ABSTRACT

Title of dissertation: MODELS FOR BUDGET CONSTRAINED AUCTIONS: AN APPLICATION TO SPONSORED SEARCH & OTHER AUCTIONS

Abhishek Pani, Doctor of Philosophy, 2010

Dissertation directed by: Professor Subramanian Raghavan
Robert H. Smith School of Business

The last decade has seen the emergence of auction mechanisms for pricing and allocating goods on the Internet. A successful application area for auctions has been sponsored search. Search firms like Google, Bing and Yahoo have shown stellar revenue growths due to their ability to run large number of auctions in a computationally efficient manner. The online advertisement market in the U.S. is estimated to be around $41 billion in 2010 and expected to grow to $50 billion by 2011 (http://www.marketingcharts.com/interactive/us-online-advertising-market-to-reach-50b-in-2011-3128/). The paid search component is estimated to account for nearly 50% of online advertising spend.

This dissertation considers two problems in the sponsored search auction domain. In sponsored search, the search operator solves a multi-unit allocation and pricing problem with the specified bidder values and budgets. The advertisers, on the other hand, regularly solve a bid determination problem for the different keywords, given

their budget and other business constraints. We develop a model for the auctioneer that allows the bidders to place differing bids for different advertisement slots for any keyword combination. Despite the increased complexity, our model is solved in polynomial time. Next, we develop a column-generation procedure for large advertisers to bid optimally in the sponsored search auctions. Our focus is on solving large-scale versions of the problem.

Multi-unit auctions have also found a number of applications in other areas that include supply chain coordination, wireless spectrum allocation and transportation. Current research in the multi-unit auction domain ignores the budget constraint faced by participants. We address the computational issues faced by the auctioneer when dealing with budget constraints in a multi-unit auction. We propose an optimization model and solution approach to ensure that the allocation and prices are in the core. We develop an algorithm to determine an allocation and Walrasian equilibrium prices (when they exist) under additive bidder valuations where the auctioneer's goal is social welfare maximization and extend the approach to address general package auctions. We, also, demonstrate the applicability of the Benders decomposition technique to model and solve the revenue maximization problem from an auctioneer's standpoint.

MODELS FOR BUDGET CONSTRAINED AUCTIONS: AN APPLICATION TO SPONSORED SEARCH & OTHER AUCTIONS

by

Abhishek Pani

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2010

Advisory Committee:
Professor Subramanian Raghavan, Chair/Advisor
Dean Gnanalingam Anandalingam
Professor Peter C. Cramton (Dean's Representative)
Research Professor Kislaya Prasad
Associate Professor Sivakumar Viswanathan

Acknowledgments

I would like to thank my advisor, Professor S Raghavan for his guidance, patience, advice and invaluable inputs during the course of my thesis reseach. I look forward to working with him in the future on other interesting research projects.

I am grateful to Dean Anandalingam for his invaluable insights and guidance throughout the process. I would also like to thank Professor Peter Cramton, Professor Kislaya Prasad and Professor Siva Viswanathan for agreeing to serve on my thesis committee and for their feedback on my thesis.

I would also like to acknowledge the contributions of other faculty members at the University of Maryland who either collaborated with me or taught me during graduate school. A special thanks to Professors Michael Ball, Galit Shmueli, Wedad Elmaghraby, Larry Ausubel and Bruce Golden.

I want to acknowledge the help and encouragement from my family and friends. It would have been impossible to complete this dissertation without their support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Auctions have been used since ancient times to price and allocate items. High value art items and wholesale flower trade in Netherlands have used auctions for centuries. In the last decade, the growth of the internet along with that in computational power, has resulted in various auction forms gaining a great deal of prominence in the business and academic community. Some of the biggest names in the e-commerce domain, like eBay, Yahoo, Bing, Facebook and Google, are in fact large auction operators. Auctions have also been used by various governments to sell public goods like wireless spectrum, airport landing slots etc. Many traditional firms like Sears and Home Depot [EK03] have successfully used auctions for procurements and have observed significant savings. The growth in the usage of auctions in the industry has spurred researchers in the academic community to study the same. Although there have been many significant contributions made by the academic community in studying the economic properties, bidding behavior and computational aspects of the many auction formats, a majority of the studies ignore the presence of budget constraints. Budget constraints are very important from the standpoint of both the bidders participating in the auction process as well as the auctioneer and thus, in this dissertation, we seek to address computational and modeling issues faced by bidders as well as the auctioneer, when faced with hard budget constraints. We focus largely on the sponsored search domain. This dissertation consists of four essays. The first essay models the sponsored search problem that the search providers like Google, Bing and Yahoo face. We develop more general sponsored search auction mechanisms than currently used in practice. The second essay deals with modeling the sponsored search problem faced by the search advertiser bidding on various sponsored search platforms. We develop a methodology that allows large advertisers to optimally bid in sponsored search auctions. The third essay addresses computational issues in determining the outcome of an auction run by a revenue maximizing auctioneer, where the bidders face a hard budget constraint and have additive valuations for the items. In the fourth essay, we model the auctioneer's objective as social surplus maximization while the bidder assumptions remain the same as in the third essay. Here we wish to find a social welfare maximizing allocation along with a set of stable prices.

1.1 Online Advertising & Sponsored Search

The growth in internet commerce and usage has affected the advertising industry fundamentally. The last few years has been marked by the explosive growth of companies like Google, Facebook and Yahoo, which while providing a host of free services, derive their revenues primarily from online advertising. In fact, the tremendous profitability of this medium has attracted the likes of Microsoft and Ask to come up with their own advertising platform. From the perspective of the advertisers, online advertising provides the best return on investment amongst all alternatives as the advertiser has a much better idea about the end-user intent and profile. Also, since the cost of advertising on the internet is much smaller than television or print, search advertising provides a platform for small businesses, which earlier would have been priced out by larger firms from the advertising space, to compete on an equal footing. This new form of advertising has affected traditional advertising channels like Yellow Pages, since small, local advertisers can target the same audience more effectively online.

Online advertising includes search advertising, referrals and lead generation ads, banner ads, email ads, content related ads, social ads and rich media ads. The most prominent of these formats is sponsored search advertising. Recent reports on internet advertising estimate the revenues to be around $50 billion for 2011 (http://www.marketingcharts.com/interactive/us-online-advertising-market-to-reach-50b-in-2011-3128/). Large advertisers on the internet include firms in domains such as financial services, consumer electronics, retail, leisure, automotive and entertainment. Search based advertising accounted for about 50% of the revenue while display advertising, social media advertising, referrals and classifieds accounted for a significant proportion of the rest. Although internet advertising has demonstrated impressive growth rates in the last few years, the amount spent on it as a proportion of the entire advertising budget is still less than 10 per cent. Growing usage and penetration rates will push significant ad budgets from other mediums to the internet. Also, the ability to dynamically determine suitable ads to display along with better understanding of the user intent, based on textual or search keyword analysis, makes the online ad market very attractive to firms using the internet for commercial purposes.

GoTo (Overture Services), provided the first sponsored search platform on the internet in 1998. Overture was later acquired by Yahoo and formed the basis for its ad platform till the launch of its new platform, Panama, in 2007. Google modified the Overture model by including click-thru rates for various keywords and introduced its system in 2002. In the last four years, Ask.com and Bing have also adapted a model similar to that of Google, with modifications to account for demographic and geographic targeting. All the search platforms mentioned above have the following format - when a search user specifies a set of keywords to search for, the search engines display algorithmic (*organic*) results. They also display a set of results, clearly marked as advertisements, to the top or right of the screen. These text ads, in response to the user query, are called sponsored search ads. The ad typically consists of a title and a short description related to the business, along with the business's URL. The placement of the ad is determined differently by various search engine operators. These ads also follow the pay-per-click (PPC) model. In the PPC framework, the advertiser has to pay Google, Bing or Yahoo only if the search user clicks on the respective ad. The payment rules for the advertiser's also varies with the search platform. Most of the popular platforms use a $next-price$ payment rule in which the advertiser pays an amount related to the bid of the advertiser positioned immediately below her. Also, one should note that all the above allocations and price determinations are performed under budget constraints. Hence, the problem that Google and others seek to solve is one of expected revenue maximization such that the budget constraints of bidders are not violated. On the other hand, the bidders have to determine their bids, knowing that they have an explicit budget constraint. Thus, unlike standard auction literature, where budget constraints are rarely discussed, any study related to the sponsored search domain has to explicitly account for the budget constraint. The presence of the budget constraint, however, significantly increases the computational complexity of the problem and also, results in incentive issues that cannot be addressed by extending results from auction models for the unconstrained problem.

1.2 Literature Review

The last six years have seen a number of papers being published that specifically deal with the problem of sponsored search. Feng et.al. [FBP07] discuss the various sponsored search allocation mechanisms, specifically that of Google and Yahoo (Overture). They use simulations to compare the revenues on various platforms under different ranking schemes. Parkes & Sandholm [PS05] discuss the significant limitations in the bidding language being used on various ad platforms. They emphasize the need for an expressive bidding language for the sponsored search domain and detail the benefits that search providers and advertisers can obtain by having this additional expressiveness. The paper provides a framework to deal with computational issues for auctions that have greater expressiveness. Goodman [Goo05] deals with one of the significant issues faced by the sponsored search domain, namely click fraud. The author describes a pay-per-percentage of impressions model, instead of the current pay-per-click model, to make the ad system immune from click and impression fraud. Edelman & Ostrovksy [EO05] empirically show that bidders display strategic behavior for keyword combinations and that the strategic behavior persists over time. Animesh et. al. [ARV05] investigate the difference in search behavior observed in the sponsored search domain depending on the type of good being sold. They classify the goods as experience or credence goods and determine some of the drivers for this difference in behavior. Jansen & Resnick [JR05] investigate the interaction between the organic and sponsored search term on the buying behavior of a search user.

Mehta et.al. [MSVV07] address the budget constrained issue in search auctions and provide an algorithm to maximize the auctioneer's revenue. They use an online bi-partite matching algorithm along with a trade-off revealing Linear Program(LP) that takes into account the bid as well as remaining budget for each advertiser.

Edelman et.al. [EOM07] and Varian [Var07] independently show that the current next price auctions being used by Google and Yahoo are not incentive compatible, contrary to what intuition may suggest. Aggarwal et.al. [AGM06] address the issue of non-incentive compatibility in current sponsored search auctions and describe a method to convert these auctions into truthful auctions. Andelman & Mansour [AM04] formulate a multi-unit allocation problem with budget constraints and discuss an approximation method for solving the problem. However, their formulation doesn't properly model the sponsored search problem. The paper models a one-shot allocation and pricing problem where bidders have additive valuations and face hard budget constraints. In the sponsored search problem, however, the budget varies over the various rounds and the participants and items being auctioned off also do not remain the same in the multiple rounds. Borgs et.al. [BCI+05] demonstrate the lack of incentive compatibility in auctions with budget constraints.

Zhou & Lukose [ZR06] show that equilibrium behavior in sponsored search auctions is vulnerable to vindictive bidding. Iyengar & Kumar [IA06] present a general model for the *pay-per-click* auction where the advertisers have multi-dimensional private valuations per click. They characterize the set of incentive compatible and individually rational allocation rules. Liu et.al. [LJA06] discuss a class of auctions called the weighted price auctions and its relation to the sponsored search problem. They also analyze the equilibrium bidding strategy for this class of auctions. Asdemir [Asd06] conducts an empirical study to identify bidding patterns observed online and develops a one shot simultaneous move and an infinite horizon alternative move game to model the observed patterns.

Although the sponsored search problem from the search engine's perspective has been relatively well studied, the problem faced by the advertiser has very limited literature. Kitts & Leblanc [KL04] formulate the general problem that the advertiser faces and discuss some methods used in the industry to estimate the system parameters. Ruschmevicheintong & Williamson [RW06] model the advertisers problem when the number of slots available is just one. They discuss the stochastic knapsack problem and its relation to the sponsored search problem. They further develop an adaptive algorithm to solve the advertiser's problem when the click-thru rates are unknown. The paper, however, does not discuss or model the real-world case of multiple ad slots with a combinatorial set of keywords.

Researchers have studied the economic properties of multi-unit auctions in great detail. Specifically, Vickrey-Clarke-Groves(VCG) mechanism properties and associated computational details have been addressed in a number of papers. However, there are very few papers that explicitly talk about the economic properties or computational issues faced by auction participants, when faced with hard budget constraints. In Benoit & Krishna [BK01], the authors show that when there are two items and two bidders with a budget constraint, then if the items are auctioned of sequentially, it is always optimal to sell the more valuable object first. The paper only deals with the case of common value items and does not model cases when more than two objects are present. However, it is evident that even in the multi-unit case, the sequence in which one auctions of the items will result in different revenues for the auctioneer. Thus, we need to consider all of the items at the same time and not sequentially. Che & Gale [YG00] consider the selling of a single item and discuss a non-linear pricing approach for the optimal allocation in the presence of budget constraints. Ausubel & Milgrom [AM02] describe a multi-unit, ascending auction format that can be suitably modified to account for hard budget constraints. However, their solution approach is a heuristic and the paper does not give a complete characterization of the problem being solved. Borgs et.al. [BCI+05] discuss the lack of known incentive compatible (IC) mechanisms under strict budget constraints. It is not even known whether there exists an IC mechanism when hard budget constraints are present.

1.3 Objective of this Dissertation

The rapid revenue growth in the paid search advertisement segment in the last five years has resulted in a number of theoretical and empirical research papers by researchers belonging to diverse domains such as computer science, operations research, economics, marketing and statistics. As discussed in the previous section, there has been significant work, particularly by the computer science community, related to economic and computational properties of various sponsored search platforms. There is, however, very limited amount of research that deals specifically with the problem that advertiser's on such platforms face. Further, modeling the budget constrained multi-unit allocation problem has not been addressed well in the standard auction literature. This dissertation seeks to address gaps in existing research in these areas. We specifically model the sponsored search problem faced by the search platform provider and the advertiser as well as discuss computational and modeling issues for a multi-unit allocation problem with budget constraints, in a general setting.

The first essay (chapter 2) deals with the optimization problem that sponsored search platform providers face. We discuss current allocation and pricing models prevalent in the search space. We, further, model the online ad allocation and pricing problem faced by firms like Google and Yahoo as an assignment problem. Current bidding systems allow only a single bid per keyword combination at a specific point in time.

In our model we allow the option to specify different willingness-to-pay values for the various positions. Further, we discuss the application of the Hungarian algorithm to solve the assignment problem and its relation to the Vickrey-Clarke-Groves (VCG) set of prices. Currently, the bidders are allocated upto one ad slot, of uniform dimension, in any auction round. However, it is expected that biddable video and graphic ads will be allowed by the various search providers in the near future and thus, the bidders could be allocated more than one ad slot of standard dimension in any iteration. We model this extension where the bidder has an option to bid on contiguous set of ad slots and show that optimization techniques can be used to allocate space to multi-media ads.

In the second essay (chapter 3), we discuss the problem faced by an advertiser on the current sponsored search platforms. We use the toolkit of integer programming(IP) to model the problem faced by the search advertiser. We examine the constraint structure of the problem and show that the LP relaxation will have an optimal solution that is integral for nearly all decision variables. In fact, we demonstrate that we need not consider all positions while bidding for a keyword combination. Rather, we should consider only those points that lie on the upper convex hull of the (total revenue, total cost) plot for the specific keyword combination. Further, bidders like eBay and Amazon, bid on millions of keywords daily on the various search platforms . Thus, loading the bid data at different positions and solving the LP relaxation is in itself a difficult task. We use the specific properties of the sponsored search domain to design a column generation approach to solve the relaxation problem faced by large advertisers. We, then, extend the column generation method and use a branch-and-price approach to build an algorithm to solve the IP to optimality.

In the third essay (chapter 4), we formulate a multi-unit auction with additive valuations and hard budget constraints. The objective of the auctioneer is assumed to be revenue maximization. The problem is modeled using integer programming and the computational intractability of the problem, in certain budget ranges, is demonstrated using simulations. A Benders decomposition method is then developed and used to solve the problem for these hard instances and an associated auction interpretation is provided.

The fourth essay (chapter 5) again deals with a multi-unit auction with additive valuations and hard budget constraints. However, the objective for the auctioneer is social welfare maximization. Solving the social welfare maximization problem will result in an allocation that maximizes overall value but the prices determined could be such that the bidders and the auctioneer would have a motivation to deviate and increase their respective surpluses. Hence, we need to modify the constraint space for the standard social maximization model under budget constraints to obtain prices and an allocation that are in the core (i.e., stable). In particular, assuming they exist, we would like single item prices (i.e., Walrasian equilibrium prices) in the additive setting. Also, of note is that unlike the unconstrained auction problem where VCG mechanism can be used to determine the efficient allocation, for the budget constrained auction, the VCG mechanism has no specific economic property. Thus, we need to simultaneously determine item prices as well as allocation. Building upon Day & Raghavan [DR07], we provide a constraint generation approach to deal with the exponential number of constraints associated with the core and provide an algorithm to solve the separation problem. We, also, extend the model to general package auctions where the valuations are not restricted to be additive.

Chapter 2

Optimizing Sponsored Search Auctions

In the recent past, revenues from sponsored online search based ads for firms like Google and Yahoo has grown tremendously. Potential advertisers provide their valuations for various keywords, and also specify overall budget for a period of time. The search operator then solves a multi-unit allocation problem with the specified bidder values and budgets, and determines two things - the order to place the ads and the amount to charge the respective bidders. In this chapter, we describe a model and method based on mathematical programming, to solve the online sponsored ad selection problem as an assignment problem. This would enable search engines to efficiently solve the problem in real time and increase their revenue by solving the problem to optimality. In contrast to the current sponsored search auction models, our model also allows bidders to specify different willingness-to-pay based on allocated position. We also consider the situation where text ads of varying length or a combination of text and graphical ads can be displayed. This introduces a packing problem and we describe two potential auction models for this problem.

2.1 Introduction

In the last few years, a new, targeted form of consumer advertising has emerged on the internet. These ads based on keywords specified by the consumer are called sponsored search ads. Sponsored search keyword advertising has become a multi-billion dollar industry with firms like Google deriving a majority of their income from sponsored search. The industry has shown double-digit growth in revenues and profit over the same period, and continues to expand its use of sponsored ads as a medium for growth. This chapter describes an enhanced auction format for sponsored search that can potentially increase the revenue from the advertisers in sponsored ad site activities.

The essence of the problem involves potential advertisers willing to pay for being featured at the top of a list of sponsored sites whenever users (i.e. potential consumers) do a search on some keywords, and the search company (like Google) wanting to be paid when these links are clicked on. Generally, the higher the placement in the set of sponsored links the better it is for the advertiser in terms of generated traffic. Potential advertisers know this and usually bid higher amounts to get placed higher in the set of displayed sponsored links. Conversely, the search engines only get paid if there is a click on the sponsored link. Thus, the search engine has an incentive to place ads that are of high quality (i.e. a higher propensity for the user to click on) and have attracted suitably high bids.

In fact, in their use of sponsored ad-sites to generate revenue, search engines such as Google, Bing and Yahoo have become large auction houses where they solve an allocation and pricing problem each time the consumer initiates a search. The bidders specify the maximum amount that they are willing to pay for a keyword and also specify overall budget limits for a pre-defined time period across a set of keywords. The search engine operator provides options for placing bids on more than one keyword or even combinations of keywords. Based on these bidder inputs, the search engine operators use heuristics to determine the order of display as well as the price paid by each advertiser for the ad placement.

