If there is a clinched win the
revenue is calculated at that price
            RevClACount = RevClACount + 1         ' Add one to the amount of the
clinched revenue count


        Else
            ClinchA = ClinchA       ' If there is no clinched win the amount is
equal to the former one

        End If
        TextBox1.Text = TextBox1.Text & "Price:" & S & "  \  DemandA:" & DACount &
"  -  A Clinched:" & ClinchA



        If DBCount > 1 And (DACount + DCCount + DDCount + DECount) < Time Then  '
Checks if B has any clinched wins

            ClinchB = 1
            RevClB = RevClB + (ClinchB * S)      ' If there is a clinched win the
revenue is calculated at that price
            RevClBCount = RevClBCount + 1

        Else
            ClinchB = ClinchB

        End If

        TextBox1.Text = TextBox1.Text & "    \ DemandB:" & DBCount & "   -   B
Clinched:" & ClinchB


        If DCCount > 1 And (DACount + DBCount + DDCount + DECount) < Time Then

            ClinchC = 1
            RevClC = RevClC + (ClinchC * S)
            RevClCCount = RevClCCount + 1


        Else
            ClinchC = ClinchC

        End If

        TextBox1.Text = TextBox1.Text & "   \ DemandC:" & DCCount & " -   C
Clinched:" & ClinchC


        If DDCount > 1 And (DACount + DBCount + DCCount + DECount) < Time Then

            ClinchD = 1
            RevClD = RevClD + (ClinchD * S)
            RevClDCount = RevClDCount + 1


        Else
            ClinchD = ClinchD

        End If

        TextBox1.Text = TextBox1.Text & "   \ DemandD:" & DDCount & " -   D
Clinched:" & ClinchD


        If DECount > 1 And (DACount + DBCount + DCCount + DDCount) < Time Then

            ClinchE = 1
            RevClE = RevClE + (ClinchE * S)
            RevClECount = RevClECount + 1

        Else
            ClinchE = ClinchE

        End If

' Checks the amount of total demand

        DT = (DACount + DBCount + DCCount + DDCount + DECount)
        TextBox1.Text = TextBox1.Text & "   \ DemandE:" & DECount & " - E
Clinched:" & ClinchE & "    --> " & " Total Demand:" & DT & "- Total Clinch:" &
(ClinchA + ClinchB + ClinchC + ClinchD + ClinchE) & vbNewLine & vbNewLine
        Time2 = Time - (ClinchA + ClinchB + ClinchC + ClinchD + ClinchE)   '
updates the available time limit


        Conn1.Close()


        '--------------------------------------------   Loop for Auction rounds


       ' Database Connection

        Dim Conn As New OleDbConnection("Provider=Microsoft.JET.OLEDB.4.0;" & _
       "Data Source=C:\\Users\\Kourosh\\Documents\\Auction.mdb;")
        Conn.Open()


       ' Stays in the loop till the amount of demands equals the available time
units

        Do While DT > Time2

            S = S + PI    ' Adds the price increment to the starting price




            Dim CmdA As New OleDb.OleDbCommand

            Dim CmdB As New OleDb.OleDbCommand

            Dim CmdC As New OleDb.OleDbCommand

            Dim CmdD As New OleDb.OleDbCommand

            Dim CmdE As New OleDb.OleDbCommand


            CmdA.CommandText = "SELECT  A FROM  " & GroupN & "  WHERE A > '" & S &
"'  "   ' Produces the query statement for selecting the marginal values higher
than the current price

            CmdB.CommandText = "SELECT  B FROM  " & GroupN & "  WHERE B > '" & S &
"'  "   ' Produces the query statement for selecting the marginal values higher
than the current price
```

```
            CmdC.CommandText = "SELECT  C FROM  " & GroupN & "  WHERE C > '" & S &
"'  "

            CmdD.CommandText = "SELECT  D FROM  " & GroupN & "  WHERE D > '" & S &
"'  "

            CmdE.CommandText = "SELECT  E FROM  " & GroupN & "  WHERE E > '" & S &
"'  "

            CmdA.Connection = Conn

            'CmdA.Transaction = Trans

            CmdB.Connection = Conn

            'CmdB.Transaction = Trans

            CmdC.Connection = Conn

            'CmdC.Transaction = Trans

            CmdD.Connection = Conn

            'CmdD.Transaction = Trans

            CmdE.Connection = Conn

            ' CmdE.Transaction = Trans

            DACount = 0 - ClinchA   ' The amount of clinched win of the vendor is
subtracted
            DBCount = 0 - ClinchB
            DCCount = 0 - ClinchC
            DDCount = 0 - ClinchD
            DECount = 0 - ClinchE

            Dim MyReaderA As OleDb.OleDbDataReader

            MyReaderA = CmdA.ExecuteReader

            Do While MyReaderA.Read And DACount < VTLA  ' Checks the vendor time
unit limit during the reading process

                Dim DA As String = MyReaderA.Item("A")

                DACount = DACount + 1    ' Counts the amount of demand read for
vendor A

            Loop

            MyReaderA.Close()


            Dim MyReaderB As OleDb.OleDbDataReader

            MyReaderB = CmdB.ExecuteReader
            Do While MyReaderB.Read And DBCount < VTLB   ' Checks the vendor time
unit limit during the reading process

                Dim DB As String = MyReaderB.Item("B")
                DBCount = DBCount + 1     ' Counts the amount of demand read for
vendor B