The sponsored ad auction has attracted a lot of attention in the research community recently. Feng et al. [FBP07] analyze and compare different sponsored search allocation mechanisms as well as explore the effect of total number of slots auctioned on auctioneer revenues. Rolland & Patterson [RP03] propose an expert system to match advertisers and web users. Lim & Tang [LT05] use a simple one-stage game theoretic model with two bidders to study how advertisers compete for positions in a search engine based on their expected click-thru rates. Weber & Zhang [WZ07] study paid placement strategies and find that in order to maximize revenue, the search engine design should be based on a weighted average of relative quality performance in addition to bid amount. Kumar, Dawande & Mookerjee [KDM03] propose a complex pricing model to maximize the search engine's revenue; the price is based on the number of impressions of the ad and the number of clicks on the ad. There has also been some work that has examined efficient algorithms for allocating sponsored site positions. For example, Zhan, Shen & Feng [ZSF09] examine simultaneous pooled auctions and methods to improve revenue as well as suggest a modified Vickrey-Clarke-Groves(VCG) mechanism that the auctioneer can use to get the same expected revenue as in the pooled auction.

Varian [Var07] characterizes the Nash equilibrium of Google sponsored auctions and shows that the generalized second price auction (followed by Google, Bing and Yahoo) is not incentive compatible and is not equivalent to the VCG mechanism. Aggarwal et al. [AGM06] independently show the lack of truth telling as a dominant strategy for these auctions and develop a truthful mechanism called the ladder auction. In the next section, we describe these two papers in greater detail.

2.2 Current Sponsored Search Models

The models which various sponsored search engines use for ad auctions are a variant of the auction run by Google. The auction mechanism that Yahoo used prior to 2007 can be interpreted as a special case of the Google mechanism. Varian [Var07] uses the phrase "position auctions" to describe various variants used in the sponsored search domain. The paper considers an auction problem with m bidders and K slots. V_{ik} denotes the value that bidder i has for slot k. A key assumption in the paper is that the bidder valuations for slots are *separable* i.e., the value of a bidder for a slot can be expressed in terms of a bidder specific factor and a position specific factor. Thus, $V_{ik} = U_i.CTR_k$, where U_i is the expected profit per click for bidder i and CTR_k is the click-thru rate of ad slot k. It is assumed that CTR is a monotonically decreasing function with respect to the position i.e., the CTR at a given position will be strictly greater than the CTR of the position below. Let b_i be the bid that bidder i places in the ad auction. Varian [Var07] first describes an auction model where the price paid by bidder i for an allocated slot k is equal to the bid placed by the bidder allocated slot $k + 1$. The paper, then, provides a link between the symmetric Nash equilibrium prices for this auction setting and the assignment game in which bidders can be assigned no more than one ad slot. Under the assumption that bidder's valuation is separable, the position auction is nothing but a competitive equilibrium of the assignment game.

Varian [Var07] also provides an insight into the actual Google ad auction. The allocation and pricing of slots occurs in two phases. In the first phase, Google sorts the bidders in decreasing order of their "value" to Google. Here, the value to Google is defined as the product of advertiser's quality score and bid. The slots are assigned based on this ordered list. Bidders with greater value to Google will be assigned the higher slots. In the second phase of the auction, the price paid by each bidder is determined, conditioned on the slot allocated in the first phase. Each bidder is expected to pay the minimum amount to retain the allocated slot. For example, assume slot $k + 1$ is allocated to bidder $i + 1$. Bidder $i + 1$ has bid b_{i+1} and has a quality score of e_{i+1}. If bidder s is assigned slot k, then the price that bidder s would pay, if a web surfer clicks on the ad, is $\frac{b_{k+1}e_{k+1}}{e_s}$, where e_s is the quality score of bidder s. Further, the paper demonstrates that the position auctions are not incentive compatible.

Aggarwal et.al. [AGM06] independently show the lack of incentive compatibility in sponsored search auctions. They, however, define a new auction format, called the laddered auction, that they show is incentive compatible. Under assumption of separability for the click-thru rate (i.e., the click-thru rate can be separated into a bidder specific and a position specific factor), they show that their auction is revenue equivalent to the currently used position auctions.

All search providers ask a bidder to provide a single bid associated with a keyword. Aggarwal et.al. [AGM06] suggest that Google also uses a single bid for the respective keywords along with the expected click-thru rate at position 1. There are three limitations of with existing sponsored search auction implementations, as well as the research of Varian [Var07] and Aggarwal et.al. [AGM06]. The first issue is that they do not allow bidders to express differential willingness to pay based on slot position. In fact, this single bid that is currently obtained from the bidder makes a VCG implementation suffer from an effect similar to loss of "voter-sovereignty" [AGM06]. Secondly, it is assumed that the quality score for an advertiser is independent of position. Lastly, the models proposed by Varian [Var07] and Aggarwal et.al. [AGM06] assume a separable structure for bidder valuations and click-thru rates respectively. Our model allows bidders to bid differently for the various slots and thus, provides an option to improve rank under a VCG implementation. We, also, provide the option to have differential quality scores for the bidders as a function of position. Further, our model does not rely on the assumption of separability of either the bidder valuations or the click-thru rates.

2.3 Proposed Assignment Model

In this section, we address all the limitations of the current sponsored search models described in the previous section and model closely the sponsored ad problem. We formulate the item pricing and allocation problem from the perspective of the auctioneer (search engine operator). The auctioneer sells K slots to the bidders and aims to maximize its expected revenue while adhering to specified budget constraint and individual rationality, based on declared values. The auctioneer is assumed to have estimates for the click-thru rates (i.e. the probability that a user will click on an ad at a specific position) based on statistical analysis of historical data. The bidders, in turn, bid for combination of various key words at the respective slot positions and specify their aggregate budget for the day.

The setup for our model is as follows - a user specifies the search words to the search engine operator. The search engine operator (auctioneer) has bids and budget from the advertisers as well as historical data that indicates the probability that the displayed ads will be clicked on. The bidder then solves an optimization problem to maximize the expected value subject to various constraints.

The click-thru rate defined above is a function of the "degree of match" and the "position". The degree of match is defined as the number of keywords in the bidder defined bundle that match what is specified by a user (e.g., - if an internet user searches for the words 'hat coat' and say a bidder has bid for a bundle 'hat coat shoes', then the degree of match is 2). In our model, we assume that the click-thru rates are symmetric i.e. any keyword set with the same degree of match will have at the same click-thru rate at the respective position. For example, let us assume that the web user specified the search term *hat coat shoe*. Then, in our model we assume that the click-thru rates for the words *hat* and *coat* will be the same for a given bidder. This assumption is not restrictive but allows us to reduce the number of active bids in our model.

Let us assume there are m bidders and K slots (or positions). $V_{ij}^{\prime k}$ denotes the value that bidder i has for keyword bundle j at slot k. Let b_i be the current budget level of bidder i. The decision variables for the above problem are $x_{ij}^k \in \{0, 1\}$ (the allocation of keyword bundle j at position k to bidder i). The estimated click-through rate for bundle j at position k is denoted by c_j^k. The auctioneer's objective is to maximize the total expected value obtained from all the bidders.

The number of bids from each bidder in the model can also be reduced from an exponential order to linear in the number of keywords as follows. For each degree of match, at every position, select the maximum bid from the bidder. Thus, the number of active bids from each bidder reduces from order $K(2^n - 1)$ to Kn , where n is the number of keywords specified by the user and K is the number of ad slots. The index j, henceforth, is used to represent the number of keyword matches i.e.

the degree of match. Note that the degree of match for a keyword combination is incorporated in the estimated click-thru rate, c_j^k. Also, in the pre-processing stage, for every user specified search, the auctioneer will take the current remaining budget of bidder i into account to determine the respective input bid for any bidder i.e. $V_{ij}^k = min\{V_{ij}'^k, b_i\}$. The auctioneer's problem can be stated as:

$$\max \quad \sum_{i=1..m} \sum_{j=1..n} \sum_{k=1..K} V_{ij}^k c_j^k x_{ij}^k \tag{2.1}$$

$$\text{s.t.} \quad \sum_{i=1..m} \sum_{j=1..n} x_{ij}^k \leq 1; \text{for } k = 1..K \tag{2.2}$$

$$\sum_{j=1..n} \sum_{k=1..K} x_{ij}^k \leq 1; \text{for } i = 1..m \tag{2.3}$$

$$x_{ij}^k \in \{0, 1\}; \text{for } i = 1..m, j = 1..n, k = 1..K \tag{2.4}$$

Constraint (2.2) indicates that no slot can be allocated to more than one bidder while constraint (2.3) indicates that no bidder can be allocated to more than one slot. Constraints (2.2) and (2.3) together constitute a TUM matrix. In fact, the above problem is an assignment problem where every bidder is bidding on specific slots. For every bidder and for each slot k, find $V_i^k = max_j\{V_{ij}^k c_j^k\}$ [1]. The assignment problem that the auctioneer solves is with V_i^k as the value that bidder i has for slot k and the decision variable being x_i^k.

[1]if click-thru rates are not symmetric, then V_i^k can be calculated over an exponential number of bids.

$$\begin{aligned}
Obj = \max \quad & \sum_{i=1..m} \sum_{k=1..K} V_i^k x_i^k && (2.5)\\
\text{s.t.} \quad & \sum_{i=1..m} x_i^k \leq 1; \text{for } k = 1..K && (2.6)\\
& \sum_{k=1..K} x_i^k \leq 1; \text{for } i = 1..m && (2.7)\\
& x_i^k \in \{0,1\}; \text{for } i = 1..m, k = 1..K && (2.8)
\end{aligned}$$

Thus, the auctioneer can solve a linear program to determine the allocation. Alternatively, specialized algorithms like the well-known Hungarian method [Wol98] obviate the need for using any optimization software to solve the problem. The above method can easily be extended to settings where the click-thru rate is a function of other parameters like reputation of the bidder along with degree of match. For example, c_j^k can be modeled as the product of three functions, $f(i)$, *g(j | length of search term)* and $h(k)$. $f(i)$ is a measure of bidder reputation for bidder i, $h(k)$ is an exponential decay function to measure the effect of position on click-thru rate[FBP07] and *g(j | length of search term)* measures the degree of fit for keyword j, given the search term specified by the search user.

From Leonard [Leo83], we know that we can get the VCG price and allocation for an assignment model by solving two sequential linear programs. The first problem to be solved would correspond to the primal assignment problem described above. The second problem to be solved would use the dual variables u_i and v_k. The v_k variables will correspond to the VCG prices while u_i will be the respective bidder surpluses. The second LP is

$$
\begin{aligned}
\min \quad & \sum_{k=1..K} v_k && (2.9)\\
\text{s.t.} \quad & u_i + v_k \geq V_i^k; \text{for } i = 1..m, k = 1..K && (2.10)\\
& \sum_{i=1..m} u_i + \sum_{k=1..K} v_k = Obj && (2.11)\\
& u_i, v_k \geq 0; \text{for } i = 1..m, k = 1..K && (2.12)
\end{aligned}
$$

Here Obj is the objective function value corresponding to the optimal solution to the primal assignment problem. Note however, we can in fact, obtain the allocation and VCG prices by solving the assignment problem using the Hungarian algorithm [BdVSV02]. Demange et.al. [DGS86] use the Hungarian approach but increase the price of each item in the over-demanded set by a single unit in each iteration. We will, however, determine the price increment in each iteration using the standard Hungarian procedure i.e., the dual variables for the assignment constraints may change by more than a single unit.

Note that solving the assignment problem with the Hungarian algorithm provides expected VCG payments. To find the actual amount that a bidder pays each time her ad is clicked on, we proceed as follows: If a bidder i has been allocated slot k, then the amount the bidder will pay is $\frac{v_k}{c_j^k}$; where c_j^k corresponds to the click-thru rate for the bundle that maximized $V_{ij}^k c_k^j$ for bidder i and v_k is the VCG price for the keyword combination k.

The sponsored ad auction model we describe here is in fact equivalent to bidders bidding in a multi-unit setting with unit demand (as the ad auctions are run repeatedly during the day). This naturally raises the question whether using VCG is each round is incentive compatible. We show that, even under this multi-round setting, solving for the VCG outcome for every search instance is not incentive compatible as the following theorem illustrates.

Theorem 1. *For the sponsored search auction, modeled as a static game with one round, truthful reporting of true value and budget is a dominant strategy for every bidder under VCG mechanism. However, for multiple rounds under budget constraints, the VCG solution for each round does not result in truthful reporting for the bidders.*

Proof. Let us assume that the bidder's have the same value for any slot that they are allocated i.e. they place a single bid for all slots. We will now consider the single stage as well as the multi-stage auction setting.

1. Single stage auction - The bidders declare their value for slots (denoted by v_i) and the budget (denoted by (b_i)). The auctioneer selects those bidders for whom the $v_i \leq b_i$ i.e., only those bidders that have sufficient budget to participate in the auction. Thus, the budget constraint is not binding for any of the bidders and is not relevant to the single round pricing and allocation problem. The proof of truthfulness is identical to that for VCG.

2. Multi stage auction - Let the true value and budget of the bidders be (v_1, b_1) and (v_2, b_2) respectively. Also, let $b_1 > v_1$ and $v_2 > b_2$. Further, we assume $v_1 > v_2$. Assume that both the bidders bid their true values and budgets. We shall show that one of the bidders can do better ex-post by deviating from her true value.

   For the first round, bidder 1 gets the slot and pays v_2 to the auctioneer, under the VCG scheme. Thus, the remaining budget of bidder 1 is $(b_1 - v_2)$. For the second round, the effective bid of bidder 1 will be $min\{v_1, (b_1 - v_2)\}$ while that of bidder 2 remains v_2. If $v_1 \leq (b_1 - v_2)$, then bidder 1 gets the slot and pays a price of v_2. However, if $v_1 > (b_1 - v_2)$, then the following two cases can occur:

(a) If $v_2 > b_1/2$, then the slot is allocated to bidder 2 at a price of $b_1 - v_2$

(b) If $v_2 \leq b_1/2$, then the slot is allocated to bidder 1 at a price of v_2

Thus, we have a case corresponding to $v_2 > b_1/2$, where the price paid by the bidder is not independent of her bid. Thus, truth-telling is not a dominant strategy for the bidders.

□

Note that the theorem illustrates an extreme situation. Most search auctions involve a large number of bidders, multiple keyword bundles, large number of auctions through the day and a budget that is usually irrelevant for a particular instance or even many multiple instances of the problem. Thus, for most practical purposes, despite the result of Theorem 1, we believe, payments as discussed prior to Theorem 1 are appropriate.

There are two key aspects to the superiority of our proposed model over that being used by Google, Bing and Yahoo. First, the current search platforms accept only a single bid for the advertiser. The advertiser, thus, determines the position that maximizes her revenue metric for the keyword combination and bids accordingly. However, since the search platform provider doesn't guarantee a specific position, the advertiser could be allocated a lower position and end up paying higher than her value for that position. Thus, the model followed by Google or Yahoo will result in "advertiser regret". In our implementation, since we collect and use bids for each position that the keyword could be bid to, the advertiser doesn't face any regret no matter what position the keyword combinations are slotted at.

Secondly, in the current Google implementation, the estimated click-thru rate for the first position for each keyword combination is used to determine the order of allocation. In our model, we use the click-thru rates at all positions to arrive at the optimal answer and thus, are able to arrive at a superior value maximization outcome.

2.4 Models for Multiple Slot Allocation

The current sponsored search models do not provide the bidders an opportunity to be allocated more than one slot for a specific search instance. Search engines like Ask currently support graphical images in their search feature. It is, thus, conceivable in the future one could have a combination of text and image ads displayed in the sponsored search results. In the recent past, various search engines such as Google and Bing have explored options for placing search-based ads with formats differing from the standard text ads. These graphical ads could possibly span multiple slots and thus, the auctioneer would need to account for the varying dimensions of the image while allocating space to the bidders. Here, we describe two models for allocating sponsored search ads where the ad spans across contiguous slots.

Model 1(a)

In this model, the search engine operator can display both banner and text ads based on the search term specified. We aim to solve the winner determination problem when the auctioneer is intending to maximize revenue, given the bid declarations. This model can also be used for modeling the case of displaying only text ads but with variable length. Bidders bid for relative position and length of the displayed advertisement. The auctioneer aims to maximize revenue and the bidders are assumed to pay only if the ads are clicked on.

Let the basic unit of available space be a slot. The number of slots is assumed to be L. Hence, each bidder can place up to $L(L+1)/2$ bids. Also, let p indicate the package index. Each package is composed of two elements $[R_p,\ L_p]$, where R_p is the relative position of the advertisement and L_p is the length of the ad. For e.g., in a 3 slot allocation problem the number of possible packages that the bidder could place bids on would be 6 and the package index would be as follows:

Package Index (p)	Package Description $[R_p,\ L_p]$
1	[1,1]
2	[1,2]
3	[1,3]
4	[2,1]
5	[2,2]
6	[3,1]

The auctioneer wants to maximize her revenue subject to the following constraints:

1. No bidder is allocated more than one package.
2. The total length of the allocated packages cannot exceed the available space.
3. The allocation of slots is contiguous and no two allocated packages can have the same relative position.

Decision variables for the problem are:

1. $x_{ip} \in \{0,1\}$; 1, if bidder i is allocated package p; 0, otherwise
2. $y_p \in \{0,1\}$; 1, if package p has been allocated to any bidder; 0, otherwise

The formulation for the problem from the auctioneer's standpoint is :

$$\max \quad \sum_{i=1..m} \sum_{p=1..P} V_{ip} x_{ip} \tag{2.13}$$

$$\text{s.t.} \quad \sum_{p=1..P} x_{ip} \leq 1; \text{for } i = 1..m \tag{2.14}$$

$$\sum_{i=1..m} x_{ip} \leq y_p; \text{for } p = 1..P \tag{2.15}$$

$$\sum_{p=1..P} L_p y_p \leq L \tag{2.16}$$

$$\begin{aligned} &\sum_{p=1..L} y_p \leq 1 \\ &\sum_{p=2L..3L-3} y_p - \sum_{p=L+1..2L-1} y_p \leq 0 \\ &\sum_{p=3L-2..4L-6} y_p - \sum_{p=2L..3L-3} y_p \leq 0 \\ &...and \quad so \quad forth \end{aligned} \tag{2.17}$$

$$x_{ip}, y_p \in \{0, 1\}; \text{for } i = 1..m, p = 1..P \tag{2.18}$$

Here, P is the total number of packages. The number of equations in (2.17) will be equal to the number of slots available for the auction. Also, the formulation above can further be strengthened by generating cover inequalities corresponding to the 0/1 knapsack constraint (2.16). (2.14) indicates that no bidder can be allocated more than one package. (2.16) ensures that the length of the slots allocated doesn't violate the available slot length. (2.15) and (2.17) are feasibility constraints for the allocated packages. (2.17) checks for feasibility of package allocation, while taking the relative position of the packages into account. For example, in the three slot auction case, we can have only one of the packages with index 1, 2 and 3 be assigned to bidders in the optimal solution i.e. $y_1 + y_2 + y_3 \leq 1$. The relative position of the three packages is 1 and thus, these three packages would occupy overlapping slots, if more than one of the packages was assigned to the various bidders. (2.17) prevents the occurrence of this infeasible solution.

The model detailed above actually corresponds to solving an auctioneer's problem with exact keyword match. The Overture model (used by Yahoo prior to 2007) was a special case of the above general model ($L_p = 1$ for all packages). Although the model above doesn't take into account the click-thru rate, one can model the same by modifying the constants in the objective function to $c_{ip}V_{ip}$ instead of V_{ip} above, where c_{ip} is the click-thru rate. The modification for click-thru gives an allocation where the auctioneer is maximizing expected value over exact match bids only. The current Google model would, hence, be a special case of this modified formulation (again, L_p =1 for all packages and instead of c_{ip}, Google uses c_{i1}).

Model 1(b) *(inexact match; combinatorial bids on keywords):*

The package is now defined as [R_p, L_p, (keyword combination)]. The bidder submits the bids for various packages to the proxy agent. The auctioneer, then, finds for every feasible [R_p, L_p] combination the keyword grouping which maximizes the $c^j_{ip}V_{ip}$. Thus, once the auctioneer does this pre-processing, the problem effectively reduces to that described earlier in model 1(a).

Model 2

The bidders place direct bids on the contiguous slots. The model we propose here is the same as Model 1 but the mathematical formulation is different. If the auctioneer has 3 slots to auction off, the possible packages that the bidders could bid on would be

Package Index (p)	Package Description [Slots]
1	[1]
2	[2]
3	[3]
4	[1,2]
5	[2,3]
6	[1,2,3]

The package [1, 3] will not be a feasible package as the respective slots are not contiguous. If the number of slots being auctioned off is L, then the maximum number of packages that the bid on will be $L(L + 1)/2$.

Let i be the bidder index, j the package index and k the index on slots. The auctioneer's problem can be stated as maximizing value (or expected value, as the case maybe) subject to no package being allocated to more than one bidder and that the overall allocation should not have overlapping slots. The decision variable for the problem is x_{ij}, where x_{ij} is 1 if bidder i is allocated package j and is 0, otherwise.

$$\max \quad \sum_{i=1..m} \sum_{j=1..P} V_{ij} x_{ij} \tag{2.19}$$

$$\text{s.t.} \quad \sum_{j=1..P} x_{ij} \leq 1; \text{for } i = 1..m \tag{2.20}$$

$$\sum_{i=1..m} x_{ij} \leq 1; \text{for } j = 1..P \tag{2.21}$$

$$\sum_{j=1..P} a_{kj} (\sum_{i} x_{ij}) \leq 1; \text{for } k = 1..K \tag{2.22}$$

$$x_{ij} \in \{0, 1\}; \text{for } i = 1..m, j = 1..P \tag{2.23}$$

Here, P is the total number of packages. Equations (2.20) and (2.21) ensure that no bidder is allocated more than one package and no package is allocated to more than one bidder. (2.22) indicates that the allocated slots are part of a feasible package. Here, is a_{kj} a constant, which takes a value of 1, if slot k is part of package j and 0, otherwise. As discussed for Model 1, one can modify the objective function to include click-thru rate so that the auctioneer maximizes expected revenue.

Let us define matrix, A, such that a_{kj} is an element of matrix A. A will be a matrix with K rows and P columns, where K is the number of slots and P is the maximum package index. For the three slot auction example, A will have the following structure:

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

The constraints in the model (2.19)-(2.23) have the structure shown below:

$$\begin{pmatrix} e^T & & \\ & e^T & \\ & & e^T \\ I & I & I \\ A & A & A \end{pmatrix}$$

Here, e^T is a row vector of 1's while A is an interval matrix and therefore, TUM. In fact, the interval matrix A is identical for each slot. The first two blocks consisting of e^T and I correspond to the constraint set (2.20)-(2.21) while the last block comprised of the A's corresponds to (2.22). Notice that the various blocks are TUM. Preliminary results with this formulation show all LP relaxations are integral. The TUM structure of parts of the constraint matrix may offer an apparent explanation

Table 2.1: Max weight bipartite matching (Hungarian Approach) (in secs)

Slots	Bidders = 20	50	100	200	500
5	0	0	0	0.2	1.18
10	0	0	0	0	1.7
20	0	0	0	0	2.5
50		0	0	0	2.8
100			0	0.58	3.1
200				0.4	3.4
500					3.8

for this. Consequently, Model 2 is preferred over Model 1.

2.5 Computational Experiments

All of the models were tested with simulated data. The structure of the generated data for the sponsored search assignment problem was as follows - data was generated on the bids and click-thru rates for various keyword bundle, ad slot and bidder combination. Click-thru rates and bids varied with the degree of match and rank of the ad according to the following rules:

1. The bid for each match decreases with position (i.e. lower the position, lower is the willingness to pay).

2. For the same degree of match, the click-thru rate decreases with position (i.e. reduces as we go lower down in terms of position). Also, for the same position, the click-thru rate decreases as the degree of match reduce.

Table 2.1 shows the solution time for the ad assignment problem, solved using Hungarian algorithm implemented on an Intel Pentium M 1.6 GHz processor with 512 MB RAM. The algorithm was implemented in C. The time shown is the average time over 10 randomly generated problem instances for each bidder-slot combination.

Table 2.2: IP solution for Model 1 (in secs)

Slots	Bidders = 50	100	200
5	0.06	0.5	0.34
10	0.3	0.51	1.26

Table 2.3: IP solution for Model 2 (in secs)

Slots	Bidders = 50	100	200
5	0.18	0.28	0.64
10	0.66	1.21	3.27

The results show that we can, in fact, include click-thru rates, combinatorial set of keywords and differential willingness to pay for every bidder and still arrive at an optimal allocation almost instantaneously. This obviates the need for using heuristics, which is the current industry practice.

Tables 2.2 and 2.3 show the time to determine optimal allocation for the integer programs corresponding to the multiple slot allocation problems model 1 and 2, respectively. The solution was obtained using OPL Studio 7.0 on an Intel Pentium 801 MHz processor with 512 MB RAM. The time shown is the average time over 5 randomly generated problem instances for each bidder-slot combination.

The results show that an additional dimension of length for the search ads can be incorporated into the model and problem sizes typically faced by current search operators can be solved very quickly. Adding cover inequalities as well as analysis of the TUM structure of the constraint sub-blocks can possibly further reduce the solution time.

2.6 Conclusion

In this chapter, we have modeled the sponsored search problem faced by a search provider as an assignment problem and have shown a way to calculate the VCG prices for the same. Currently, a VCG based auction is used by Facebook for its ad platform. Our model includes bids for the various positions for each keyword combination unlike the current models where only a single bid is imputed for a specific keyword. Further, we account for the different click-thru rates that a keyword encounters at various positions. These modifications help to deal with advertiser regret in current models and also, provides a higher social value. The changes proposed in our model, however, still allow the problem to be solved nearly instantaneously.

We, also, extend the current models to include bidding on multiple contiguous slots, a feature not feasible in current models. This approach can be used for search advertising as well be extended for an auction based framework for banner ads of different sizes. We could, also, easily extend our modeling framework to include features such as volume discounts and bidding language to limit competition.

Chapter 3

Optimal Bid Determination in Sponsored Search

In the last seven years, internet search based ads have been the fastest growing segment in the entire advertising domain and account for nearly half of the online advertising spend. Potential advertisers provide their valuations for various keywords and also, specify overall budget for a period of time. The search operator, then, solves a multi-unit allocation problem with the specified bidder values and budgets, and determines two things - the order to place the ads and the amount to charge the respective bidders. The advertisers, on the other hand, regularly solve a bid determination problem for the various keywords, given their respective budgets and other business constraints. In this chapter, we provide an optimization-based methodology to solve the advertiser's bid determination problem in sponsored search auctions. We first prove some structural properties of the optimal solution in such a setting. Finally, we consider the large scale nature of this problem, and based on the structural properties of the optimal solution, develop an efficient solution methodology to rapidly solve this problem. Traditional linear programming (LP) and integer programming (IP) methods are not particularly suited for this problem as the largest advertisers are known to bid on close to a million keywords with shared revenue metrics many times in a day. Thus, even loading all the keyword data and performing simple data manipulation tasks like numeric ordering is a computationally intensive exercise.

3.1 Introduction

The search engines earn a large portion of their revenue by being a multi-unit, auction platform for text ad inventory. The search based advertising platforms like Google, Yahoo, Bing etc. provide the following basic framework - firms wanting to advertise select a set of keyword combinations to bid on. These advertisers also specify their respective daily budgets. When a search user types a word or a set of words in the search box, the platforms will match the user specified terms to the keywords bid on by the firms, determine an order to display the ads in and also determine the amount that the firm has to pay if the user clicks on the displayed ad. This model is often called the pay-per-click (PPC) model. The match type is selected by the firm and in general, belongs to one of the four types - broad, exact, negative and phrase match. Bing additionally provides functionality to further filter the match based on consumer demographics. Insufficient budget for a campaign can adversely affect the advertiser's performance as the corresponding ads are not displayed once the daily budget is exceeded.

The payment and allocation rules vary across various search platforms. However, they all are variants of Google's Adwords model. This model allocates the advertisers slots based on a multiplicative function of a proxy for the click-thru rates (often referred to as the quality score) and the advertiser's bid. The payment rule on the various platforms is, typically, an increment over the bid (or a function of bid and quality score) of the bidder allocated the immediate lower ad position.

Given the rules and mechanisms of the search ad platforms, the advertisers face a computationally difficult problem. The bids of the competitors for the respective keywords are not known. Also, the number of keywords that advertisers bid on daily varies from a few thousand for small enterprises to tens of millions for firms like eBay and Amazon. The cost curve, revenue potential and the click-thru rate that the advertiser faces for each keyword is an unknown function of the position the ad is shown on. Reports available on the various platforms let the advertiser know the daily number of times the ads corresponding to each of the keyword combinations were shown, the number of clicks that the ads got, the average position where the ad was displayed and the cost per click incurred by the advertiser for each keyword combination. Further, from analysis of the web data log files on the advertiser's server, the advertiser can make an estimate of the revenue generated per ad click corresponding to each keyword combination. Once these estimates have been determined, the firm still has to understand the performance trade-off between the keywords, given the daily allocated budget.

The rapid growth of the search ad domain has spurred research activity in the computer science, economics and operations research community specifically addressing issues related to various sponsored ad platforms. This research has primarily been focused on modeling the problem that search platform providers like Google and Yahoo face. Feng et.al. [FBP07] analyze and compare different sponsored search allocation mechanisms as well as explore the effect of total number of slots auctioned on auctioneer revenues. Rolland & Patterson [RP03] propose an expert system to match advertisers and web users. Lim & Tang [LT05] use a simple one-stage game theoretic model with two bidders to study how advertisers compete for positions in a search engine based on their expected click-thru. Weber & Zheng [WZ07] study paid placement strategies and find that in order to maximize revenue, the search engine design should be based on a weighted average of relative quality performance in addition to bid amount. Varian [Var07] characterizes the Nash equilibrium of Google sponsored auctions and also, shows that the generalized second price auction (followed by Google and Yahoo) is not incentive compatible and is not equivalent to the VCG mechanism. Aggarwal et.al. [AGM06] independently show the lack of truth telling as a dominant strategy for these auctions and develop a truthful mechanism called the ladder auction.

Although there have been a few papers addressing the problem faced by the search provider, there is very limited research into the problem faced by the advertisers. To our knowledge, there are only four papers of note that have attempted to address issues that the search advertisers face. Kitts & Leblanc [KL04] provide an overview of the various problems that the advertiser faces. They discuss the bid determination formulation as well as parameters that need to be estimated to solve the bid determination problem. However, they do not describe solution methods or solution algorithms. Rusmevichientong & Williamson [RW06] formulate the keyword selection problem that advertisers face. They first formulate the problem assuming knowledge about the click-thru rates for various keywords. The paper develops a link with the stochastic knapsack problem for the static case and uses that as a baseline to provide an adaptive algorithm when the click-thru probabilities are unknown. The authors do not consider multiple positions for the keyword and thus, are not able to capture the trade-off that occurs between keywords at various positions. Ghose & Yang [GY08] model the relationship between the click-thru rate, conversion rate and bid with variables such as as position, keyword length and information specificity (brand terms v/s retailer specific terms). They show that the revenue potential of keywords varies significantly when compared to each other. Further, they demonstrate a strong impact of the position attained by the keyword on observed revenue. Thus, the paper makes a strong case for using optimization techniques to determine optimal bids, although the authors do not address this issue in their paper. Feldman et.al. [FMPS07] discuss a simple heuristic based on randomization between two uniform bid strategies that advertisers can use to bid on keywords. The authors assume that the advertiser has flexible daily budgets and the auction market is static. The heuristic has the same bid for all keywords and adjusts budgets for the following day based on prior days overspend/ underspend.

To the best of our knowledge, this is the first time that (i) the structural aspects of the search advertiser's problem have been studied, and (ii) a solution algorithm to solve large-scale versions (i.e., of real-world size) of this problem has been provided. Given the novelty of sponsored search, there is a critical need for research tools and techniques that support advertisers in their efforts to bid optimally in search auctions. The methods described in following sections fill this academic and practical need.

3.2 Problem Formulation & Model Description

The popular sponsored search auctions (Google, Yahoo, Bing, Ask etc.) are, typically, next-price auctions. The advertiser determines the set of keyword combinations that she believes are related to her business and will generate the desired traffic. The advertiser then determines the amount of money that she is going to spend each day on the advertising campaign. Also, for each keyword combination selected, the advertiser has to place a bid. This bid determines the position that the ad will be placed in, whenever there is a search user who specifies the keyword combination. Each of the platform providers have their own proprietary method for determining the order of ad placement. However, although the systems are like a black-box for the advertiser, she can use her own historical data, along with aggregate historical data, to determine the various revenue and cost parameters.

The problem that the advertiser solves is of revenue maximization. We could model the objective as profit maximization too, by comparing the revenue generated by a click to the cost of the click. However, it appears that in the industry advertisers typically have specified budgets to spend and do not have utility for any unspent amount. Based on the past data, it is reasonable to believe the advertiser has available the estimated expected cost and the expected revenue or benefit for a keyword at the various positions. This may be obtained by experimentation. Please note that for all future discussions in this chapter the words keyword and keyword combination can be used interchangeably. Let the keywords the advertiser has selected to be bid for be denoted by i. Also, let pos_i be the position that the keyword is bid to, B be the advertiser's daily budget and bid_i be the bid for keyword i. bid_i is the decision variable for the advertiser. The advertiser has to estimate the following functions -

1. $pos_i = f(bid_i) \Rightarrow$ estimated position for keyword i when placing a bid of bid_i

2. $click_i = g(i, pos_i) \Rightarrow$ estimated daily clicks for the keyword combination i, if allocated position pos_i

3. $rev_i = h(i, pos_i) \Rightarrow$ estimated average revenue per click for keyword i if allocated to position pos_i

4. $cpc_i = l(i, pos_i) \Rightarrow$ estimated cost per click (CPC) for keyword i if allocated to position pos_i

The problem that the advertiser, thus, seeks to solve is -

$$\max \quad \sum_{i=1..I} rev_i.click_i \tag{3.1}$$

$$\text{s.t.} \quad \sum_{i=1..I} cpc_i.click_i \leq B \tag{3.2}$$

$$bid_i \geq bid_{i,min} \tag{3.3}$$

Here $bid_{i,min}$ is the minimum bid required for a keyword to participate in any auction on a search engine.

We now show below how the above problem is reformulated as an integer program. The input data to the IP model is -

1. R_{ij} - expected total revenue for keyword i at position j. This value is obtained by multiplying the revenue per click obtained at the various positions to the number of clicks that the keyword combination attracts at the respective positions.

2. C_{ij} - expected total cost for keyword i at position j. This value is obtained by multiplying the cost per click obtained at the various positions (available through the reports provided by Google, Bing, Yahoo etc.) to the number of clicks that the keyword combination attracts at the respective positions.

3. B - budget for the advertiser

The decision variable for the advertiser is the position that she should bid each keyword to. Once we know what position we should bid a keyword to, we can find out the related bid as we assume that we know the cost landscape (i.e. once we know the cost per click (CPC) for keyword i at some position j, then we need to bid a small increment over the CPC to get that position). Thus, the decision variable is -

x_{ij}, where $x_{ij} = 1$, if keyword i is allocated to position j and is 0, otherwise. Also, x_{0j} denotes a slack variable that takes a value 1, if the optimization model decides not to bid for the keyword i.e. chooses position 0. The corresponding total revenue and total cost for position 0 are R_{i0} and C_{i0}, both of which equal zero.

The optimization problem that the advertiser solves ($IPOPT$) is -

$$\max \quad \sum_{i=1..n} \sum_{j=0..m} R_{ij} x_{ij} \tag{3.4}$$

$$\text{s.t.} \quad \sum_{j=0..m} x_{ij} = 1; \text{for } i = 1..n \tag{3.5}$$

$$\sum_{i=1..n} \sum_{j=0..m} C_{ij} x_{ij} \leq B \tag{3.6}$$

$$x_{ij} \in \{0, 1\}; \text{for } i = 1..n, j = 0..m \tag{3.7}$$

Proposition 1. *$IPOPT$ is NP-complete*

Proof. If $j = 1$ i.e. only one slot is available for bidding, then $IPOPT$ reduces to the knapsack problem. Thus, it is NP-complete. □

Proposition 2. *The LP relaxation of $IPOPT$ can have no more than two fractional x_{ij}. Further, the fractional allocation corresponds to one keyword combination.*

Proof. The optimal solution to the LP relaxation of $IPOPT$ is a basic feasible solution and hence, can have upto $n+1$ of the x_{ij} variables as non-zero, as there are $n+1$ rows in the formulation. Due to the equality sign in the first set of constraints in $IPOPT$, the separable nature of these constraints will result in every constraint having at least one x_{ij} greater than zero. Thus, for the first set of constraints to be satisfied, either n of the x_{ij} variables will be greater than zero or $n + 1$ of the x_{ij} variables will be greater than zero. Hence, there are only three possibilities for the optimal solution to the LP relaxation - n of the x_{ij} variables equal 1, $n - 1$ of the x_{ij} variables equal 1 and two x_{ij} are fractional (with one being a slack variable) or

$n-1$ of the x_{ij} variables equal 1 and two x_{ij} variables are fractional, while adding upto 1. $\square$

For an advertiser bidding on any of the platforms (Google, Yahoo, Bing, Ask etc.), the following assumptions are observed to be true -

1. Number of clicks increase as we move to a higher position

2. Cost per click at a higher position is never lower than that at a lower position, for a given keyword. Thus, total expected cost increases at higher positions

3. Total revenue increases with position; thus, providing an incentive for the advertisers to bid for higher slots.

Theorem 2. *For every keyword combination i, find the convex hull of the (R_{ij}, C_{ij}) pairs. Any (R_{ij}, C_{ij}) pair not on the convex hull will not be in the optimal solution to the LP relaxation of $IPOPT$.*

Proof. Suppose that $x_{ij} > 0$ in the optimal solution for some (i, j) tuple, for which (R_{ij}, C_{ij}) is not part of the upper convex hull. Then -

$$\frac{R_{i,j+1} - R_{ij}}{C_{i,j+1} - C_{ij}} > \frac{R_{ij} - R_{i,j-1}}{C_{ij} - C_{i,j-1}}$$

Since x_{ij} is part of the optimal solution for the relaxed problem, the objective function value is $OPT = \sum_{ij} R_{ij} x^{\star}_{ij} = \sum_{i' \neq i, j' \neq j} R_{i'j'} x^{\star}_{i'j'} + R_{ij} x^{\star}_{ij}$ Now consider $x_{i,j-1}$ and $x_{i,j+1}$ and write a new solution as

$\hat{x}_{i,j+1} = x^{*}_{i,j+1} + \left(\frac{C_{ij} - C_{i,j-1}}{C_{i,j+1} - C_{i,j-1}}\right) x^{*}_{ij}$

&

$\hat{x}_{i,j-1} = x^{*}_{i,j-1} + \left(\frac{C_{i,j+1} - C_{ij}}{C_{i,j+1} - C_{i,j-1}}\right) x^{*}_{ij}$

Set $\hat{x}_{i,j} = 0$. Also, all other $\hat{x}_{i,j} = x^{*}_{ij}$

Observe, since $\sum_{j} x_{ij} \leq 1$ and $\left(\frac{C_{ij} - C_{i,j-1}}{C_{i,j+1} - C_{i,j-1}}\right) x^{*}_{ij} + \left(\frac{C_{i,j+1} - C_{ij}}{C_{i,j+1} - C_{i,j-1}}\right) x^{*}_{ij} = C_{ij} x^{*}_{ij}$ the new solution will be feasible for the original relaxed problem. Let New_OPT denote the objective function value of $\hat{x}$. Then $New\_OPT =$

$$\sum_{ij} R_{ij}\hat{x}_{ij} = \sum_{i' \neq i, j' \neq j} R_{i'j'}x^*_{i'j'} + R_{i,j-1}\left(\frac{C_{i,j+1} - C_{ij}}{C_{i,j+1} - C_{i,j-1}}\right)x^*_{ij} + R_{i,j+1}\left(\frac{C_{ij} - C_{i,j-1}}{C_{i,j+1} - C_{i,j-1}}\right)x^*_{ij}$$

Thus,

$$New\_OPT - OPT = \left(R_{i,j-1}\frac{C_{i,j+1} - C_{i,j}}{C_{i,j+1} - C_{i,j-1}} + R_{i,j+1}\frac{C_{ij} - C_{i,j-1}}{C_{i,j+1} - C_{i,j-1}} - R_{ij}\right)x^*_{ij}$$

$$= \left(\frac{(R_{i,j+1} - R_{i,j-1})C_{ij} - C_{i,j+1}(R_{ij} - R_{i,j-1}) - C_{i,j-1}(R_{i,j+1} - R_{ij})}{C_{i,j+1} - C_{ij}}\right)x^*_{ij}$$

But from assumption about the slope at j, we have

$$\frac{R_{i,j+1} - R_{ij}}{C_{i,j+1} - C_{ij}} > \frac{R_{ij} - R_{i,j-1}}{C_{ij} - C_{i,j-1}}$$

$$i.e. (R_{i,j+1} - R_{i,j-1})C_{ij} - C_{i,j+1}(R_{ij} - R_{i,j-1}) - C_{i,j-1}(R_{i,j+1} - R_{ij})C_{i,j+1} \geq 0$$

Thus, $New\_OPT > OPT$ and hence, $x_{ij} > 0$ cannot be part of the optimal solution. In other words, for every keyword combination i, only those (R_{ij}, C_{ij}) pairs that are on the upper convex hull can be part of the optimal solution. □

The (R_{ij}, C_{ij}) pairs are in a plane and the plot is monotonically non-decreasing. We use the following algorithm to determine the convex hull for each keyword.