            Loop

            MyReaderB.Close()


            Dim MyReaderC As OleDb.OleDbDataReader

            MyReaderC = CmdC.ExecuteReader

            Do While MyReaderC.Read And DCCount < VTLC

                Dim DC As String = MyReaderC.Item("C")
                DCCount = DCCount + 1
                '
            Loop

            MyReaderC.Close()


            Dim MyReaderD As OleDb.OleDbDataReader

            MyReaderD = CmdD.ExecuteReader

            Do While MyReaderD.Read And DDCount < VTLD

                Dim DD As String = MyReaderD.Item("D")
                DDCount = DDCount + 1

            Loop

            MyReaderD.Close()


            Dim MyReaderE As OleDb.OleDbDataReader

            MyReaderE = CmdE.ExecuteReader

            Do While MyReaderE.Read And DECount < VTLE

                Dim DE As String = MyReaderE.Item("E")
                DECount = DECount + 1

            Loop

            MyReaderE.Close()


            '--------------------------------   Clinch  Check  for  Auction  Rounds
Loop

Checks if A has any clinched wins
```

```
            If DACount > 1 And (DDCount + DBCount + DCCount + DECount) < Time Then
'

                ClinchA = 1
                RevClA = RevClA + (ClinchA * S)     ' If there is a clinched win
the revenue is calculated at that price
                RevClACount = RevClACount + 1       ' Add one to the amount of the
clinched revenue count

            Else
                ClinchA = ClinchA

            End If

            TextBox1.Text = TextBox1.Text & " Price:" & S & "   \   DemandA:" &
DACount & "  - A Clinched:" & ClinchA


            If DBCount > 1 And (DDCount + DACount + DCCount + DECount) < Time Then
' Checks if B has any clinched wins

                ClinchB = 1
                RevClB = RevClB + (ClinchB * S)   ' If there is a clinched win the
revenue is calculated at that price
                RevClBCount = RevClBCount + 1      ' Add one to the amount of the
clinched revenue count

            Else
                ClinchB = ClinchB

            End If

            TextBox1.Text = TextBox1.Text & "   \ DemandB:" & DBCount & " - B
Clinched:" & ClinchB


            If DCCount > 1 And (DACount + DBCount + DDCount + DECount) < Time Then

                ClinchC = 1
                RevClC = RevClC + (ClinchC * S)
                RevClCCount = RevClCCount + 1

            Else
                ClinchC = ClinchC

            End If

            TextBox1.Text = TextBox1.Text & "   \ DemandC:" & DCCount & " - C
Clinched:" & ClinchC


            If DDCount > 1 And (DACount + DBCount + DCCount + DECount) < Time Then

                ClinchD = 1
                RevClD = RevClD + (ClinchD * S)
                RevClDCount = RevClDCount + 1
            Else
                ClinchD = ClinchD

            End If

            TextBox1.Text = TextBox1.Text & "   \ DemandD:" & DDCount & " - D
Clinched:" & ClinchD


            If DECount > 1 And (DACount + DBCount + DDCount + DCCount) < Time Then

                ClinchE = 1
                RevClE = RevClE + (ClinchE * S)
                RevClECount = RevClECount + 1

            Else
                ClinchE = ClinchE

            End If
            ' Checks the amount of total demand
            DT = (DACount + DBCount + DCCount + DDCount + DECount)
            TextBox1.Text = TextBox1.Text & "   \ DemandE:" & DECount & " - E
Clinched:" & ClinchE & "     --> " & " Total Demand:" & DT & "- Total Clinch:" &
(ClinchA + ClinchB + ClinchC + ClinchD + ClinchE) & vbNewLine & vbNewLine
            Time2 = Time - (ClinchA + ClinchB + ClinchC + ClinchD + ClinchE)  '
updates the available time limit


        Loop

        ' Shows the final round result and aslo the revenue gained from each
vendor

        TxtRev.Text = "Price:" & S & "  \  DemandA:" & DACount & "  - A Clinched:"
& ClinchA & "  \ DemandB:" & DBCount & " - B Clinched:" & ClinchB & "  \ DemandC:"
& DCCount & " - C Clinched:" & ClinchC & "   \ DemandD:" & DDCount & " - D
Clinched:" & ClinchD & "  \ DemandE:" & DECount & " - E Clinched:" & ClinchE & "
--> " & " Total Demand:" & DT & "- Total Clinch:" & (ClinchA + ClinchB + ClinchC +
ClinchD + ClinchE) & vbNewLine & vbNewLine
        TxtRev.Text = TxtRev.Text & "Revenue A:" & (((DACount - RevClACount) * S)
+ (RevClA)) & "  -  Revenue B:" & (((DBCount - RevClBCount) * S) + (RevClB)) & "
- Revenue C:" & (((DCCount - RevClCCount) * S) + (RevClC)) & "  -Revenue D:" &
(((DDCount - RevClDCount) * S) + (RevClD)) & "  - Revenue E:" & (((DECount -
RevClECount) * S) + (RevClE)) & vbNewLine & vbNewLine
        TxtRev.Text = TxtRev.Text + "Total Revenue:" & (((DACount - RevClACount) *
S) + (RevClA) + ((DBCount - RevClBCount) * S) + (RevClB) + ((DCCount -
RevClCCount) * S) + (RevClC) + ((DDCount - RevClDCount) * S) + (RevClD) +
((DECount - RevClECount) * S) + (RevClE))