Algorithm 1 (Convex Hull Algorithm for a Keyword)**.**

1. For the keyword i, initialize variables new_j to 0 and $convex\_list$ to $\{\}$.

2. Find the slopes (i.e. $\frac{R_{ij} - R_{i,new_j}}{C_{ij} - C_{i,new_j}}$) from new_j to all other $j's$ for keyword i, such that $j > new_j$.

3. Determine the j with the highest slope. This j lies on the convex hull.

4. Update $new_j \leftarrow j$ and $convex\_list \leftarrow convex\_list \cup \{j\}$.

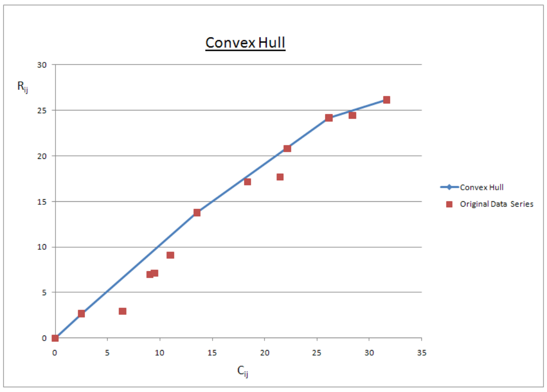

Figure 3.1: Convex Hull for a Keyword.

5. If new_j is not the highest j available for the given i, then GO TO Step 2. Else STOP.

Figure 3.1 illustrates the convex hull that we calculate for each keyword. Note that in the following sections, any reference to position corresponds to only relevant positions i.e. positions for the keyword that lie on the convex hull described above. Thus, all the indices for position would be referring only to the points on the convex hull. Further, index 0 corresponds to the origin.

3.3 Algorithm to solve LP relaxation

From proposition 1, we know that the advertiser's problem is computationally difficult. Based on proposition 2 we can now develop an algorithm to solve the LP relaxation of $IPOPT$ and get a solution where only one keyword combination will have a fractional allocation. The intuition for the algorithm to solve the LP relaxation to optimality is as follows - allocate the keyword combinations to various relevant positions greedily. Specifically, order the keyword-position pairs by their descending $bang-for-buck$ ratios. Keep on allocating the keywords in order till the allocation results in the budget constraint either being exactly matched or exceeded. If the keyword allocation that results in the budget constraint being violated has been already allocated to a certain position, then take a convex combination of the allocations to the current and previous allocation such that the budget constraint is satisfied. This final allocation will correspond to the fractional component in the optimal solution. The steps below formally describe the algorithm and solve the LP relaxation of the advertiser's problem to optimality.

Algorithm 2 (LP Optimal Algorithm)**.**

1. Initialize list to blank (the list has a structure $[R_{ij}\ C_{ij}\ flag_{ij}]$). For each keyword i, find (R_{ij}, C_{ij}) pairs that lie on the convex hull of the R_{ij} v/s C_{ij} plot and add to the list. Initially, all $flag_{ij}$ are set to zero. Also, i varies between 1 to n and j varies between 0 to m (0 is a dummy slot corresponding to no bid).

2. Take all the entries in the list obtained above and order the list by decreasing $\frac{\Delta R_{ij}}{\Delta C_{ij}}$, where $\frac{\Delta R_{ij}}{\Delta C_{ij}} = \frac{R_{ij}-R_{i,j-1}}{C_{ij}-C_{i,j-1}}$; Also, set B_{new} = current available budget (initially, set to B).

3. Let $x_{i0} = 1$ for all i (i.e. initially, each keyword is allocated to the dummy slot). Set all $flag_{i0}$ to 1.

4. Select the highest $\frac{\Delta R_{ij}}{\Delta C_{ij}}$ from the list that have $flag_{ij}$ set to 0. If none exists, then STOP

5. Let $\pi = \frac{B_{new}}{\Delta C_{ij}}$. If $\pi < 1$, then set x_{ij} to π, $x_{i,j-1}$ to $(1 - \pi)$ and STOP. Else set x_{ij} to 1 and $x_{i,j-1}$ to 0.

6. Set $flag_{ij}$ to 1, for the current i and j.

7. GO TO Step 4.

Proof. Here we provide a proof of optimality for the above algorithm. Let l be the last keyword assigned (to a position other than slot 0) before the budget is exhausted (i.e. the keyword under consideration in Step 5 when the algorithm terminates). The position that l is assigned is j. Also, let A be the set of keywords that are assigned to some position other than 0, i.e. $x_{i0} \neq 1$ for $i \in A$ and let $\overline{A}$ be the set of keywords which at the termination of the algorithm are still assigned to slot 0 i.e. $x_{i0} = 1$ for $i \in \overline{A}$.

The dual of the advertiser's problem (3.4)-(3.7) can be written as -

$$\begin{aligned} \min \quad & By + \sum_{i=1..n} \lambda_i \\ \text{s.t.} \quad & C_{ij}y + \lambda_i \geq R_{ij} \quad \forall i, j \\ & y, \lambda_i \geq 0 \end{aligned}$$

We construct a dual solution as follow. Let $y = \frac{\Delta R_{lj}}{\Delta C_{lj}} = \frac{R_{lj} - R_{l,j-1}}{C_{lj} - Cl,j-1}$, $\lambda_l = R_{lj} - C_{lj}y$. Also, let $\lambda_i = R_{i\bar{j}} - C_{i\bar{j}}y$ for $i \epsilon A$, where $\bar{j}$ is the position that i has been bid to at the termination of our algorithm. For $i \epsilon \overline{A}$, $\lambda_i = 0$. Now, to prove optimality of our algorithm, we will have to show primal feasibility, feasibility of the dual values described above and complementary slackness of the primal and dual solutions.

1. Primal feasibility - by construction of the algorithm

2. Dual Feasibility - For $i \epsilon \overline{A}$, $\lambda_i = 0$. Thus, we need to show that $C_{ij}y \geq R_{ij}$. Now, for $j = 0$, this holds trivially. For $j \neq 0$, since keyword l was selected at termination before any $i \epsilon \overline{A}$, by the ordering in our algorithm $\frac{\Delta R_{lj}}{\Delta C_{lj}} \geq \frac{\Delta R_{ij'}}{\Delta C_{ij'}}$, where $i \epsilon \overline{A}$ and $j' = 1..m$. From convexity of (R_{ij}, C_{ij}) we know that

   (a) $\dfrac{R_{i1} - R_{i0}}{C_{i1} - C_{i0}} > \dfrac{R_{i2} - R_{i0}}{C_{i2} - C_{i0}} > ... > \dfrac{R_{im} - R_{i0}}{C_{im} - C_{i0}}$

   (b) $\dfrac{R_{i1} - R_{i0}}{C_{i1} - C_{i0}} > \dfrac{R_{i2} - R_{i1}}{C_{i2} - C_{i1}} > ... > \dfrac{R_{im} - R_{i,m-1}}{C_{im} - C_{i,m-1}}$

(c) $\dfrac{R_{ij} - R_{i0}}{C_{ij} - C_{i0}} > \dfrac{R_{ij} - R_{i,j-1}}{C_{ij} - C_{i,j-1}}; \forall i, j$

Now, $R_{i0} = C_{i0} = 0$. Hence, using the ordering of our algorithm along with the convexity property, we have $y = \frac{\Delta R_{lj}}{\Delta C_{lj}} \geq \frac{\Delta R_{ij'}}{\Delta C_{ij'}}$ i.e. $y \geq \frac{R_{ij'}}{C_{ij'}}$ $\forall i \epsilon \overline{A}$ and $j' = 1..m$. Thus, dual feasible.

3. Complementary slackness - To show

   (a) $y(B - \sum_{ij} C_{ij}x_{ij}) = 0 \Rightarrow$ is true by the construction of our algorithm as we terminate when we hit the budget constraint i.e. $B - \sum_{ij} C_{ij}x_{ij} = 0$.

   (b) $\lambda_i(1 - \sum_{j} x_{ij}) = 0 \Rightarrow$ For $i\epsilon\overline{A}$, $\lambda_i = 0$. For $i\epsilon A$, $\sum_j x_{ij} = 1$ (by the algorithm). For $i = l$, we will have two cases - either $x_{i1} > 0$ or there exists two x_{ij} that are fractional and add upto 1. In the first case, $\lambda_l = 0$ by our definition while in the second case $\sum_{j=0..m} x_{lj} = 1$

   (c) $x_{ij}(C_{ij}y + \lambda_i - R_{ij}) = 0 \Rightarrow$ For $i\epsilon\overline{A}$, $x_{ij} = 0$. For $i = l$, $\lambda_l = R_{lj} - C_{lj}y$. Now for $i\epsilon A$, let $\overline{j}$ be the position to which i is bid. By our algorithm, $x_{ij} = 0$ for $j \neq \overline{j}$. For $j = \overline{j}$, $\lambda_i = R_{i\overline{j}} - C_{i\overline{j}}y$. Hence, all complementary slackness conditions are satisfied.

□

In Sinha & Zoltners [SZ79], a multiple choice knapsack problem is defined as follows -

$$
\begin{aligned}
\min \quad & \sum_{k=1..m} \sum_{j \epsilon N_k} c_{kj} x_{kj} && (3.8)\\
\text{s.t.} \quad & \sum_{k=1..m} \sum_{j \epsilon N_k} a_{kj} x_{kj} \geq B && (3.9)\\
& \sum_{j \epsilon N_k} x_{kj} = 1 && (3.10)\\
& x_{kj} \geq 0, integer && (3.11)
\end{aligned}
$$

Here, the coefficients c_{kj} and a_{kj} are non-negative and N_k is the set of multiple choice classes that are mutually exclusive. The authors describe the applicability of the above formulation to problems in capital budgeting, menu planning and system design for reliability. The paper discusses an algorithm to solve the LP relaxation of the above problem. The sponsored search problem can be mapped to the multiple choice knapsack problem as follows - the coefficient c_{kj} is similar to R_{ij}, a_{kj} to C_{ij} and the choice class N_k correspond to the various positions that we are bidding to. Also, the objective function changes to a maximization function instead to the minimization objective in the above definition. The algorithm detailed by Sinha & Zoltners [SZ79] is similar to the one described earlier for the sponsored search problem. However, our proof of optimality is more direct and quite different from that in Sinha & Zoltners [SZ79].

Zemel [Zem84] and Dyer [Dye84] independently provide an O(n) algorithm to solve the LP corresponding to the multiple-choice knapsack problem. Both papers use the convexity property of the dual function and a partitioning approach to arrive at an optimal solution. Pisinger [Pis95] uses an *expanding core* approach to determine the set of solution points that have a high probability of yielding an optimal answer for the IP.

3.4 Column Generation Approach

The bidding problem that the advertisers solve daily is very large. Typically, for medium to large sized advertiser the number of keywords that they bid on daily can vary from tens of thousands to about a million. The largest advertisers in the search domain like eBay and Amazon are rumored to place bids on millions of keywords everyday. However, the largest keyword groups that share a common set of revenue metrics, budget target and objective function is of the order of a few hundred thousand. Hence, solving the LP relaxation of the problem may itself be an insurmountable task. The huge number of variables and constraints involved, along with the natural decomposible structure of the advertiser's problem, necessitates the need for a column generation approach to solve the LP relaxation as well as embedding it within a branch-and-bound algorithm to solve the IP.

To solve the advertiser's optimal bid determination problem using column generation, we will use the following propositions.

Proposition 3. *Let A be the set of keywords which have columns corresponding to all their possible positions included in our consideration set. For a keyword $i \epsilon A$ allocated to a given position 'j' in the current iteration, the keyword can never be allocated to a position higher than 'j' in any future iteration. Thus, these columns can be removed from further consideration. As a consequence, for a given keyword $i \epsilon A$, any keyword not allocated to a position 1 to m in the current iteration will never be part of the final optimal solution and thus, can be deleted from the problem.*

Proof. Let some $i \epsilon A$, where i is a keyword with all its columns added in some prior iteration, be allocated to position j in current iteration. Thus, by our LP optimal algorithm, i was allocated to some position 1,2, .., $j-1$ in some previous iterations. In the current optimal solution keyword i is allocated to position j, implying $\frac{R_{i,j+1}-R_{ij}}{C_{i,j+1}-C_{ij}} \geq \frac{\Delta R_{l\bar{j}}}{\Delta C_{l\bar{j}}}$, where l is the last keyword allocated by our algorithm to some position $\bar{j}$. The entering variable for the next iteration will have columns with bang-for-buck greater than $\frac{\Delta R_{l\bar{j}}}{\Delta C_{l\bar{j}}}$. This new keyword, using our LP optimal algorithm will either be considered for allocation to a position before keyword i or after keyword i. If this new keyword is allocated before i, then the remaining budget when determining the position that i will be allocated to is less than that of the previous iteration. Thus, i cannot be allocated to a higher position than last iteration. If the new keyword is allocated after i, then the remaining budget when i is allocated is the same as the last iteration. Since, in the last iteration the keyword i was optimally determined to be allocated to position j with the given remaining budget, the allocation will remain optimal in the current iteration too. Thus, positions $j+1$ to m can never be part of the optimal solution in any future iteration and can be removed. □

Proposition 4. *Let y^k be the dual value of the budget constraint in the optimal solution obtained in the k^{th} iteration of the column generation approach. y^k is a monotonically non-decreasing function with respect to 'k'.*

Proof. From our proof of optimality for the LP optimal algorithm earlier, we know $y^k = \frac{\Delta R_{l\bar{j}}}{\Delta C_{l\bar{j}}}$, where l is the last keyword allocated to some position $\bar{j}$. In iteration $k+1$, some new keyword i, with its corresponding columns, is part of the consideration set. Thus, at least one of the $\frac{\Delta R_{ij}}{\Delta C_{ij}}$ is greater than $\frac{\Delta R_{l\bar{j}}}{\Delta C_{l\bar{j}}}$. Now using our LP optimal algorithm to solve for optimality in iteration $k+1$, we can have two options -

1. The last keyword assigned in iteration $k+1$ before the budget runs out is still l. This allocation will happen at position $\bar{j}$ (same as iteration k) or lower as the amount of budget available to l in iteration $k+1$ is now less than

that available in k. Hence, according to our algorithm, it will be assigned to position $\bar{j}$ or one of the lower positions (i.e. higher slope than $\frac{\Delta R_{l\bar{j}}}{\Delta C_{l\bar{j}}}$) Thus, $y^{k+1} = \frac{\Delta R_{lj}}{\Delta C_{lj}} \geq \frac{\Delta R_{l\bar{j}}}{\Delta C_{l\bar{j}}}$, where $j \leq \bar{j}$.

2. The budget, based on our algorithm, gets exhausted before we reach keyword l. Hence, the last keyword assigned in the $k+1^{th}$ iteration optimal solution has $\frac{\Delta R_{new,j}}{\Delta C_{new,j}} \geq \frac{\Delta R_{l\bar{j}}}{\Delta C_{l\bar{j}}}$, where *new* is the last keyword assigned to position j in current iteration. Thus, $y^{k+1} = \frac{\Delta R_{new,j}}{\Delta C_{new,j}} > \frac{\Delta R_{l\bar{j}}}{\Delta C_{l\bar{j}}} = y^k$

□

Proposition 5. *If $\frac{R_{i1}}{C_{i1}} \leq y$ for some $i\epsilon\overline{A}$, then the reduced cost for all j for the given i will be less than or equal to zero.*

Proof. Reduced cost for i at position $j = RC_{ij} = R_{ij} - yC_{ij} - \lambda_i$. Now, $RC_{ij} - RC_{i1} = (R_{ij} - yC_{ij} - \lambda_i) - (R_{i1} - yC_{i1} - \lambda_i) = (R_{ij} - R_{i1}) - y(C_{ij} - C_{i1})$. Thus, $\dfrac{RC_{ij} - RC_{i1}}{C_{ij} - C_{i1}} = \dfrac{R_{ij} - R_{i1}}{C_{ij} - C_{i1}} - y$. But from our assumptions we know, $C_{ij} > C_{i1}$ and $R_{ij} > R_{i1}$. Also, from convexity property, we have $\frac{R_{ij}-R_{i1}}{C_{ij}-C_{i1}} \leq \frac{R_{i1}}{C_{i1}}$. Now, $y \geq \frac{R_{i1}}{C_{i1}}$. Thus, $y \geq \frac{R_{ij}-R_{i1}}{C_{ij}-C_{i1}}$ and hence, $RC_{ij} - RC_{i1} \leq 0$. Note that $RC_{i1} = R_{i1} - \frac{\Delta R_{l\bar{j}}}{\Delta C_{l\bar{j}}} C_{i1}$ as $\lambda_i = 0$ $(i\epsilon\overline{A})$. Since, by our assumption, $y = \frac{\Delta R_{l\bar{j}}}{\Delta C_{l\bar{j}}} \geq \frac{R_{i1}}{C_{i1}}$, we have $RC_{i1} \leq 0$ and thus, $RC_{ij} \leq 0$ □

Proposition 6. *If the reduced cost of keyword 'i' ($i\epsilon\overline{A}$) at position 'j' is positive, then the reduced cost of the keyword 'i' at position 1 will also be positive.*

Proof. The reduced cost at position $j = R_{ij} - yC_{ij} - \lambda_i$, which is given as greater than zero. For $i\epsilon\overline{A}$, $\lambda_i = 0$. Thus, $\frac{R_{ij}}{C_{ij}} > y$. The reduced cost at position 1 for i is $RC_{i1} = R_{i1} - yC_{i1}$. Thus, $\frac{RC_{i1}}{C_{i1}} = \frac{R_{i1}}{C_{i1}} - y$. But from the convexity assumption, we have $\frac{R_{i1}}{C_{i1}} > \frac{R_{ij}}{C_{ij}}; \forall j > 1$. Hence, RC_{i1} is positive. □

The column generation method for solving the advertiser's problem would involve the following approach - start by including a small set of keywords (randomly chosen) and the respective relevant positions that these keywords could be bid to. Using propositions (3) to (6), in each iteration we add a certain number of keywords and respective columns from the set of keywords not under consideration. Also, we remove certain keywords or a subset of columns for keywords under current consideration based on the same set of propositions. Then using algorithm 2, we solve the current set of columns to optimality and check for feasible entering keywords in the next iteration. The algorithm terminates when no new keyword can be found to enter the set of columns under consideration.