        ' Shows the final price

        TxtFP.Text = S

        Conn.Close()


    End Sub
    Private Sub Form1_Load(sender As System.Object, e As System.EventArgs) Handles
MyBase.Load


        'Variable to hold path to database
        Dim DatabasePath As String = "C:\\Users\\Kourosh\\Documents\\Auction.mdb;"

        'SQL String to connect to database
        Dim ConnString As String = _
            "Provider=Microsoft.Jet.OLEDB.4.0;" & _
            "Data Source=" & DatabasePath & _
            "Persist Security Info=False"



        Dim SchemaTable As DataTable

        'Connect to the database
        Dim conn As New System.Data.OleDb.OleDbConnection(ConnString)

        Try
            LstData.Items.Clear()

            conn.Open()

            'Get table and view names
            SchemaTable =
conn.GetOleDbSchemaTable(System.Data.OleDb.OleDbSchemaGuid.Tables, New Object()
{Nothing, Nothing, Nothing, Nothing})

            Dim int As Integer
            For int = 0 To SchemaTable.Rows.Count - 1
                If SchemaTable.Rows(int)!TABLE_TYPE.ToString = "TABLE" Then
                    'Add items to list box
                    LstData.Items.Add(SchemaTable.Rows(int)!TABLE_NAME.ToString())
                End If
            Next

        Catch ex As Exception
            MessageBox.Show(ex.Message.ToString(), "Data Load Error",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        End Try

        conn.Close()


    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs)

    End Sub

    Private Sub ComboBox1_SelectedIndexChanged_1(sender As System.Object, e As
System.EventArgs)

    End Sub
    Private Sub LstData_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles LstData.SelectedIndexChanged


    End Sub

End Class
```

Appendix II

Iterative Auction Form (Uniform Version)

```
Imports System.Data.OleDb
Imports System.IO
Imports System.Data.Sql
Imports System.Data.SqlClient

Public Class Form1

    Private Sub BtnSt_Click(sender As System.Object, e As System.EventArgs)
Handles BtnSt.Click

        TextBox1.Clear()   ' clears the text boxes
        TxtFP.Clear()
        TxtRev.Clear()

        ' Variable Declaration

        Dim S As Decimal = TxtS.Text  ' Gets the number of text boxes
        Dim PI As Decimal = TxtPI.Text
        Dim Time As Integer = TxtT.Text
        Dim VTLA As Integer = TxtVTL.Text
        Dim VTLB As Integer = TxtVTL.Text
        Dim VTLC As Integer = TxtVTL.Text
        Dim VTLD As Integer = TxtVTL.Text
        Dim VTLE As Integer = TxtVTL.Text

        Dim GroupN As String = LstData.SelectedItem
        'Dim AR As String = TextBox1.Text
        Dim DACount As Integer = 0
        Dim DBCount As Integer = 0
        Dim DCCount As Integer = 0
        Dim DDCount As Integer = 0
        Dim DECount As Integer = 0




        Dim DT As Integer

        ' --------------------------------  First Auction Round

        ' Database Connection

        Dim Conn1 As New OleDbConnection("Provider=Microsoft.JET.OLEDB.4.0;" & _
              "Data Source=C:\\Users\\Kourosh\\Documents\\Auction.mdb;")
        Conn1.Open()
        Dim Trans1 As OleDb.OleDbTransaction
        Trans1 = Conn1.BeginTransaction


        Dim CmdA1 As New OleDb.OleDbCommand

        Dim CmdB1 As New OleDb.OleDbCommand

        Dim CmdC1 As New OleDb.OleDbCommand

        Dim CmdD1 As New OleDb.OleDbCommand

        Dim CmdE1 As New OleDb.OleDbCommand


        ' Produces the query statement for selecting the marginal values higher
than the start price

        CmdA1.CommandText = "SELECT  A FROM  " & GroupN & "  WHERE A > '" & S & "'
"

        CmdB1.CommandText = "SELECT  B FROM  " & GroupN & "  WHERE B > '" & S & "'
"

        CmdC1.CommandText = "SELECT  C FROM  " & GroupN & "  WHERE C > '" & S & "'
"

        CmdD1.CommandText = "SELECT  D FROM  " & GroupN & "  WHERE D > '" & S & "'
"

        CmdE1.CommandText = "SELECT  E FROM  " & GroupN & "  WHERE E > '" & S & "'
"

        CmdA1.Connection = Conn1

        CmdA1.Transaction = Trans1

        CmdB1.Connection = Conn1

        CmdB1.Transaction = Trans1

        CmdC1.Connection = Conn1

        CmdC1.Transaction = Trans1

        CmdD1.Connection = Conn1

        CmdD1.Transaction = Trans1

        CmdE1.Connection = Conn1

        CmdE1.Transaction = Trans1



        ' Reads the data and checks the vendor time unit limit
        Dim MyReaderA1 As OleDb.OleDbDataReader

        MyReaderA1 = CmdA1.ExecuteReader

        Do While MyReaderA1.Read And DACount < VTLA

            Dim DA As String = MyReaderA1.Item("A")

            DACount = DACount + 1  ' Counts the amount of demand read for vendor A

        Loop

        MyReaderA1.Close()