The column generation algorithm to solve the LP relaxation is as follows -

Algorithm 3 (LP Column Generation)**.**

1. Select a set of keywords, A, and add corresponding columns at every position such that the budget, B, will be completely used in the current optimal solution. From the remaining set of keywords, $\overline{A}$, for each keyword add the column corresponding to the dummy position 0.

2. Solve the restricted LP master $\Rightarrow$ Use the LP optimal algorithm described earlier to solve the LP master to optimality. This will give us the optimal solution to the current set of columns. Also, let the dual variables corresponding to the budget constraint and allocation constraints be y and λ_i respectively. $y = \frac{\Delta R_{lj}}{\Delta C_{lj}}$, where l is the last keyword assigned to some position j before exhausting the budget. Also, $\lambda_i = R_{ij} - C_{ij}y$, if $x_{ij} = 1$ and $i \epsilon A$ in the current optimal solution; otherwise $\lambda_i = 0$.

3. Remove the columns corresponding to $i \epsilon A$ that follow proposition 3

4. Discard all $i \epsilon \overline{A}$ that have $\frac{R_{i1}}{C_{i1}} < y^k$, where k is the current iteration (follows from propositions 4 and 5).

5. To determine the entering keyword (and its associated columns), find $max\{\frac{R_{i1}}{C_{i1}}\}$, where $i \epsilon \overline{A}$. Note - although this will be the entering keyword, we may not select the keyword corresponding to $max\{\frac{R_{ij}}{C_{ij}}\}$, where $j = 1..m$ and $i \epsilon \overline{A}$

6. STOP if no column has positive reduced cost (or if within certain pre-determined bounds). Else, GO TO step 2

Note that we need not calculate the convex hull of the keywords that will enter the consideration set in an iteration. We just need to find the $max\{\frac{R_{ij}}{C_{ij}}\}$ for the respective i to figure out the first relevant position on the convex hull. Solving the LP relaxation using column generation approach may not terminate in an optimal solution that is integral. We can, however, use the bounds obtained while solving the restricted LP to solve for the IP.

The branching scheme that we propose would be on the original allocation variables x_{ij} for the keyword i with fractional allocation between positions j and $j+1$ i.e., the branching is of the form (A) $\sum_{k=j+1..m} x_{ik} = 0$ and (B) $\sum_{k=j+1..m} x_{ik} = 1$. A key point to note is that our proposed branching scheme preserves the structure of the LP column generation algorithm. Also to note is that points not on the convex hull can be part of the optimal IP solution. For e.g., let us consider a bidder with a budget of \$2.5 and a single keyword to bid upon. Further, assume the bidder's expected cost and revenue at positions 1, 2 and 3 are (\$3,\$6), (\$2,\$1.5) and (\$1,\$1) respectively. Position 2 does not lie on the convex hull. However, if we determine the optimal solution to the integer program, the bidder would select position 2 (i.e., select a point that is not on the convex hull).

We now briefly discuss the LP column generation as applied to the two branches (A) $\sum_{k=j+1..m} x_{ik} = 0$ and (B) $\sum_{k=j+1..m} x_{ik} = 1$. On branch (A), we need to find all points not lying on the upper convex hull between x_{ij} and $x_{i,j+1}$. Let there be p such points. Note that $x_{i,j+1}$ is not included in these p points. To apply our LP column generation procedure we need to compute (i.e., add to our existing points) the upper convex hull starting at x_{ij} along with these p points and add the associated variables. Also, delete the variables/ columns associated with the variables set to zero on branch (A).

On branch (B), we necessarily spend at least $C_{i,j+1}$. Thus, we reduce the budget by $C_{i,j+1}$ (i.e., $B \leftarrow B - C_{i,j+1}$ and solve the problem (using LP column generation and sorting)) starting at point $j+1$ on the upper convex hull. Notice that no new points are added to the upper convex hull on branch (B).

3.5 Computational Results

We ran the LP, LP column generation and IP column generation on problems of different sizes. Cost and revenue data was generated for each keyword-position combination from a uniform distribution. Further, monotonicity in both cost and revenue values was maintained with respect to position (i.e. the expected revenue and cost at a higher position was higher than that of a lower one). The assumption of monotonicity is in-line with the empirical results and observations in Feng et.al. [FBP07]. For e.g. - for a given keyword, the cost and revenue at the lowest position that we model for will be drawn from U[0,n]. Suppose the realization for cost is x. Then, the cost for the next higher position will be drawn from U[x,$x+n$]. Revenue values for each keyword-position combination are also generated by a similar approach. Note that for large advertisers, a small subset of keywords contribute a large part of the revenue. The expected revenue for most keywords at the various positions modeled for is negligible. However, in the revenue data we generated, the revenue distribution was not skewed. Thus, from a computational standpoint, the problem we solve is more difficult than the datasets in real-world.

The size of the problems that we ran the various algorithms on are typical of what advertisers face in the sponsored search space. The largest advertisers in the sponsored search domain bid on tens of millions of keywords daily. However, the largest keyword groups that share a common set of revenue metrics and budget targets is of a smaller order. Based on keyword groups that share similar revenue objectives, more than 99% of such keyword groupings have problem sizes from a few hundred to about a few hundred thousand keywords. Thus, our results are representative of the performance expectations that we can achieve solving real-world sponsored search problems. We are unable to solve for small sized SEM problems (e.g. 10,000 keywords at 30 positions) using OPL since the revenue and cost data for each keyword-position combination has to be uploaded into memory at the same time and thus, there doesn't exist sufficient system memory to solve the problem. The number of variables in the problem instances that we solved for range from 30,000 to 24 million. The corresponding number of constraints were between 1001 and 800,001. Table 3.1 indicates the size of the problems that we run our algorithms against. Note that the column generation algorithms have an additional parameter which provides the option of specifying the number of columns that can be added in each iteration.

The results indicate the time to solve the problem optimally, averaged over five problem instances. Further, the number of positions that we generated revenue and cost data for each keyword was thirty. The algorithms were implemented on a 3 GB

Table 3.1: Problem Dimension

Keywords	Bid Levels	# Variables	# Constraints
1000	30	30,000	1001
10000	30	300,000	10001
50000	30	1.5 million	50001
100000	30	3 million	100001
250000	30	7.5 million	250001
500000	30	15 million	500001
800000	30	24 million	800001

Table 3.2: Computational Time for Algorithms (in secs)

Keywords	Algo = LP	LPCG	B&P(multi)	B&P(single)
1000	0.263	0.345	0.957 (10)	0.3567
10000	7.56	4.58	15.523 (10)	12.1
50000	171.02	46.94	143.5 (10)	218.16
100000	667.44	188.96	713.87 (20)	1029.75
250000	4041.44	1187.02	4628.72 (25)	Unable to solve
500000	Unable to solve	7231.33	67020.21 (50)	Unable to solve
800000	Unable to solve	31802.21	Unable to solve	Unable to solve

RAM, 32 bit machine with a 3-GHZ Intel Pentium processor speed. The code was implemented in Python on a Linux environment. The results are shown in Table 3.2.

The numbers in the bracket refer to the parameter that controls the number of columns added in one iteration (i.e., number of columns added in each iteration <= number of keywords/number of iterations).

The results in Table 3.2 indicate that we are able to solve for medium sized problems (upto 250,000 keywords) using the LP approach. However, for larger problems, the LP approach does not work since loading the parameters associated with all the decision variables into RAM along with ordering the list of keyword-position tuples by diminishing marginal values takes a long time. This occurs due to swapping of data between RAM and the system hard disk since the RAM does not have sufficient space to accommodate all the data. In such instances, the LP column generation approach can be used to determine the optimal solution. For the same system specification, we are able to solve for problem sizes that are at least three times larger (800,000 keywords) than ones we can address using the LP technique. We have, thus, demonstrated two approaches to solve the linear relaxation problem for problem sizes that are significantly larger than the ones that can be be solved by OPL.

We also demonstrate the use of the Branch & Price method to solve the IP formulation for medium to large sized problems (100,000 to 500,000 keyword problems). Again, we are able to solve for problem instances in the integer programming case that are larger by an order of magnitude than the ones that can be solved by OPL.

In the advertising campaigns that are run by large online advertisers, no single keyword contributes to a significant part of revenues or cost when measured across all keywords. Since no more than one keyword can have a non-integral allocation in the optimal solution to the LP relaxation of the original problem, we can follow a randomization strategy to determine a feasible integral solution from the LP relaxation solution. We could pick one of the two positions with a certain probability for the keyword with non-integral solution and have an integral solution. Due to the relative insignificance of a single keyword for an advertiser, we would expect this randomization strategy to arrive at an integral solution that gives us an expected revenue very close to that of the IP optimal solution while ensuring that the spend is close to the targeted budget. An example of a randomization strategy would be to select position j with probability π and position $j-1$ with a probability of $1-\pi$ (refer to Step 5 in Algorithm 2 for the definition of π).

Advertisers may have hard budget constraints for a period of time (e.g., for a month or a quarter) but typically are willing to accept small variations in daily budget spent on advertising. The randomization strategy that we propose could result in slight overspend or underspend on a daily basis. However, we would not be, in expectation, violating the hard budget constraint that is specified over the longer time period. In case the advertiser has a hard budget constraint on a daily basis, instead of randomizing over positions $j-1$ and j for the keyword with fractional allocation, we would just pick position $j-1$ and not exceed the daily budget constraint. Thus, we could use the LP column generation approach to produce near-integral solution for very large problem instances and come up with an integral solution that is close to optimal by either a randomization strategy on the keyword with fractional allocation variables or with a deterministic approach when faced with a hard budget constraint. We would be able to solve for problems larger in size than those that we address using the Branch & Price framework.

3.6 Conclusion

In this chapter, we have modeled the sponsored search problem from the advertiser's standpoint as a multiple choice knapsack problem and have shown that the optimal solution will have no more than one keyword with a fractional allocation in the optimal solution. Further, we develop a LP column generation approach and a Branch & Price method to deal with the huge number of variables that medium to large advertisers face.

The result show that it is possible to solve very large SEM problems using column generation approaches on a single machine with modest specifications. Further, the constraint structure of the problem, along with the fact that no single keyword contributes to a significant portion of the revenue, supports using LP column generation to arrive at a very good approximation to the optimal integer solution. Thus, the problem formulation and algorithms described in this paper can be used in a real business context by advertising agencies, SEMs and large advertisers to successfully come up with bids for the largest of campaigns that they manage.

Chapter 4

A Benders Approach to Solve Budget Constrained Auctions

Combinatorial auctions are an increasingly popular method used for allocation and pricing in various business transactions. Over the last few years, availability of computation power has resulted in the successful development of combinatorial auctions in the fields of procurement, supply chain coordination, transportation services, wireless bandwidth allocation and internet search. Although, the auctions have an underlying budget constraint, nearly all the auction applications ignore this constraint to make the problem computationally tractable and have an allocation and price that follows incentive compatibility. In this chapter, we consider the case of a budget constrained auction and attempt to develop a method to provide a computationally efficient solution using the decomposible structure of the problem.

4.1 Introduction

In the previous two chapters, we discussed an application of combinatorial auctions to the sponsored search space. We further described computational issues and economic properties that arise due to hard budget constraints, which need to be accounted for by the auctioneer as well as the bidders. Besides the sponsored search space, combinatorial auctions have been an active area of research for applications to domains such as supply chain planning and coordination, resource procurement, allocating public goods like wireless spectrum auctions, treasury market, network resource allocation and electricity pricing. Most of the academic literature as well as real world implementations do not take the hard budget constraints faced by participants into account. There is an implicit assumption that the participants have some budget constraints, but it is typically assumed that the bidders have accounted for these constraints in their bids. Thus, the auctioneer need not explicitly model the budget constraint in the allocation and pricing problem that she seeks to solve.

Elmaghraby & Keskinocak [EK03] discuss the application of combinatorial auctions to procurement of transportation services at Home Depot. The paper describes the bidding mechanism developed at Home Depot to allow carriers to bid on various lanes based on the network structure and demand at various nodes. The savings at Home Depot by using combinatorial auctions is estimated to be in millions of dollars. Bichler et. al.[BDHK06] provide details about the various features of a combinatorial auction in an industrial procurement framework. It is estimated that nearly 40% of firms that spend more than $100 million a year use auction mechanisms for procurement. Although it is difficult to obtain details about specific auctions in use, due to the proprietary nature of various implementations, the paper provides an insight into a combinatorial auction implementation at Mars,Inc. Caplice & Sheffi [CS06] explore the usage of combinatorial auctions in the procurement of freight transportation services. They discuss a winner determination problem with side constraints that include business guarantee constraints. Ball et.al.[BDH06] provide an application of combinatorial auctions to airport slot allocation.

Many researchers in the computer science and operations research community have addressed the computational issues related to combinatorial auctions. Lehmann et.al. [LMS06] discusses the hardness of algorithms for winner determination and details a search algorithm to solve the problem in a computationally efficient manner. Leyton-Brown et.al.[LBNS06] perform tests to determine the empirical hardness of the combinatorial auction problem. They create a test-suite to determine causes of hardness in problems beyond just the increase in problem size. Bikhchandani et.al. [BdVSV02] build on LP models to devise efficient implementations of the VCG mechanism. They also provide a primal dual framework for understanding ascending auctions implementations in a multi-unit setting. Day & Raghavan [DR09] define a compact, expressible language called matrix bidding to specify bids over all possible item bundles and show that winner determination with matrix bids is suitable for practical settings. Day & Raghavan [DR07] develop a constraint generation approach that determines allocation and prices in the core for the winner determination problem. Their implementation converges to the core prices faster than the clock proxy auction mechanism developed in Ausubel & Milgrom [AM02]

Most of the papers listed above, along with other work in the literature, typically ignore computational issues arising from side constraints. Budget and other side constraints are, however, an integral part of every auction and thus, need to be modeled explicitly while solving the winner determination problem. Andelman & Mansour [AM04] provide a model and approximate solution approach to deal with the multi-unit allocation problem with budget constraints, when the bidder valuations are additive. In this chapter, we provide a method to solve the budget constraint problem using the decomposible structure of the constraints. We provide a solution approach using Benders decomposition and discuss heuristics to obtain a good initial solution to start the iterative mechanism. Further, we provide a link between the Benders algorithm and an iterative auction implementation. We note that in this chapter the assumption is that the auctioneer is interested in revenue maximization.

4.2 Problem Definition and Model Formulation

The combinatorial auction that we consider in this chapter is as follows - an auctioneer (seller) has n items to sell. There are m bidders and bidder i has a valuation V_{ij} for item j. We will assume that the bidder valuations are additive. In many settings such as fantasy sports, internal resource allocation within a firm and other auctions settings where complementarity between items is not evident, additive valuations for items is a reasonable assumption. Thus, given the bidder valuations V_{ij} for the various items, the seller has to find an allocation and price for each item such that her revenues are maximized. The seller, however, has to ensure that the bidders are not charged beyond their budget limits.

Andelman & Mansour [AM04] studied the problem of finding a computationally efficient solution to this budget constrained problem. They formulate the problem as an Integer Program(IP) and develop an approximation algorithm using the solution to the LP relaxation and applying a rounding method. The formulation that they use is as follows -

$$\begin{aligned}
\max \quad & \sum_{i=1..m} p_i && (4.1)\\
\text{s.t.} \quad & p_i \le B_i; \text{for } i = 1..m && (4.2)\\
& p_i \le \sum_{j=1..n} z_{ij}.V_{ij}; \text{for } i = 1..m && (4.3)\\
& \sum_{i=1..m} z_{ij} \le 1; \text{for } j = 1..n && (4.4)\\
& z_{ij} \in \{0,1\}; \text{for } i = 1..m, j = 1..n && (4.5)
\end{aligned}$$

Here, z_{ij} is a binary variable denoting whether item j has been allocated to bidder i, p_i is the price that the seller charges bidder i for the allocated items, B_i is the budget of bidder i and V_{ij} is the bid (i.e. the declared valuation) of bidder i for item j.

Andelman & Mansour [AM04] show that the optimal allocation for an auction with even two bidders and identical bids and budget constraints is NP-Hard by reduction from the $PARTITION$ problem. In fact, an exact optimal allocation can be found for the budget constrained optimal allocation problem using dynamic programming with time complexity of the order $O(m4^n)$. They further develop an approximation algorithm for the problem by solving the LP relaxation of the above formulation and assigning an item j to bidder i with probability z_{ij}. The approximation ratio of the algorithm is shown to be 1.582.

However, it is hard to argue that participants in an auction would be satisfied with an approximate solution. Further, in many public good auctions, participants would like feedback on individual prices. The formulation of Andelman & Mansour [AM04] gives allocation and price for the various bundles. It does not, however, give individual item prices as an output. Typically bidders would like feedback on the individual prices for the final allocation since the bids in this setting are submitted individually on each item.

Our motivation is to find an exact allocation that maximizes revenue and also allows for individual item prices. We now describe an Integer Program that will allow us to determine individual item prices. Let B_i be the budget of bidder i and V_{ij} be the value that bidder i has for item j. The decision variables for the problem are $x_{ij} \in \{0,1\}$, the allocation of item j to bidder i and p_{ij} is the price that bidder i is charged for item j. The auctioneer aims to maximize the revenue.

$$\max \quad \sum_{i=1..m} \sum_{j=1..n} p_{ij} x_{ij} \tag{4.6}$$

$$\text{s.t.} \quad \sum_{i=1..m} x_{ij} \leq 1; \text{for } j = 1..n \tag{4.7}$$

$$\sum_{j=1..n} p_{ij} x_{ij} \leq B_i; \text{for } i = 1..m \tag{4.8}$$

$$p_{ij} \leq V_{ij} x_{ij}; \text{for } i = 1..m, j = 1..n \tag{4.9}$$

$$x_{ij} \in \{0,1\}, p_{ij} \geq 0; \text{for } i = 1..m, j = 1..n \tag{4.10}$$

The first constraint is the availability constraint (no item can be allocated to more than one bidder), the second constraint specifies the budget constraint while the third set of constraints correspond to the individual rationality (IR), with respect to the bidder's declared valuation. Individual rationality implies that the price of an item allocated to any bidder cannot be greater than the bidder's bid for that item. The objective function as well as the constraint set (4.8) are non-linear. We describe how to linearize and reformulate the above problem as a mixed integer program (MIP). Let t_{ij} be the fraction of the budget of bidder i allocated to item j and x_{ij} be a binary variable that determines allocation of item j to bidder i ($x_{ij} = 1$

denotes item j is allocated to bidder i). The MIP formulation is as follows -

$$\max \quad \sum_{i=1..m} B_i(\sum_{j=1..n} t_{ij}) \tag{4.11}$$

$$\text{s.t.} \quad \sum_{i=1..m} x_{ij} \leq 1; \text{for } j = 1..n \tag{4.12}$$

$$\sum_{j=1..n} t_{ij} \leq 1; \text{for } i = 1..m \tag{4.13}$$

$$B_i t_{ij} \leq V_{ij} x_{ij}; \text{for } i = 1..m, j = 1..n \tag{4.14}$$

$$x_{ij} \in \{0,1\}, t_{ij} \geq 0; \text{for } i = 1..m, j = 1..n \tag{4.15}$$

In the above formulation, the first constraint is the availability constraint, the second constraint specifies that the sum of the fractions of the budget allocated to items add up to to less than or equal to 1, while the third constraint is the individual rationality constraint for each bidder-item combination.