        ' Reads the data and checks the vendor time unit limit

        Dim MyReaderB1 As OleDb.OleDbDataReader

        MyReaderB1 = CmdB1.ExecuteReader

        Do While MyReaderB1.Read And DBCount < VTLB

            Dim DB As String = MyReaderB1.Item("B")  ' Counts the amount of demand
read for vendor A
            DBCount = DBCount + 1

        Loop

        MyReaderB1.Close()


        Dim MyReaderC1 As OleDb.OleDbDataReader

        MyReaderC1 = CmdC1.ExecuteReader

        Do While MyReaderC1.Read And DCCount < VTLC

            Dim DC As String = MyReaderC1.Item("C")
            DCCount = DCCount + 1
            'AR = AR & " , " & DC
            'DataGridView1.Rows.Add(V1)
        Loop

        MyReaderC1.Close()


        Dim MyReaderD1 As OleDb.OleDbDataReader

        MyReaderD1 = CmdD1.ExecuteReader

        Do While MyReaderD1.Read And DDCount < VTLD

            Dim DD As String = MyReaderD1.Item("D")
            DDCount = DDCount + 1
            'AR = AR & " , " & DD

            'DataGridView1.Rows.Add(V1)
        Loop

        MyReaderD1.Close()


        Dim MyReaderE1 As OleDb.OleDbDataReader

        MyReaderE1 = CmdE1.ExecuteReader

        Do While MyReaderE1.Read And DECount < VTLE

            Dim DE As String = MyReaderE1.Item("E")
            DECount = DECount + 1
            'AR = AR & " , " & DE
            'DataGridView1.Rows.Add(V1)
        Loop

        MyReaderE1.Close()



        TextBox1.Text = TextBox1.Text & "Price:" & S & "  \  DemandA:" & DACount


        TextBox1.Text = TextBox1.Text & "   \ DemandB:" & DBCount


        TextBox1.Text = TextBox1.Text & "  \ DemandC:" & DCCount

        TextBox1.Text = TextBox1.Text & "  \ DemandD:" & DDCount



        DT = (DACount + DBCount + DCCount + DDCount + DECount)

        TextBox1.Text = TextBox1.Text & "  \ DemandE:" & DECount & "    --> " & "
Total Demand:" & DT & vbNewLine & vbNewLine



        Conn1.Close()


        '----------------------------------------   Loop for Auction rounds

        ' Database Connection

        Dim Conn As New OleDbConnection("Provider=Microsoft.JET.OLEDB.4.0;" & _
       "Data Source=C:\\Users\\Kourosh\\Documents\\Auction.mdb;")
        Conn.Open()


        ' Stays in the loop till the amount of demands equals the available time
units

        Do While DT > Time

            S = S + PI   ' Adds the price increment to the starting price
            Dim CmdA As New OleDb.OleDbCommand

            Dim CmdB As New OleDb.OleDbCommand

            Dim CmdC As New OleDb.OleDbCommand

            Dim CmdD As New OleDb.OleDbCommand

            Dim CmdE As New OleDb.OleDbCommand

            '  Produces  the  query  statement  for  selecting  the  marginal  values
higher than the current price

            CmdA.CommandText = "SELECT  A FROM  " & GroupN & "  WHERE A > '" & S &
"'  "

            CmdB.CommandText = "SELECT  B FROM  " & GroupN & "  WHERE B > '" & S &
"'  "

            CmdC.CommandText = "SELECT  C FROM  " & GroupN & "  WHERE C > '" & S &
"'  "

            CmdD.CommandText = "SELECT  D FROM  " & GroupN & "  WHERE D > '" & S &
"'  "

            CmdE.CommandText = "SELECT  E FROM  " & GroupN & "  WHERE E > '" & S &
"'  "

            CmdA.Connection = Conn

            'CmdA.Transaction = Trans

            CmdB.Connection = Conn

            'CmdB.Transaction = Trans

            CmdC.Connection = Conn

            'CmdC.Transaction = Trans

            CmdD.Connection = Conn

            'CmdD.Transaction = Trans

            CmdE.Connection = Conn

            ' CmdE.Transaction = Trans

            DACount = 0
            DBCount = 0
            DCCount = 0
            DDCount = 0
            DECount = 0
            ' Checks the vendor time unit limit during the reading process

            Dim MyReaderA As OleDb.OleDbDataReader

            MyReaderA = CmdA.ExecuteReader

            Do While MyReaderA.Read And DACount < VTLA

                Dim DA As String = MyReaderA.Item("A")

                DACount  =  DACount  +  1   '  Counts  the  amount  of  demand  read  for
vendor A

            Loop

            MyReaderA.Close()


            Dim MyReaderB As OleDb.OleDbDataReader

            MyReaderB = CmdB.ExecuteReader

            Do While MyReaderB.Read And DBCount < VTLB

                Dim DB As String = MyReaderB.Item("B")
                DBCount  =  DBCount  +  1    '  Counts  the  amount  of  demand  read  for
vendor B

            Loop

            MyReaderB.Close()


            Dim MyReaderC As OleDb.OleDbDataReader

            MyReaderC = CmdC.ExecuteReader

            Do While MyReaderC.Read And DCCount < VTLC

                Dim DC As String = MyReaderC.Item("C")
                DCCount = DCCount + 1

            Loop

            MyReaderC.Close()


            Dim MyReaderD As OleDb.OleDbDataReader

            MyReaderD = CmdD.ExecuteReader

            Do While MyReaderD.Read And DDCount < VTLD

                Dim DD As String = MyReaderD.Item("D")
                DDCount = DDCount + 1

            Loop
            MyReaderD.Close()


            Dim MyReaderE As OleDb.OleDbDataReader

            MyReaderE = CmdE.ExecuteReader

            Do While MyReaderE.Read And DECount < VTLE

                Dim DE As String = MyReaderE.Item("E")
                DECount = DECount + 1

            Loop

            MyReaderE.Close()



            TextBox1.Text = TextBox1.Text & " Price:" & S & "  \  DemandA:" &
DACount

            TextBox1.Text = TextBox1.Text & "  \ DemandB:" & DBCount

            TextBox1.Text = TextBox1.Text & "  \ DemandC:" & DCCount

            TextBox1.Text = TextBox1.Text & "  \ DemandD:" & DDCount



            TextBox1.Text = TextBox1.Text & "  \ DemandE:" & DECount & "    --> "
& " Total Demand:" & DT & vbNewLine & vbNewLine

            ' Checks the amount of total demand

            DT = (DACount + DBCount + DCCount + DDCount + DECount)

        Loop

        ' Shows the final round result and aslo the revenue gained from each
vendor

        TxtRev.Text = "Price:" & S & "  \  DemandA:" & DACount & "  \ DemandB:" &
DBCount & "  \ DemandC:" & DCCount & "  \ DemandD:" & DDCount & "  \ DemandE:" &
DECount & "    --> " & " Total Demand:" & DT & vbNewLine & vbNewLine
        TxtRev.Text = TxtRev.Text & "Revenue A:" & (DACount * S) & "  - Revenue
B:" & (DBCount * S) & "  - Revenue C:" & (DCCount * S) & "  -Revenue D:" &
(DDCount * S) & "  - Revenue E:" & (DECount * S) & vbNewLine & vbNewLine
        TxtRev.Text = TxtRev.Text + "Total Revenue:" & (DACount * S) + (DBCount *
S) + (DCCount * S) + (DDCount * S) + (DECount * S)

        'Shows the final price

        TxtFP.Text = S

        Conn.Close()
```

```
    End Sub

    Private Sub LstData_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles LstData.SelectedIndexChanged

    End Sub


    Private Sub Form1_Load(sender As System.Object, e As System.EventArgs) Handles
MyBase.Load


        'Variable to hold path to database
        Dim DatabasePath As String = "C:\\Users\\Kourosh\\Documents\\Auction.mdb;"

        'SQL String to connect to database
        Dim ConnString As String = _
            "Provider=Microsoft.Jet.OLEDB.4.0;" & _
            "Data Source=" & DatabasePath & _
            "Persist Security Info=False"



        Dim SchemaTable As DataTable

        'Connect to the database
        Dim conn As New System.Data.OleDb.OleDbConnection(ConnString)

        Try
            LstData.Items.Clear()

            conn.Open()

            'Get table and view names
            SchemaTable =
conn.GetOleDbSchemaTable(System.Data.OleDb.OleDbSchemaGuid.Tables, New Object()
{Nothing, Nothing, Nothing, Nothing})

            Dim int As Integer
            For int = 0 To SchemaTable.Rows.Count - 1
                If SchemaTable.Rows(int)!TABLE_TYPE.ToString = "TABLE" Then
                    'Add items to list box
                    LstData.Items.Add(SchemaTable.Rows(int)!TABLE_NAME.ToString())
                End If
            Next

        Catch ex As Exception
            MessageBox.Show(ex.Message.ToString(), "Data Load Error",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        End Try

        conn.Close()


    End Sub
    Private Sub TxtPI_TextChanged(sender As System.Object, e As System.EventArgs)
Handles TxtPI.TextChanged

    End Sub
End Class
```

Appendix III

Data Generator Form

```
Imports System.Data.OleDb
Imports System.IO
Imports System.Math

Public Class Form1

    Dim array() As String = {"A", "B", "C", "D", "E"}
    Dim array2() As String

    Private Sub Form1_Load(sender As System.Object, e As System.EventArgs) Handles
MyBase.Load

    End Sub


    Private Sub BtnGD_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnGD.Click



        Dim i As Integer
        Dim z As Integer
        z = Integer.Parse(TxtV.Text) - 1     ' Gets the amount of vendors as an
array for the table creation
        ReDim array2(z)
        For i = 0 To z
            array2(i) = array(i).ToString
        Next
        TextBox2.Clear()
        For i = 0 To array2.Length - 1
            TextBox2.Text = TextBox2.Text + array2(i) + " varchar(12),"  ' Creates
the table with the amount of vendor columns specified by the user
        Next
        TextBox2.Text = Mid(TextBox2.Text, 1, TextBox2.TextLength - 1)
        TextBox2.Text = "CREATE TABLE Group10\3(" + TextBox2.Text + ")"  ' Creates
the table for the group 10

        Dim strCreate As String = TextBox2.Text

        Dim conDatabase As OleDbConnection = New
OleDbConnection("Provider=Microsoft.JET.OLEDB.4.0;" & _
           "Data               Source=C:\\Users\\Kourosh\\Documents\\Auction.mdb;")
'  Connection to the database


        Dim cmdDatabase As OleDbCommand = New OleDbCommand(strCreate, conDatabase)

        conDatabase.Open()

        cmdDatabase.ExecuteNonQuery()
        conDatabase.Close()