When we attempt to solve instances of the above MIP formulation using OPL Studio 3.7 (which uses CPLEX to solve the MIP on a machine with 801 Mhz processor and 512 MB RAM), we observe that many instances can be solved very fast. However, as the problem size increases, the solution time also increases. We also examined the effect of the budget level on the running time. We observe that in certain budget ranges, both the smaller and the larger problems take over an hour to run. Tables 4.1 and 4.2 below show how the solution time varies with respect to the budget level when the bids and budgets are drawn from a uniform distribution (10 instances each). Table 4.3 corresponds to runs when the bids are drawn from a uniform distribution but the budgets are the same for all bidders.

From these three tables, it is evident that in certain budget ranges, the problem takes a long time to solve. To try and solve these types of instances faster, we will

Table 4.1: Time to IP optimal; 100 bidders, 100 items

Budget Distribution	Avg Time (sec)
U[0,500]	4.26
U[0,250]	4.9
U[0,100]	6.73
U[0,50]	8.203
U[0,30]	abort (1 hr)

Table 4.2: Time to IP optimal; 100 bidders, 500 items

Budget Distribution	Avg Time (sec)
U[0,500]	49.625
U[0,400]	47.953
U[0,100]	abort (1 hr)
U[0,50]	295.31
U[0,25]	62.76

use the decomposible structure of the problem and apply Benders decomposition method to potentially devise a computationally efficient solution method for the problem.

Table 4.3: Time to IP optimal; 100 bidders, 100 items; Equal Budgets

Budget	Avg Time (sec)
160-200	3.934
110-159	4.078
60-109	4.543
19-59	109.029
11-18	abort (1 hr)
6-10	23.27
2-5	30.95

4.3 A Benders Decomposition Approach to the Budget Constrained Auction Problem

The Benders decomposition method separates out a mixed integer programming problem into a integral master problem with a set of complicating constraints and a set of linear sub-problems that can be solved independently. The procedure uses the master problem to fix the integer variables and the sub-problems to find the value for the continuous variables. Fixing the variables in the master problem helps us exploit the special structure of the sub-problems and solve the sub-problems efficiently. The sub-problem contributes a set of dual values that are used by the master problem to generate a suitable cut in each iteration that the current solution to the master problem violates. These constraints are also referred to as "Benders feasibility cuts" since they enforce the conditions for feasibility of the master problem. This process continues iteratively, with the master problem finding a new solution while the sub-problems generate feasibility cuts, till we reach the optimality condition.

From the MIP formulation discussed in the previous section, we can see that if we fix the values for x_{ij} in the problem, then the problem essentially decomposes into a set of independent problems by bidder. We use this observation to develop our Benders decomposition approach. The master problem that we seek to solve is an allocation problem i.e. we solve the master problem, with the Benders's feasibility cuts, to determine the allocation of item j to bidder i. The sub-problem, on the other hand, gives us the maximum willingness to pay by each bidder, for the given allocation in each iteration. The Benders master problem can be formulated as -

Master Problem

$$\max \quad Z \tag{4.16}$$

$$\text{s.t.} \quad \sum_{i=1..m} x_{ij} \leq 1; \text{for } j = 1..n \tag{4.17}$$

$$Z \leq Z_{max} \tag{4.18}$$

$$(x, Z) \in F \tag{4.19}$$

$$x_{ij} \in \{0, 1\}; \text{for } i = 1..m, j = 1..n \tag{4.20}$$

Z is a scalar and the value for Z that we obtain in each iteration determines an upper bound to the objective function of the original problem. F is the solution space defined by the Benders feasibility cuts generated in all prior iterations using the dual solution to the sub-problems in respective iterations. Thus, it is F that defines the set of constraints that links the prices obtained from the sub-problems to the allocation decision variables that are determined by solving the Benders master problem. Z_{max} is a very large number that determines an initial upper bound. In our case, we use the budget of the various bidders to determine an appropriate Z_{max}.

If we fix the allocation based on the solution to the master problem, then it is clear that we have a separable problem structure; i.e., our sub-problems decompose by bidder. Thus, each sub-problem can be interpreted as a bidder determining her willingness to pay for each item allocated to her in any iteration, conditional on her value for the individual items as well as the overall budget. The Benders sub problem that each bidder i solves is -

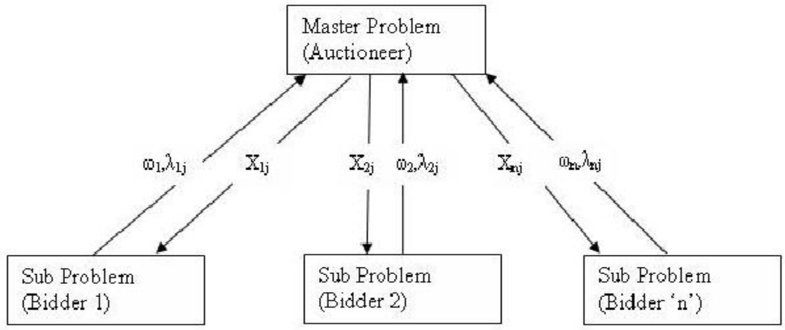

Figure 4.1: Information Exchange between Master and Sub-problems.

Bidder i's sub-problem

$$\max \quad B_i(\sum_{j=1..n} t_{ij}) \tag{4.21}$$

$$\text{s.t.} \quad \sum_{j=1..n} t_{ij} \leq 1 \tag{4.22}$$

$$\frac{B_i}{V_{ij}} t_{ij} \leq x_{ij}; \text{for } j = 1..n \tag{4.23}$$

$$t_{ij} \geq 0; \text{for } j = 1..n \tag{4.24}$$

The information exchange between the master problem (i.e. the auctioneer's problem) and the sub-problems (i.e. the bidder's problem) is shown in Figure 4.1. Here, ω_i is the dual variable corresponding to the first constraint in the formulation above while λ_{ij} is the dual for each of the constraints corresponding to second set.

The Benders decomposition algorithm for the budget-constrained auction problem is as follows.

Algorithm 4 (Benders Decomposition Algorithm)**.**

1. Set $k = 0$ (k is the iteration counter), Upper bound (UB^0) = Z_{max}, Lower

bound $(LB^0) = 0$, Z_{max} = a very large number and $F = \{(X, Z)|X \in B^{mn}, Z \in R\}$.

2. While $(UB^k - LB^k) > 0$

   (a) Set $k \leftarrow k + 1$

   (b) Solve master problem

$$
\begin{aligned}
\max \quad & Z \\
\text{s.t.} \quad & \sum_{i=1..m} x_{ij} \leq 1; \text{for } j = 1..n \\
& LB^{k-1} \leq Z \leq UB^{k-1} \\
& (X, Z) \in F \\
& x_{ij} \in \{0, 1\}; \text{for } i = 1..m, j = 1..n
\end{aligned}
$$

   (c) Set $UB^k \leftarrow Z$

   (d) Solve each of the sub-problems for the allocation $\overline{x}$ obtained above by solving the master problem. Each bidder i solves the following problem (which is the dual of bidder $i's$ sub-problem) -

$$
\begin{aligned}
v_i^k(\overline{x}) = \min \quad & \omega_i + \sum_{j=1..n} \lambda_{ij}\overline{x}_{ij} \\
\text{s.t.} \quad & \omega_i + \sum_{j=1..n} \frac{B_i}{V_{ij}} \lambda_{ij} \geq B_i \\
& \omega_i, \lambda_{ij} \geq 0; \text{for } j = 1..n
\end{aligned}
$$

   Let ω_i^k, λ_{ij}^k denote the optimal solution obtained by solving bidder $i's$ problem in the k^{th} iteration.

   (e) Set $LB^k \leftarrow \max\{LB^{k-1}, \sum_i v_i^k(\overline{x})\}$

   (f) Set $F \leftarrow F \cap \{Z \leq \sum_i \omega_i^k + \sum_{i=1..m} \sum_{j=1..n} \lambda_{ij}^k x_{ij}\}$

Note that in the first iteration of the master problem, none of the items will be allocated to any bidder. Further, the algorithm will terminate when the upper bound and lower bound are equal i.e. when we stop generating Benders feasibility cuts since the necessary condition for primal feasibility would have been satisfied.

4.4 Accelerating Convergence of the Benders Algorithm

In every iteration of the Benders algorithm, we essentially solve an integer program while determining the allocation from the master problem solution. Thus, from a computational standpoint, the solution time for the master problem effectively determines the time taken to solve the entire problem. To address this issue of convergence we can use the following strategies - generate a good initial solution, formulate the problem such that the relaxation is tight and generate good cuts to add to the master problem from the sub-problem duals.

When we apply the Benders decomposition algorithm to network optimization problems, it is very important to generate good cuts in each iteration, otherwise the Benders algorithm converges very slowly. The budget constrained optimization problem can be visualized as a bipartite graph assignment problem with additional constraints. The bidders and the items form two disjoint group of nodes while the allocation variable and the budget allocated from each bidder to every item can be represented as links between these group of nodes. The additional constraint, in this case the bidder level budget constraints, needs to be accounted for while solving the bipartite graph assignment problem. It has been been found that problems where the constraint matrix has a network flow structure, the problem typically has many optimal solutions [MW81]. Given the network structure problem, we also observe that there is inherent degeneracy in the sub-problem optimal solutions. In our case, if for a given master allocation, the optimal solution to the sub-problem hits the budget constraint, then we have multiple prices for the allocated items that would give the same revenue to the seller. For example, assume that a bidder is participating in a two-item auction with valuations of {$3,$3} for each item and a budget of $5. If in any iteration, both items are allocated to this bidder, then the bidder's maximum willingness to pay for the package is $5. Individual prices {$2,$3} as well as {$3,$2} both satisfy the budget constraint as well as the individual rationality constraint. Hence, we will have a degenerate optimal solution for the sub-problem corresponding to this bidder.

Magnanti & Wong [MW81] describe a procedure to generate "good" cuts from the sub-problems in Benders decomposition. We use the following definitions from their paper - a cut $Z \leq f(\omega^1) + xg(\lambda^1)$ dominates the cut $Z \leq f(\omega) + xg(\lambda)$, if $f(\omega^1) + xg(\lambda^1) \geq f(\omega) + xg(\lambda)$ for all $x \epsilon X$ with a strict inequality for at least one point $x \epsilon X$, where X is the set of feasible allocations of items to the bidders in our case. In the previous section, Algorithm 4 (point 2(f)) determines the nature of Benders cut. Also, a cut is called a pareto optimal cut if it is not dominated by any other cut. Associated with each cut is a dual set of variables from solving the sub-problems. Thus, if a cut is pareto optimal then the associated dual variable is also said to be pareto optimal.

To generate a pareto optimal cut for the budget constrained auction problem, we use the following theorem from Magnanti & Wong [MW81] -

Theorem 3. *Let x^0 be a point in the relative interior of X^c, where X^c is the convex hull of X. Also, let $U(\hat{x})$ with $\hat{x} \epsilon X$ denote the set of optimal solutions to the optimization problem $Max_{(\omega,\lambda)\epsilon U}\{f(\omega) + \hat{x}g(\lambda)\}$ and let x^0 solve the problem*

$Max_{(\omega,\lambda)\epsilon U(\hat{x})}\{f(\omega) + x^0 g(\lambda)\}$. *Then* x^0 *is pareto optimal.*

In Algorithm 4, we can easily solve the minimization problem independently for each bidder. However, like many other network optimization problems, the solution to the sub-problems are degenerate in nature and thus, the solution method can be accelerated by generating pareto-optimal cuts. Thus, in the step 2(f) of our algorithm, instead of adding the cut specified, we want to add a pareto-optimal cut. Using Theorem 3, the pareto optimal cut is obtained by solving the following linear program -

$$
\begin{aligned}
\min \quad & \sum_{i=1..m} \omega_i + \sum_{i=1..m} \sum_{j=1..n} \lambda_{ij} \overline{x}_{ij} \\
\text{s.t.} \quad & \omega_i + \sum_{j=1..n} \frac{B_i}{V_{ij}} \lambda_{ij} \geq B_i; \text{for } i = 1..m \\
& \sum_{i=1..m} \omega_i + \sum_{i=1..m} \sum_{j=1..n} \lambda_{ij} \overline{x}_{ij} = v(\overline{x}) \\
& \omega_i, \lambda_{ij} \geq 0; \text{for } i = 1..m, j = 1..n
\end{aligned}
$$

Here, $v(\overline{x}) = \sum_i v_i(\overline{x})$.

If we solve the budget constrained problem as described in Algorithm 4, then the initial optimal solution to the master problem could be that none of the items are assigned to any bidder and the objective function of the master problem having a value of Z_{max}, which initially is a very large number (at least as large as the sum of all the bidder budgets). Instead, we could speed up the convergence to the optimal solution for the original problem by using the bid data to find a "good" initial solution. The two algorithms detailed below provide an initial allocation based on a greedy approach.

Algorithm 5 below emulates a sequential auction process. In each iteration, the item under consideration is auctioned off to the bidder who has the largest value, truncated by the remaining budget.

Algorithm 5 (Greedy 1)**.**

1. Let $j \leftarrow 0$, $B_i^{rem} \leftarrow B_i; \forall i$ $(i = 1$ to m) and $x_{ij} \leftarrow 0; \forall i, j$ $(i = 1$ to m, $j = 1$ to n)

2. $j \leftarrow j + 1$

3. For item j (items are selected in the order of their index in the data), find $i' = argmax_i\{min(V_{ij}, B_i^{rem})\}$. Set $p_j \leftarrow max_i\{min(V_{ij}, B_i^{rem})\}$ and $x_{i'j} \leftarrow 1$. Also, set $B_{i'}^{rem} \leftarrow (B_{i'}^{rem} - p_j)$. (In case there is a tie between bidders, then randomly allocate selected item to any of the bidders in the tie set. Alternately, one could allocate to the bidder with the highest remaining budget within the tied set. If there is a tie between bidders with the highest remaining budget, one could randomly allocate the items between the bidders that belong to the tie set.)

4. If $j < m$ then Go To Step 2; m is the number of bidders participating in the auction.

5. STOP

In Algorithm 6, to find an initial solution, instead of considering items in the order that they are indexed, we consider all items and the associated remaining budget with the bidders to make an allocation. We choose to allocate the item that provides the largest value, truncated by the remaining budget to the corresponding bidder.

Algorithm 6 (Greedy 2)**.**

1. Let $k \leftarrow 0$, $B_i^{rem} \leftarrow B_i;$ for $i = 1..m$ and $x_{ij} \leftarrow 0;$ for $i = 1..m, j = 1..n$

2. Find $i', j' = argmax_{i,j}\{min(V_{ij}, B_i^{rem})\}$. Set $p_j = \{min(V_{ij'}, B_{i'}^{rem})\}$, $x_{i',j'} \leftarrow 1$ and $B_{i'}^{rem} \leftarrow B_{i'}^{rem} - p_j$.

3. Remove item j' and associated bids for the item.

4. $k \leftarrow k + 1$

5. If $k < n$ then Go To Step 2.

6. STOP

Let us consider an example where there are two bidders and three items being auctioned of. Let bidder 1 have the valuations of \$3, \$5 and \$2 for items 1, 2 and 3 respectively along with an overall budget constraint of \$7. Similarly bidder 2 has valuations \$4, \$3 and \$6 respectively with a budget of \$7. By algorithm 'Greedy 1', the items will be allocated as follows - Item 1 and 3 to bidder 2 at a price of \$4 and \$3 respectively and item 2 to bidder 1 at \$5 for a total revenue of \$12 for the seller. In case of the 'Greedy 2' algorithm, items 1 and 2 are assigned to bidder 1 at \$2 and \$5 respectively while item 3 is assigned to bidder 2 at a price of \$6. The total revenue for the auctioneer is \$13.

We compared the performance of the various formulations using OPL Studio 3.7, running on machine with 512 MB RAM and an 801 MHz processor. The bidder valuations for various items as well as the budgets were drawn from a uniform distribution. We ran the algorithms for the four different formulations - direct solution using OPL's MIP solver, standard Benders implementation (Algorithm 4), Benders with initialization via the sequential item auction (Algorithm 5) and Benders with initialization with the descending item valuation (Algorithm 6). Each value in Table 4.4 corresponds to an average over ten problem instances. Further, each (bidder, item) problem was run for five different budget levels. The entries marked with a (*) in the table indicate that the algorithm was terminated when the upper and lower bound were within about 2.5% of each other.

Table 4.4: Time to IP optimal (in secs) - Benders

Problem Size	OPL(IP)	Benders (default)	Greedy1	Greedy2
2 bidders, 5 items	0.000	0.031	0.015	0.000
	0.016	0.031	0.016	0.016
	0.016	0.453	0.016	0.000
	0.015	0.031	0.015	0.015
	0.015	0.016	0.016	0.015
5 bidders, 10 items	0.454	10.432	7.272	4.927
	0.078	11.276	4.475	11.65
	0.156	17.609	18.686	19.624
	0.516	22.131	6.662	8.96
	0.093	13.809	14.767	15.615
10 bidders, 20 items	90.672	68.297*	615.6*	219.291*
	1008.25	1315.97*	2503.82*	3277.14*
	165.4	53.65*	426.66*	516.7*
	397.594	393.2*	815.19*	752.87
	0.453	11.124*	74.76*	55.29*

In Table 4.4, we can observe that although the Benders solution with the two heuristics for initial allocation performs well for small problems, as the problem size increases, the Benders implementation with the default starting solution performs better. Hence, the applicability of the two initialization algorithms to the budget-constrained problem is limited. Also, the time taken to solve the problem instances to optimality using OPL Studio is significantly less than that using Benders. Although we could use the pareto optimal cuts as defined by Magnanti & Wong [MW81] in a Benders decomposition framework, we do not expect to match the performance of OPL due to the inherent degeneracy in the problem as well as the significant gap in solution time between OPL and Benders variants, as seen in Table 4.4. The Benders approach, however, provides a nice framework to get feedback on prices for interim allocations as well as provide a method to address the problem in a decentralized manner. The next section details an auction interpretation to the Benders framework.

4.5 An Auction Interpretation to Benders Decomposition

In Benoit & Krishna [BK01], the authors show that when there are two items and two bidders with a budget constraint, then if the items are auctioned of sequentially, it is always optimal to sell the more valuable object first. The paper deals with the case of common value items only. It also does not model cases when more than two objects are present. We know of no study which models the case of the optimal strategy from the sellers perspective when each bidder has individual valuations for the various items. However, it is evident that the sequence in which items are auctioned off will result in different revenues. Given that we do not know the bidder valuations (i.e. they are private), we need to consider all items at a time and not sequentially.

The Benders decomposition method described in this chapter can be considered as an iterative auction mechanism with all items being considered in each iteration. In every iteration, the solution to the master problem is an allocation of a subset items to each bidder. The sub-problems take this allocation and return a set of dual prices for either each allocated item (when the bidder's budget is non-binding) or for the set of allocated items and the bidder budget (when the bidder's budget is a binding constraint). The master problem can, thus, be interpreted as an auctioneer, participating in an interactive format auction, who determines an allocation vector in each round based on feedback that it receives from the bidders for the marginal values of items & package of items allocated. Each sub-problem that we solve for corresponds to a single bidder. The bidders, based on the current allocation, return a set of marginal values to the auctioneer for the allocated item as well as the budget constraint. Thus, the bidders in each iteration solve a pricing problem.

4.6 Conclusion

In this chapter, we model the multi-unit allocation problem with hard budget constraints and additive valuations. We show instances of the problem which are computationally difficult and demonstrate the applicability of Benders decomposition to the problem. The Benders approach provides an approach analogous to an iterative auction model. With the current Benders implementation using OPL Studio 7.0, there doesn't seem to be a computational advantage over the standard MIP implementation. However, the Benders method has a nice auction interpretation that could be used in an iterative manner if bidders do not want to fully reveal valuations over all items and the budget. Specifically in case of the MIP model, the bidders would have to provide valuations for all the items while the Benders decomposition approach would allow them to disclose values for only those items that have been allocated in any iteration.

We could use the cut-generation method specified by Magnanti & Wong [MW81] in a Benders decomposition framework. However, we do not expect it to perform better than the OPL implementation since the problem is inherently degenerate and there exists a significant gap in performance between the OPL implementation and various Benders variants.

Chapter 5

Core Allocation for Budget Constrained Auctions

Multi-unit budget constraint auctions are a popular form of resource allocation and pricing in a variety of settings including procurement auctions, supply chain coordination, search engine marketing and wireless spectrum allocation. The bidders submit their valuations and budgets to the auctioneer, who aims to solve a *social surplus maximization problem* with budget, allocation and individual rationality constraints. The auctioneer seeks to determine an optimal allocation for items with supporting prices that are in the "core". FCC auctions for wireless spectrum allocation and fantasy sports on the Internet are examples of real-world auctions where the entity conducting the auction aims to maximize social value. These problems can be solved using the ascending proxy auction algorithm proposed by Ausubel & Milgrom (2002). However, their approach suffers from convergence problems, lack of feedback on individual item prices, and capping of bidder valuations at budget over an exponential set of packages. In this chapter, we develop a constraint generation procedure for a setting where the bidder valuations for the items are additive. The problem is modeled as a mixed integer program (MIP) with an exponential number of constraints and we develop an approach to determine core allocation and individual item prices directly using the toolkit of integer programming. We, further, extend this approach to model the core determination problem for a general combinatorial auction (i.e., where the bidder valuations are not additive).

5.1 Introduction

The last few years has seen tremendous growth in auctions being used for resource allocation and pricing across a range of business domains. Increased computational ability has resulted in combinatorial auctions being developed for a range of applications that include procurement, supply chain coordination, wireless spectrum allocation, airport slot distribution, real-time electricity markets, communication network pricing and search engine marketing. The academic community has actively contributed to theoretical advancements as well as in developing practical models to address these domains. Even with a fairly restrictive set of assumptions, solving combinatorial auctions to optimality remains a non-trivial computational task. Hence, most academic works ignore the presence of real constraints that all the bidders face e.g. a hard budget constraint.

Ideally, we would like the auction mechanism to have properties like efficiency, incentive compatibility and individual rationality, amongst others. The presence of budget constraints not only makes the problem computationally more difficult, it also destroys some of the economic properties of the unconstrained version of the auction. In the case of auctions with unconstrained bidders, the VCG mechanism gives an incentive compatible allocation and associated prices. However, one of the underlying assumptions of the VCG mechanism is the quasi-linearity of bidder utility. If we have a hard budget constraint, then this assumption is violated as the bidder has a utility of $-\infty$ when the budget constraint is exceeded. One can easily show by a counter-example that the VCG mechanism applied to a budget constrained auction is not incentive compatible.

Borgs et.al. [BCI$^+$05] consider a multi-unit auction having multiple agents with private valuation and hard budget constraints. They show that there is no truthful auction that satisfies the properties of consumer sovereignty, independence of irrelevant alternatives and strong non-bundling. They, then, develop a randomized approach to determine an asymptotically optimal auction (the objective is revenue maximization for the seller). However, their work does not determine if there exists any deterministic allocation method that is incentive compatible, under the presence of budget constraints.

In Benoit & Krishna [BK01], the authors show that when there are two items and two bidders with a budget constraint, then if the items are auctioned of sequentially, it is always optimal to sell the more valuable object first. The paper deals with the case of common value items only and also, does not model cases when more than two objects are present. We know of no study which models the case of optimal strategy from the sellers perspective when each bidder has individual valuations for the various items. However, it is evident that the sequence in which one auctions of the items will result in different revenues for the auctioneer. Thus, we need to consider all items at a time and not sequentially. Che & Gale [YG00] consider the selling of a single item and discuss a non-linear pricing approach for optimal allocation in the presence of budget constraints. Pai & Vohra [PV08] consider hard budget constraints in a single unit auction. They derive revenue maximization auctions for the above setting and show that subsidizing participants with low budgets is never desirable from the auctioneer's standpoint. Dobzinski et.al. [DLN08] prove an impossibility result that states that there does not exist any mechanism in the case of multi-unit auctions that is both incentive compatible and pareto-optimal, when the budgets are privately known.

Given the lack of a mechanism for incentive compatibility under budgetary constraints, we consider a weaker economic criterion, namely a social welfare maximizing allocation and a set of individual item prices that are in the *core*. A core is defined as the set of feasible allocations that cannot be improved upon by a subset or coalition of the participants taking part in the allocation mechanism. The problem that we consider is to determine a core allocation and supporting prices while maximizing social welfare. Social welfare maximization arises in a number of real as well as virtual settings. In the case of many public asset auctions, the objective of the government is to maximize the social surplus. For resource allocation within various departments in a firm, the central authority that takes decisions based on the departmental budgets, aims to allocate resources to maximize social value within the firm. Fantasy sports is an increasingly popular activity in which the platform provider too is interested in overall value maximization. The following sections formulate the core allocation and pricing problem, under budget constraints, using the toolkit of integer programming. We show how to model the core with an exponential set of constraints. Rather than including these core constraints explicitly, we use a constraint generation approach identical to Day & Raghavan [DR07]. We show how to use this constraint generation approach to determine the social welfare maximizing core allocation and prices *simultaneously*.

5.2 Core Allocation for Budget Constrained Auction

The auction that we consider in this chapter is the following - an auctioneer (seller) has n items to sell and there are m bidders bidding for the various items. Each bidder has a certain valuation for each of the items and has her own overall budget limit (assumed to be a hard constraint). As discussed in the previous section, solving to optimality a budget constraint auction, even with additive valuation, is computationally hard. Further, in many real-world settings like fantasy sports or resource allocation within a firm, the valuations are additive. Thus, we initially assume that the bidder valuations are additive. Given the bidder valuations for the various items, the auction operator has to find an allocation and price for each item such that the overall social welfare is maximized. Further, the operator has to ensure that the bidders are not charged beyond their budget limits.

Let B_i be the budget of bidder i and V_{ij} be the value that bidder i has for item j. To prevent bidders from declaring arbitrarily large valuations, we limit the value of a single item by the budget (i.e., $V_{ij} \leq B_i$). The decision variables for the problem are $x_{ij} \in \{0, 1\}$, the allocation of item j to bidder i and p_j, which is the price observed by all bidders for item j. Further, let us assume that there are m bidders and n items. The operator aims to maximize the social surplus. The formulation below allows us to determine individual item prices.

$$
\begin{aligned}
\max \quad & \sum_{i=1..m}\sum_{j=1..n} V_{ij}x_{ij} && (5.1)\\
\text{s.t.} \quad & \sum_{i=1..m} x_{ij} \leq 1; \text{for } j = 1..n && (5.2)\\
& \sum_{j=1..n} p_j x_{ij} \leq B_i; \text{for } i = 1..m && (5.3)\\
& p_j \leq \sum_{i=1..m} V_{ij}x_{ij}; \text{for } j = 1..n && (5.4)\\
& x_{ij} \in \{0,1\}, p_j \geq 0; \text{for } i = 1..m, j = 1..n && (5.5)
\end{aligned}
$$

Equation (5.2) is the availability constraint (no item can be allocated to more than one bidder), equation (5.3) specifies the budget constraint while equation (5.4) corresponds to individual rationality (IR), with respect to the bidder's declared valuations. Note that in the above formulation equation (5.3) is non-linear.

In the above formulation, we could have an optimal solution with all the p_j's set to zero (implying the item prices are zero), since the objective function is $max \sum_i \sum_j V_{ij}x_{ij}$ i.e. it is independent of p_j. Also, none of the constraints necessitate that p_j be strictly greater than zero. Now, if the auctioneer declares the optimal allocation obtained above and the prices of items as zero to the bidders, then each bidder can increase her overall surplus by bidding incrementally more than zero for various items not allocated to her, without violating her own budget constraint. Thus, the losing bidders for each of the items are willing to pay more than zero to the auctioneer to increase their surplus. The seller, too, would be willing to consider such bids and change the current allocation as she can increase the revenue generated from the auction. Hence, the allocation and prices obtained by solving the above formulation is not stable as it provides motivation for both the seller as well as the bidders to collaborate and deviate from the same. For a stable allocation, the formulation has to include a set of constraints such that no coalition of bidders and seller has an incentive to deviate from the allocation at the determined prices (i.e., allocation and prices consistent with our prior informal definition of the *core*). When the solution is in the core, effectively the surplus from the current allocation and prices observed by each bidder should be greater than the surplus generated from all other possible allocations, at the current set of prices.

A novel auction mechanism used for multi-unit allocation is the Ausubel & Milgrom [AM02] ascending auction. If there are no budget constraints, then the Ausubel & Milgrom auction avoids many of the problems associated with the VCG mechanism. Limitations of the VCG mechanism include low revenues for the auctioneer, decreasing revenues as bidders are added to the auction or if bidders change their bids, and collusion being a profitable strategy for the participants.

Let T be an outcome of a combinatorial auction i.e. T corresponds to a set of allocations and payments for the bidders. Also, let the coalition C_T refer to the set of bidders receiving items under the outcome T. To help understand the method to determine a core allocation and prices for an auction under budget constraint, we will use the following definitions from Day & Raghavan [DR07] -

1. An outcome T is considered *blocked* if there exists an alternative outcome T_B that generates more revenue for the seller and for which each bidder in C_{T_B} weakly prefers to T. C_{T_B} is referred to as a *blocking coalition*
2. An outcome that is not blocked is called a *core outcome.*

The Ausubel & Milgrom[AM02] ascending proxy auction works roughly as follows. The bidders report their values for the various packages that they are interested in to the proxy. The bidders could also specify their overall budget constraint to the proxy agent. In every iteration, the auctioneer specifies a set of prices for the various packages. The proxy agent for a bidder takes these prices and figures out the package that will maximize the bidder's surplus, at the current price levels. The agent then bids for that package in the auction. In every iteration the auctioneer determines the revenue maximizing allocation and the prices for the over-demanded packages are increased incrementally. The auction terminates when there are no new bids from the proxy agents. It is important to note that all the intermediate bids are kept live throughout the auction. Ausubel & Milgrom [AM02] prove the following theorem -

Theorem 4. *The payoff vector π resulting from the proxy auction is a core imputation relative to the reported preferences: $\pi \in Core(L, w)$. Here, L is the set of players (includes seller) while $w(S)$ is the value of coalition S.*

Due to convergence to the core, the seller gets competitive revenues while the buyer allocation is efficient. Also, to note is that in every round of the auction the objective function being maximized is the seller revenues and the overall buyer surplus i.e. in every iteration, social welfare is being maximized. An issue with the Ausubel & Milgrom (AM) auction is convergence. Since, in each iteration prices are changed by a very small amount, the auction is plagued by a slow convergence to the core. The AM auction can be modified to solve core allocation problems with budget constraints. The modification proposed is to put a cap on the bidder valuation for a subset of items at the bidder specified budget. E.g.- if a bidder has a valuation of $5 for item 1, $4 for item 2 and a budget of $7, then the bidder valuation is assumed to be $5 for item 1, $4 for item 2 and $7 for items 1 & 2. However, this modification creates two issues - artificial capping of bidder valuations that makes the auctioneer ignore true item valuations of the respective bidders and a pre-processing step to the actual algorithm that necessitates determining valuations over an exponential set of packages. Continuing with the example above, let us assume that there is a second bidder that values item 1 and item 2 at $7 but has no value for each individual item. The budget for bidder 2 is assumed to be $7 (same as bidder 1). Based on the approach in the AM paper for budget constraints, the auctioneer in this case would assume that both the bidders have the same valuation for the package containing items 1 & 2, although bidder 1 has declared a strictly higher valuation. If the auctioneer had a choice between allocating both items to either bidder 1 or bidder 2, the valuation capping approach would make the auctioneer indifferent between the two bidders. The auctioneer could have achieved higher efficiency by allocating both items to bidder 1.

An argument that has been sometimes made is that if the bidder really values a subset of items higher than the budget, then she should be specifying a higher willingness to pay. However, in the real-world, valuations and budgets don't necessarily go together. E.g. a telecommunications service provider may value two cities in the US at $2 billion and $1 billion respectively. The valuation is calculated based on business potential that the firm estimates for each city. The budget, however, is determined by the ability of the firm to raise funds from various stakeholders and thus, is limited by factors that include financial market conditions and the firm's past performance. Thus, the bidder may be able to raise no more than $2.5 billion for the auction.

To understand the issue with the pre-processing step prior to running the AM auction, let us assume there are 50 items being auctioned off and that the bidders have additive valuations over this set of items. We will have to determine the value of each subset of items where the value of a subset is the sum of values for the individual items in the subset, if the sum is less than the budget. If the total value of the items in the subset exceeds the budget, then the value is capped at the bidder's budget. With 50 items, we will have to evaluate nearly 10^{15} bundles. Hence, even if we assume that truncation of values at budget is acceptable, we cannot use the AM algorithm to solve a package auction of this size.

Further, the AM ascending auction mechanism fails to provide feedback on individual item prices, even when the valuation is additive and is specified for each individual item by the bidder.

Although the work of Day & Raghavan [DR07] and Hoffman et.al. [HMvdHW06] discuss implementations for faster convergence to the core, both the papers ignore budget constraints. The allocation is the efficient allocation for the given bids and the core prices are calculated for this allocation. Further, Day & Raghavan [DR07] show that it is a Nash equilibrium to bid truthfully. However, in the case of budget constrained auction, the core allocations need not be the social welfare maximizing allocation obtained by ignoring budget constraints; and thus, we have to determine *both the allocation and prices simultaneously* that satisfy the various constraints. In the following section, we formulate the pricing and allocation problem with core constraints as a mixed-integer program. We demonstrate that for the additive valuation case, we can determine the core allocation and prices, if they exist, by a constraint generation approach. In fact, our approach is able to solve the core allocation and pricing problem under budget constraints with individual item price feedback. We, also, give examples of budget constrained auctions under additive valuation that will not have a core solution with individual item prices. In such cases, we can use our proposed method with a modification that has package prices instead of item prices.

5.3 Core Problem Formulation as a MIP

Borgs et.al. [BCI$^+$05] describe the lack of any deterministic incentive compatible mechanism currently known for budget constrained auctions. In the case of auctions run by public sector organizations like the FCC as well as in the case of combinatorial exchanges jointly operated by bidders and sellers, the objective of the entity running the auction is to maximize the social welfare. Similar situations of social welfare maximization is also found in online environments dealing with "funny money" e.g. fantasy sports. In this section, we formulate the social welfare maximization problem as an integer program. The auction operator is trying to maximize social surplus for all the participants concerned. In the formulation of Andelman & Mansour [AM04], they considered the problem from the auctioneer's standpoint i.e. the auctioneer is interested in revenue maximization. Thus, the optimal solution obtained would be such that, after the bids and budgets are submitted, the auctioneer would have no incentive to deviate from the determined allocation and prices since the objective function was revenue maximization for the auctioneer.

Instead, let us assume that the auctioneer and bidders both have agreed to an exchange that they believe is "fair". The auction operator doesn't favor either party but is interested in obtaining an allocation and corresponding set of non-discriminatory prices that all participants accept and don't have an incentive to deviate from. Thus, we seek to determine an allocation and set of prices that are in the core; i.e., given the bids and budgets, there will be no losing bidders that could have bid more than the current set of prices paid by the winners, given their declared bids and budgets. Note that in the analysis below, the item valuations for all bidders are assumed to be additive.

The following example will clarify the need and motivation for the core formulation described later in the chapter. Let us consider a seller auctioning off two items A and B. Also, assume that there are two bidders participating in the auction. Bidder 1 values each item at \$5 and has an overall budget of \$7. Bidder 2 values the two items at \$4 each and has an identical budget constraint as bidder 1. If the auctioneer is interested in revenue maximization, then we can use the model described in the previous chapter or that in Andelman & Mansour [AM04] to determine the optimal allocation and associated prices. In the revenue maximization case, item A will be allocated to bidder 1 at \$5 and item B will be allocated to bidder 2 at \$4. The total revenue that the seller receives is \$9. Further, there is no other feasible allocation that is individually rational to the bidders and will result in higher revenues to the seller. Hence, the seller has no incentive to collaborate with any bidder and change allocation or prices. The allocation is, thus, in the core.

Let us now consider the case where the auctioneer is performing social welfare maximization instead of revenue maximization. Under social value maximization, both items A and B will be allocated to bidder 1, at a combined price of \$7, since the budget constraint cannot be violated. Assume that the prices declared are \$3.5 and \$3.5 respectively. Bidder 2, who is currently getting zero surplus as she is not assigned any item, can increase her surplus by bidding $\$3.5 + \delta$ on item B. Also, the seller can increase her revenue from \$7 to $\$7 + \delta$ by accepting the offer of bidder 2. Thus, the price and allocation determined by solving the social welfare maximization problem is not stable, unlike that for the revenue maximization solution considered previously.

If instead bidder 1 is allocated item A and bidder 2 is allocated item B, one should note that the prices \$3.5 and \$3.5 (for items A and B respectively) are in the core, since neither bidder can bid more on the respective losing items without violating one's own budget constraint. Also, to note is that there are other prices that support the core allocation. For example, the same allocation as above along with prices of \$3.5 and \$4 for the respective items is also in the core. However, this core point is not bidder-pareto optimal. An outcome is defined as bidder-pareto optimal if there is no other outcome in the core weakly preferred by every bidder in C_T (Day & Raghavan [DR07]). However, it be clear that bidder 2 prefers being assigned item B at a price of \$3.5 rather than \$4. In the formulation that we describe later in the chapter, we formulate the problem to determine the core solution and discuss an approach by which our formulation can be modified suitably to determine a bidder-pareto optimal outcome.