        Dim SqlString As String = "Insert Into Group10\3 (A, B, C, D, E) Values
(?,?,?,?,?) "      ' Sql command for inserting the values to the specified columns
        Dim         conn         As         OleDbConnection         =         New
OleDbConnection("Provider=Microsoft.JET.OLEDB.4.0;" & _
           "Data Source=C:\\Users\\Kourosh\\Documents\\Auction.mdb;")
        Dim T As Integer
        Dim TL As Integer
        T = 0
        TL = TxtTL.Text
        Dim AlphaMean As Decimal = (0.8)
        Dim AlphaRange As Decimal = (0.2)
        Dim V As Integer = TxtV.Text

        Dim BetaMean As Decimal = (1.0)
        Dim BetaRange As Decimal = (1.0)
        Dim MeanSale As String = 100
        Dim SlotOrder As Integer = 1
        Dim Half As Decimal = 0.5

        Dim RndNum As Random
        Dim Alpha1A As New Decimal
        Dim Beta1A As New Decimal
        Dim AlphaA As New Decimal
        Dim BetaA As New Decimal
        Dim Alpha1B As New Decimal
        Dim Beta1B As New Decimal
        Dim AlphaB As New Decimal
        Dim BetaB As New Decimal
        Dim Alpha1C As New Decimal
        Dim Beta1C As New Decimal
        Dim AlphaC As New Decimal
        Dim BetaC As New Decimal
        Dim Alpha1D As New Decimal
        Dim Beta1D As New Decimal
        Dim AlphaD As New Decimal
        Dim BetaD As New Decimal
        Dim Alpha1E As New Decimal
        Dim Beta1E As New Decimal
        Dim AlphaE As New Decimal
        Dim BetaE As New Decimal
        Dim NumberA As New Decimal
        Dim NumberB As New Decimal
        Dim NumberC As New Decimal
        Dim NumberD As New Decimal
        Dim NumberE As New Decimal
        RndNum = New Random
        Alpha1A = RndNum.Next((AlphaMean - (Half * AlphaRange)) * 100, (AlphaMean
+ (Half * AlphaRange)) * 100)      ' Generates Random Number for Alpha A according
to details
        Beta1A = RndNum.Next((BetaMean - (Half * BetaRange)) * 100, (BetaMean +
(Half * BetaRange)) * 100)                ' Generates Random Number for Beta A
according to details

        AlphaA = (Alpha1A / 100)  ' Conversion to decimal with 0.1
        BetaA = (Beta1A / 100)   ' Conversion to decimal with 0.1

        Alpha1B = RndNum.Next((AlphaMean - (Half * AlphaRange)) * 100, (AlphaMean
+ (Half * AlphaRange)) * 100)
        Beta1B = RndNum.Next((BetaMean - (Half * BetaRange)) * 100, (BetaMean +
(Half * BetaRange)) * 100)

        AlphaB = (Alpha1B / 100)
        BetaB = (Beta1B / 100)

        Alpha1C = RndNum.Next((AlphaMean - (Half * AlphaRange)) * 100, (AlphaMean
+ (Half * AlphaRange)) * 100)
        Beta1C = RndNum.Next((BetaMean - (Half * BetaRange)) * 100, (BetaMean +
(Half * BetaRange)) * 100)

        AlphaC = (Alpha1C / 100)
        BetaC = (Beta1C / 100)

        Alpha1D = RndNum.Next((AlphaMean - (Half * AlphaRange)) * 100, (AlphaMean
+ (Half * AlphaRange)) * 100)
        Beta1D = RndNum.Next((BetaMean - (Half * BetaRange)) * 100, (BetaMean +
(Half * BetaRange)) * 100)

        AlphaD = (Alpha1D / 100)
        BetaD = (Beta1D / 100)

        Alpha1E = RndNum.Next((AlphaMean - (Half * AlphaRange)) * 100, (AlphaMean
+ (Half * AlphaRange)) * 100)
        Beta1E = RndNum.Next((BetaMean - (Half * BetaRange)) * 100, (BetaMean +
(Half * BetaRange)) * 100)

        AlphaE = (Alpha1E / 100)
        BetaE = (Beta1E / 100)


        Do While T < TL  ' Loop till the time unit limit
            T = T + 1

            NumberA = BetaA * MeanSale * ((T ^ AlphaA) - ((T - 1) ^ AlphaA))  '
Produces Marginal Value for Vendor A
            NumberB = BetaB * MeanSale * ((T ^ AlphaB) - ((T - 1) ^ AlphaB))  '
Produces Marginal Value for Vendor B
            NumberC = BetaC * MeanSale * ((T ^ AlphaC) - ((T - 1) ^ AlphaC))
            NumberD = BetaD * MeanSale * ((T ^ AlphaD) - ((T - 1) ^ AlphaD))
            NumberE = BetaE * MeanSale * ((T ^ AlphaE) - ((T - 1) ^ AlphaE))
            NumberA = Decimal.Round(NumberA, 1)  ' Decimal Round to 0.1
            NumberB = Decimal.Round(NumberB, 1)
            NumberC = Decimal.Round(NumberC, 1)
            NumberD = Decimal.Round(NumberD, 1)
            NumberE = Decimal.Round(NumberE, 1)


            Using cmd As New OleDbCommand(SqlString, conn)    ' Puts the marginal
values inside the vendor columns
                cmd.CommandType = CommandType.Text

                cmd.Parameters.AddWithValue("A", NumberA)
                cmd.Parameters.AddWithValue("B", NumberB)
                cmd.Parameters.AddWithValue("C", NumberC)
                cmd.Parameters.AddWithValue("D", NumberD)
                cmd.Parameters.AddWithValue("E", NumberE)
                conn.Open()
                cmd.ExecuteNonQuery()
                conn.Close()
            End Using
        Loop




    End Sub

    Private Sub Button1_Click(sender As System.Object, e As System.EventArgs)
Handles Button1.Click
        Form2.Show()
    End Sub
End Class
```

References

Abrache, J., Crainic, T. G., Gendreau, M., & Rekik, M. (2007). Combinatorial Auctions. *CIRRELT*, 1-2.

An Introduction to Digital Signage. (2011). Retrieved from wirespring: http://www.wirespring.com/pdf/intro_to_digital_signage.pdf

Arvind, K. T., & Suresh, K. N. (2006). Narrowcasting of wireless advertising in malls. *European Journal of Operationa*, 1023–1038.

Assmus, G., Farley, J.U. & Lehmann, D.R. (1984). How advertising affects sales: meta-analysis. *21*, 65-74.

Ausubel L. M. (2004). An efficient ascending-bid auction for multiple objects. *The American Economic Review*, 1452-147.

Ausubel, L. M., & Milgrom, P. (2006). The lovely but lonely Vickrey auction. *Combinatorial Auctions*, 17-40.

Bellamy, R., Swart, C., Wendy, A., Kellogg, Richards, J., & Brezin, J. (2001). Designing an E-Grocery Application for a Palm Computer: Usability and Interface Issues. *Personal Communications, IEEE*, 60-64.

Bodapati, & Anand, V. (1999). The impact of out-of-store advertising on store sales.

Boutilier, C., Dean, T., & Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *(JAIR)*, 1-94.

Burke, & Raymond, R. (2009). *Behavioral effects of digital signage.*

Buyya, R. (2002). Economic Paradigm for Distributed Resource Management and Scheduling for Service Oriented Grid Computing. *Ph. D thesis*.

Cavalieri, S., Garetti, M., Macchi, M., & Taisch, M. (2000). An experimental benchmarking of two multi-agent architectures for production scheduling and control. *Computers in Industry*, 139-152.

Clarke, D. (1976). Econometric Measurement of the Duration of Advertising Effect on Sales. *13*, 345-357.

Clarke, E. H. (1971). Multipart pricing of public goods. *Public choice*, 17-33.

deVries, & Vohra. (2003). Combinatorial Auctions: A Survey . *INFORMS Journal on Computing*.

Ferguson, D. F., Nickolaou, C., Sairamesh, J., & Yemini, Y. (1995). Economic Models for Allocating Resources in Computer Systems.

Fujishima, Yuzo, Leyton-Brown, K., & Shoham, Y. (1999). Taming the computational complexity of combinatorial auctions Optimal and approximate approaches. 548–553.

Ghani, R., Cumby, C., Fano, A., & Krema, M. (2008). Data Mining for Consumer Modeling and Personalized Promotions. *Data mining methods and applications*.

Gijsbrechts, E., Campo, K., & Goossen, T. (2003). The impact of store flyers on store traffic and store sales: a geo-marketing approach. 1-16.

Global Industry Analysts, I. (2011). *Digital Signage Systems: A Global Strategic Business Report*. Retrieved from Global Industry Analysts, Inc.: www.strategyr.com

Groves, T. (1973). Incentives in teams. *Econometrica: Journal of the Econometric Society*, 617-631.

Gurley, S. (2010). *Ad-Funded Digital Signage: Is There A Future In It?* Retrieved from www.symon.com

Gutierrez, B. P. (2006). In-Store Media: How Effective are they? Evidence from the Philippines. *Philippine Management Review*, 65-82.

Gutierrez, B. P. (2008). In-store media: How effective are they? Evidence from the Philippines. *Philippine Management Review*, 65-82.

Harris, B. (2006). *If not TV…?* The Hub.

Harrison, J. V., & Andrusiewicz, A. (2004). A virtual marketplace for advertising narrowcast over digital signage networks. *Electronic Commerce Research and Applications*, 163-175.

Harrison, V., & Andrusiewicz, A. (2003). *ENHANCING DIGITAL ADVERTISING USING DYNAMICALLYCONFIGURABLE MULTIMEDIA.*

Huberman, B. A., & Clearwater, S. (1995). A multi-agent system for controlling building. *Proceedings of the First International Conference*, 171–176.

Jin, S., Zeng, Y., & Wang, C. (2010). Assessment of advertising effectiveness through audience's eye movements. *Proceedings of the 2010 ASME international design* *engineering technical conferences & computers and information in engineering conference*.

Jones, J. P. (2006). *When ads work: new proof that advertising triggers sales.* ME SHARPE Inc.

Jones, J. P. (2007). *When Ads Work.* New York: MESharp.

Kagel, J. H., & Levin, D. (2002). *Common Value Auctions and the Winner's Curse.* New Jersey: Princeton University Press.

Khouja, M., & Robbins, S. S. (2003). Linking advertising and quantity decisions in the single-period inventory model. *International Journal of Production Economics*, 93-105.

Kutanoglu E., Wu S. D. (1999). On combinatorial auction and Lagrangean relaxation for distributed resource scheduling. *IIE Trans.*, 813-826.

Kutanoglu, E., & Wu. (2006). Incentive compatible, collaborative production scheduling with simple communication among distributed agents. *International Journal of Production Research*, 421-446.

Mas-Colell, A., Whinston, M. D., & R., G. J. (1995). *Microeconomics.* Oxford University Press.

McAfee, P. R., & McMillan, J. (1996). Analyzing the Airwaves Auction . 75-159.

Menezes, F. M., & Monteiro, P. K. (2005). *An Introduction to Auction Theory.* New York: Oxford University Press Inc.

Morrison, G. (2008). *Excerpts from workshop on the Effectiveness of high-tech in-store media:Canadian Advertising Research Foundation.*

Müller, S. (2011). Auctions. *International Encyclopedia of the Social & Behavioral Sciences*, 917-923.

Parkes , D. C., & Kalagnanam, J. (2005). Models for Iterative Multiattribute Procurement Auctions. *Management Science*, 435-451.

Parkes, D. C., & Ungar, L. (2001). Proceedings of 5th International Conference on Autonomous Agents (AGENTS-01). *An Auction-Based Method for Decentralized Train Scheduling*, 43-50.

Parkes, D., Cramton, P., Shoham, Y., & R., S. (2006). Iterative combinatorial auctions. *Combinatorial Auctions*, 41–77.

Payne, T., David, E., Jennings, N. R., & Sharifi, M. (2006). Auction mechanisms for efficient advertisement selection on public displays in: 17th European Conference on Artificial Intelligence.

PRN. (2012). *media where & where it matters*. Retrieved 2012, from PRN: http://www.prn.com/services.html

Rassenti, Stephen, J., Smith, V. L., & Bulfin, L. R. (1982). A Combinatorial Auction Mechanism for Airport Time Slot Allocation. *Bell Journal of Economics*, 402-417.

Rassenti, Stephen, J., Vernon, L. S., & Bulfin, R. L. (1982). A combinatorial auction mechanism for airport time slot allocation. *Bell J. Econom*, 402–417.

Rayid Ghani, Chad Cumby, Andrew Fano,. (2008). Data Mining for Individual Consumer Models and Personalized Retail Promotions. *Auerbach Pub*, 201-223. Retrieved from http://www.bworldonline.com/

Reeves, D. M., Wellman, M. P., MacKie-Mason, J. K., & Osepayshvili, A. (2005). Exploring bidding strategies for market-based scheduling,. *Decision Support System*, 67-85.

Rogers, A., David, E., Payne, T. R., & Jennings, N. R. (2007). the 6th international joint conference on Autonomous agents and multiagent systems. In *An advanced bidding agent for advertisement selection on public displays.* Hawaii.

Rothkopf, Michael, H., Aleksandar, P., & Harstad, R. M. (1998). Computationally manageable combinatorial auctions. *Management Sci.*, 1131–1147.

Sandholm T. (1999). Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence. MIT Press.

Sandholm, T. (2000). Issues in computational Vickrey auctions. *International Journal of Electronic Commerce*, 107-129.

Sandholm, T. (2002). Algorithm for optimal winner. *Artificial Intelligence*, 135-154.

Sandholm, T., Suri, S., Gilpin,, A., & Levine, D. (2005). CABOB: A Fast Optimal Algorithm for Winner Determination in Combinatorial Auctions. *MANAGEMENT SCIENCE*, 374–390.

Shen W. (2002). Distributed manufacturing scheduling using intelligent agents. *IEEE Intelligent Systems*, 88–94.

Shen, W., & Wang, L. (2006). Agent-based distributed manufacturing process planning and scheduling: A state-of-the-art survey. *IEEE Transactions on Systems, Man, and Cybernetics*, 563–577.

Smith, R. G. (1980). The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *EEE Transactions on Computers*, 1104-1113.

Uta Juttner, Martin Christopher, Susan Baker. (2005). Demand chain management-integrating marketing and supply chain management. *Industrial Marketing Management*, 377-392.

Vickrey, W. (1961). Counterspeculation, auctions, and competitive sealed tenders,. *The Journal of finance*, 8-37.

Waller, M., Johnson, M. E., & Davis, T. (1999). Vendor-managed inventory in the retail supply chain. *Journal of business logistics 20*, 183-204.

Wang, C., Wang, Z., Ghenniwa, H. H., & Shen, W. (2011). Integrated Due Date Management through Iterative Bidding. *IEEE Transactions on Systems, Man and Cybernetics*, 1182-1198.

Wellman, M. P., Walsh, E., Wurman, P. R., & MacKie-Mason, J. K. (2001). Auction Protocols for Decentralized Scheduling, Games and Economic Behavior. 271-303.

Wikipedia. (2012, 09 28). *Auction*. Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Auction

Williams, & Diane. (2005). *The Arbitron Retail Media Study: Volume I: The Impact of Retail Audio Broadcasting in Grocery and Drugstores.* Arbitron.

Xue, D., Sun, J., & Norrie, D. H. (2001). An intelligent optimal production scheduling approach using constraint-based search and agent-based collaboration. *Computers in Industry* , 209-231.

Yackey, B. (2010). *DSE: Walmart reveals 18-month results for SMART Network, Net World Alliance article.* Retrieved from digitalsignagetoday: http://www.digitalsignagetoday.com/article/159854/