Let p_j be the price of item j and $x_{ij} \in \{0, 1\}$ indicates if item j is allocated to bidder i. Let us assume that the value (in our case, the bid) of item j to bidder i is V_{ij} and the budget for bidder i is B_i. The problem that the auction operator solves to determine the core allocation and prices for a budget constraint auction with additive valuation and an objective of social surplus maximization can be formulated as

$$\max \quad \sum_{i=1..m}\sum_{j=1..n} p_j x_{ij} + \sum_{i=1..m}\sum_{j=1..n} (V_{ij} - p_j) x_{ij} \tag{5.6}$$

$$\text{s.t.} \quad \sum_{i=1..m} x_{ij} \leq 1; \text{for } j = 1..n \tag{5.7}$$

$$\sum_{j=1..n} p_j x_{ij} \leq B_i; \text{for } i = 1..m \tag{5.8}$$

$$p_j \leq \sum_{i=1..m} V_{ij} x_{ij}; \text{for } j = 1..n \tag{5.9}$$

$$\sum_{j=1..n} (V_{ij} - p_j) x_{ij} \geq max\{\sum_{j\in S}(V_{ij} - p_j), \sum_{j\in S} V_{ij} - B_i\}; \quad \text{for } i = 1..m, S \subseteq J \tag{5.10}$$

$$x_{ij} \in \{0, 1\}, p_j \geq 0; \text{for } i = 1..m, j = 1..n \tag{5.11}$$

The objective function consists of two parts - the first part corresponds to the seller revenue while the second is the total buyer surplus. The seller is interested in maximizing her own revenues while each bidders wants to maximize her respective surplus. For an allocation and price to be in the core, the surplus to any bidder for the current allocation and prices should be at least as large as that generated by any subset of items at the current price level. This is not a requirement of the core definition in Day & Raghavan [DR07] but we can determine a set of supporting Walrasian equilibrium item prices by incorporating equation (5.10) in our model. In fact, the equilibrium item prices will be a subset of the core supporting price vector. Equations (5.7), (5.8) and (5.9) are identical to equations (5.2), (5.3) and (5.4) respectively. The constraint set (5.10) has the following interpretation - the surplus from the allocation at the current price levels is higher than the surplus generated by any other allocation, where the price paid for any allocation is capped at the declared bidder budget. The chosen subsets of J do not violate the bidder's budget constraint; i.e., if the sum of the individual prices is higher than the bidder's budget, then the payment for the items is assumed to be capped at the declared budget.

These additional constraints result in an exponential number of constraints being added to the social welfare maximization problem defined by equations (5.6)-(5.9). Also, one should note that constraints (5.8) and (5.10) are non-linear.

We linearize equations (5.6)-(5.11) by introducing the variable t_{ij}, where t_{ij} is the fraction of the budget of bidder i allocated to item j. A MIP corresponding to equations (5.6)-(5.11) is then

$$\max \quad \sum_{i=1..m} \sum_{j=1..n} V_{ij} x_{ij} \tag{5.12}$$

$$\text{s.t.} \quad \sum_{i=1..m} x_{ij} \leq 1; \text{for } j = 1..n \tag{5.13}$$

$$\sum_{j=1..n} t_{ij} \leq 1; \text{for } i = 1..m \tag{5.14}$$

$$t_{ij} \leq x_{ij}; \text{for } i = 1..m, j = 1..n \tag{5.15}$$

$$\sum_{i=1..m} B_i t_{ij} \leq \sum_{i=1..m} V_{ij} x_{ij}; \text{for } j = 1..n \tag{5.16}$$

$$\sum_{j=1..n} (V_{ij} x_{ij} - B_i t_{ij}) \geq \sum_{j \in S} (V_{ij} - \sum_{i=1..m} B_i t_{ij}); \quad \text{for } i = 1..m, S \subseteq J \tag{5.17}$$

$$\sum_{j=1..n} (V_{ij} x_{ij} - B_i t_{ij}) \geq \sum_{j \in S} (V_{ij}) - B_i; \text{for } i = 1..m, S \subseteq J \tag{5.18}$$

$$x_{ij} \in \{0, 1\}, t_{ij} \geq 0; \text{for } i = 1..m, j = 1..n \tag{5.19}$$

Constraint (5.13) is the item allocation constraint. Constraint (5.14) dictates the fractional budgetary allocations for a bidder will not exceed one. Constraint (5.15) indicates that no part of a bidder's budget will be allocated to an item, unless that item is assigned to the bidder. The individual rationality for the allocated items is ensured by constraint (5.16). Constraints (5.17) and (5.18) together define the core constraints, where the payments for the current allocation are capped by the bidder's budget.

Note that we can modify the objective function (5.12) to obtain a bidder-pareto core solution rather than any general solution in the core. The new objective function to determine the core allocation will be $\sum_{i=1..m} \sum_{j=1..n} V_{ij}x_{ij} - \alpha \sum_{j=1..n} (\sum_{i=1..m} B_i t_{ij})$. Here, α is a constant that is assigned a very small value. Due to the presence of the term $\alpha \sum_{j=1..n} (\sum_{i=1..m} B_i t_{ij})$ in the objective function, the value maximizing core solution will correspond to a set of prices from the bidders that maximize the difference between the first and second terms in the new objective function. This can be re-interpreted as a value maximizing core solution where the second term in the new objective function is minimized. The second term is nothing but the sum of prices paid by the bidders multiplied by a constant, α. Hence, this formulation results in an optimal core allocation that is supported by a set of prices such that the sum of the prices for the items is minimized. In the next section, we describe a constraint generation approach to deal with the exponential number of constraints in a computationally efficient manner.

5.4 Constraint Generation Approach to Determine Core

The core allocation and pricing problem for the budget constrained auction problem can be solved by the ascending proxy auction of Ausubel & Milgrom [AM02]. The AM auction makes a strong assumption that the bidder valuations for packages are capped at the respective budgets. Further, the auction suffers from multiple issues - slow rate of convergence, the core solution being dependent on the step size for incrementing the price on over-demanded items, lack of feedback on individual item prices in case of additive valuation and the need to evaluate package valuations on all possible packages, even when the valuations are additive. If we directly solve the formulation described in the previous section, we avoid the limitations of the Ausubel & Milgrom auction. However, in our case, the core is defined by an exponential number of constraints and thus, we need to determine an approach that addresses associated computational issues.

We can characterize the core allocation and price as follows - if the current prices and allocation are in the core, then no bidder can increase her own surplus by bidding more than the current winner on any item not allocated to her without violating her declared budget constraint. To deal with the exponential number of constraints that define this core, we seek to separate out the set of core defining constraints that don't have any effect on the optimal solution from those that define the optimal allocation and prices. We use an approach identical to that in Day & Raghavan [DR07] to generate appropriate constraints to define core. In their work, the authors consider a standard multi-unit auction with no budget constraint. Thus, they start of with an efficient allocation for the stated valuations and then determine a set of supporting prices that are in the core. However, in case of budget constrained auctions, it is not necessarily the case that the allocation that maximizes value will be a core allocation. Thus, we determine both the core allocation and prices simultaneously.

We start with solving the linearized version of equations (5.1)-(5.5), i.e. equations (5.12)-(5.16). If the prices and allocations do not violate any of the core constraints (i.e. equations (5.17) or (5.18)), then we have arrived at a core point. However, to verify whether a core constraint is being violated for a bidder, we need to compare the surplus at current price levels with an exponential subset of item groups. In other words, we need to check the surplus at the current price levels for each bidder over all possible set of items, S, where $S \subseteq J$ (here, J denotes the set of all packages that can be created from n items). We show how this can be achieved by solving the separation problem $SEP(i)$, described below, to find the most violated constraint corresponding to equation (5.17) for a bidder i. If the objective function for $SEP(i)$ has a value less than or equal to zero, then no core constraint is being violated for the bidder at the current allocation and price levels. However, if the objective value is greater than zero, then we need to find the set of items that result in the constraint being violated. The price of these items cannot exceed the bidder's budget. If the price exceeds the budget, we add constraint (5.18) instead, since the right side of constraint (5.18) is now greater than that for (5.17). Steps 4 and 5 in Algorithm 6 provide the details to find the most violated constraint for a given bidder, at a particular price level and allocation.

SEP(i)

$$Z^i_{sep} = \max \quad \sum_{j=1..n} (V_{ij} - \sum_{i=1..m} B_i \bar{t}_{ij}) y_j - \sum_{j=1..n} (V_{ij}\overline{x}_{ij} - B_i \bar{t}_{ij}) \tag{5.20}$$

$$\text{s.t.} \quad \sum_{j=1..n} (\sum_{i=1..m} B_i \bar{t}_{ij}) y_j \leq B_i \tag{5.21}$$

$$y_j \in \{0,1\}; \forall j = 1..n \tag{5.22}$$

Note that $\overline{x}_{ij}$ and $\bar{t}_{ij}$ are given for problem $SEP(i)$. Also, let Z^i_{SEP} denote the value of the objective function for the problem $SEP(i)$. Solving the separation problem is essentially solving a knapsack problem for every bidder to determine the most violated constraint. For each bidder, we determine the most violated constraint, add it to the formulation comprising equations (5.12)-(5.16) (including the constraints added in all previous iterations) and resolve the problem. The stopping criterion for this algorithm is when we can find no Walrasian equilibrium prices that are being violated by current allocation. Algorithm 7 describes in detail the solution approach.

Algorithm 7 (Algorithm to find Core)**.**

1. Let $F \leftarrow \{x_{ij} \in \{0,1\}, t_{ij} \geq 0; \forall i = 1..m, j = 1..n\}$

2. Solve (5.12)-(5.16)

$$\begin{aligned}
\max \quad & \sum_{i=1..m} \sum_{j=1..n} V_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{i=1..m} x_{ij} \leq 1; \forall j = 1..n \\
& \sum_{j=1..n} t_{ij} \leq 1; \forall i = 1..m \\
& t_{ij} \leq x_{ij}; \forall i = 1..m, j = 1..n \\
& \sum_{i=1..m} B_i t_{ij} \leq \sum_{i=1..m} V_{ij} x_{ij}; \forall j = 1..n \\
& (x, t) \in F
\end{aligned}$$

   Denote the optimal solution by $(\bar{x}, \bar{t})$

3. Initialize $S_i \leftarrow \{\}; \forall i$

4. Perform

   for each i

   Set $Pass(i) \leftarrow 0$

   Solve $SEP(i)$ and denote by y_j^* the optimal solution to $SEP(i)$

   if $Z_{sep}^i > 0$

   $S_i = \{j | y_j^* = 1\}$

   $F \leftarrow F \cap \{ \sum_{j=1..n} (V_{ij} x_{ij} - B_i t_{ij}) \geq \sum_{j \in S_i} (V_{ij} - \sum_{i=1..m} B_i t_{ij})\}$

   else $Pass(i) \leftarrow 1$

5. Check

if $Pass(i) = 1$ for each i

DONE

else

Go To Step 2

An important point of note is that the a core allocation and supporting set of individual item prices (i.e. Walrasian equilibrium prices) may not exist for certain bidder valuations and budgets. The algorithm that we described above finds the core with individual item prices, if it exists. In case, there is no such supporting set of prices, then the algorithm detects infeasibility and will terminate without determining the core. In such a case, we can use the constraint generation approach similar to that described in Algorithm 7 to find a core allocation with supporting package prices. We will need to modify step 2 in Algorithm 7 to add constraints that account for multiple allocated packages not having the same items (we will describe this procedure in section 5.6).

The following example demonstrates a two bidder case where a core allocation with individual item prices does not exist. Let us assume that there are four items that the bidders declare their valuation for. Bidder 1's valuation for the items is {19, 5, 18, 18} while bidder 2 has a valuation of {19, 16, 18, 17}. Further, the two bidders have a budget of \$21 and \$22 respectively. In the first iteration, items 1, 2 and 3 will be allocated to bidder 2 while item 4 will be assigned to bidder 1. Each item will be be priced at zero. At the current set of prices and allocation, the most violated constraint for both bidders would involve all the four items. Thus, the most violated constraint for the bidders will be $19x_{11} + 5x_{12} + 18x_{13} + 18x_{14} \geq 60 - 22(t_{21} + t_{22} + t_{23} + t_{24})$ and $19x_{21} + 16x_{22} + 18x_{23} + 17x_{24} \geq 70 - 21(t_{11} + t_{12} + t_{13} + t_{14})$ respectively.

Note that the left hand side of both these constraints has to be greater than 38 and 49 respectively, irrespective of what allocation and prices we arrive at. This is, however, not possible since it requires that at least three of the x_{ij}'s for each bidder equate to 1. Thus, there is no feasible allocation that can support these two constraints.

An issue which we have not addressed in this chapter is the incentive of the bidders to declare bids or budgets that deviate from their true values. Borgs et.al. [BCI$^+$05] describe the lack of any known incentive compatible mechanism for the multi-unit auction setting with budget constraints. As there are no known incentive compatible mechanisms, we focused on determining core allocation and associated prices assuming that the bidders have correctly stated their values for individual items and overall budget. However, the bidders could shade their bids or budgets and increase their surplus.

To limit the ability of bidders to substantially shade their values or budgets, we require that the budget stated by a bidder has to be greater than the maximum value that the bidder has for any individual item. This ensures that bidders who, for example, state very high bids for individual items cannot shade their budgets significantly and still be allocated items by the auctioneer. In fact, by shading the budget, the bidder will be reducing the possible number of items that the auctioneer considers for allocation to the bidder. Thus, in the presence of competition, the bidder negatively affects her own probability of winning items from which she could have derived a positive utility, if she had stated her true budget. However, if there only a few bidders participating in the auction, the bidders may have an incentive to shade their budgets.

An alternate approach to determine a core solution would be to use the simultaneous ascending auction described by Cramton [Cra06]. Simultaneous ascending auctions are ideally suited for settings where the complementarities between the individual items is weak. All the items are auctioned off simultaneously and the bidders have the flexibility to bid on any set of items. The auction terminates when none of the bidders raise the bid on any of the items being auctioned off. In the additive valuation case, the simultaneous ascending auction does not have any exposure problems since there are no complementarities between the items. Also, the auction provides the bidders an opportunity to discover item prices over multiple rounds. A key advantage of this auction format is its simplicity.

The simultaneous ascending auction format converges to a core allocation when the individual items are substitutes, bidders are price takers and the bid increments in each auction round are infinitesimally small. In the additive setting that we have described so far only one of the conditions i.e., items being substitutes, holds true. In the computational experiments that we describe in the next section, we have cases where the competition is limited and thus, bidders are not price takers. In practice, bid increments ranging from 5 to 20% are used. An advantage of the simultaneous ascending auction is that the bidders do not have to state their budget to the auctioneer. In fact, a variant of this format called the clock auction precludes the bidders from declaring their bids to the auctioneer (the bidders only express the desired quantities at the price denoted by the clocks associated with each item). In the clock auction format the bidders which items they are still bidding for in each round. A limitation of the clock auction is that under budget constraints current prices of item A can affect the decision to bid for item B. For e.g., assume a bidder has valuations of $5 each for items A and B and a budget constraint of $8. If the current prices for each of the items goes over $4, then the bidder will stop bidding for one of the two items. Our model does not face this issue.

Three key issues with using the simultaneous ascending auction format are demand reduction, collusive bidding and convergence issues related to small bid increments. In case of limited bidder competition, the auctioneer revenues as well as auction efficiency can be significantly affected due to bid shading by bidders with higher valuations [Cra06]. Collusion between bidders in FCC auctions has also been observed in practice. The auctioneer can modify the auction rules to reduce collusion and encourage competition. Cramton [Cra06] suggests methods that include concealing bidder identities, setting high reserve prices and offering preference for small businesses. The core allocation approach that we have proposed has incentive issues similar to that observed in simultaneous ascending auctions. However, like the simultaneous ascending auctions, incentive issues are of less concern to us if there is substantial bidder competition. Our method, in fact, addresses a key issue related to auction convergence faced in the simultaneous auction setting since our final allocation and prices are based on a one-time set of inputs from bidders regarding their individual item values and overall budget. Further, our model considers all possible core solutions while the choice of bid increments prevents the ascending auction format from doing so.

5.5 Computational Experiments

We ran the core allocation and price determination algorithm for the additive valuation case. The algorithm was run on various problem instances where we varied three parameters - the number of bidders, the number of items being auctioned off and the bidder budgets. The bidder valuations for each item were assumed to be drawn from a uniform distribution, $U[0, k]$. The bidder budgets were also drawn from a uniform distribution. The budget for a bidder is a value drawn from the distribution $U[k, k + p]$. Here, p is a parameter that we will vary to observe the computation time as a function of the bidder budget. We specify four levels for p where level 1 corresponds to the highest budget level and level 4 is the lowest budget level (i.e. tightest budget constraint). Also, the value for p at each budget level is half the value of p at the previous budget level. For example, if level 1 corresponds to a budget drawn from $U[k, k + p]$ then budgets for bidders corresponding to level 2 will be drawn from $U[k, k + (p/2)]$. The values for p were selected from the list $[kn, kn/2, kn/4, kn/8]$, where k is the number used above in the uniform distribution and n is the number of items being auctioned off.

The algorithm was run using GLPK 4.43 & libraries that allow calls to the GLPK routine using Python. We used PyGLPK and PyMathProg to write the algorithm in Python and make appropriate calls to GLPK solver. The code was run on a 1 GB RAM, 1 GHz Intel Centrino machine with Linux OS. The results are shown in Table 5.1.

Note that the entry *NFCS* stands for the problem instances where we found 'No Feasible Core Solution'. Each cell in the table indicates the average computation time across ten problem instances. The results indicate that our method can determine the existence of core solution as well as core allocation and prices in a reasonable time frame. Further, we observe that tighter budget constraints typically result in higher computation time, for a given problem size.

Table 5.1: Computation Time for Core Allocation & Prices (Algorithm 7) (in secs)

		Budget Level			
Bidders	Items	1	2	3	4
2	2	0.0281	0.0305	0.0398	0.0485
	5	0.4038	0.9050	1.0256	1.1350
	10	1.7846	5.0212	5.9336	9.8690
5	2	0.0630	0.0844	0.0868	0.0776
	5	0.7063	1.0289	0.9680	0.8325
	10	7.5686	8.3460	8.4360	NFCS
10	2	0.2034	0.3124	0.2720	0.3361
	5	2.2700	2.1900	1.4396	1.9960
	10	13.661	14.4236	16.2740	16.6422
20	2	0.8764	0.9340	1.0332	1.1093
	5	4.8166	5.5278	4.8360	6.4280
	10	27.6780	28.2726	31.3874	32.7579
50	2	3.484	4.9486	3.915	4.0079
	5	22.746	26.655	28.556	32.6245
	10	115.306	122.24	133.578	238.25

5.6 Core Formulation for General Package Auctions

In this section, we generalize the core pricing and allocation problem for a combinatorial auction where no assumptions are made on the bidder valuations (i.e., the bidder valuations need not be additive). Similar to the approach in Section 5.4, we develop a non-linear formulation to define the core and then reformulate it as a MIP.

Let i be the bidder index varying from 1 to m and j be the item index varying from 1 to n. Also, let N be the set of all packages that can be created from the items and S denote a specific package. Note that the auctioneer has a single copy of each item and the bidders use the OR bidding language to express their preferences over the packages. A bidder can be allocated more than one package and the budget of a bidder has to be greater than the bidder's maximum value over any package that the bidder is interested in. We specify this requirement on the bidder budget so that the there is a lower bound on how much the bidder can shade her budget. Further, for any bidder that shades the budget a lot, lesser number of packages will be considered by the auctioneer for possible allocation to her. Thus, the bidder will have to take this trade-off between budget shading and lower probability of being allocated various packages into consideration. The inputs to the optimization problem are $V_i(S)$ and B_i, where $V_i(S)$ is the value (bid) of bidder i for package S and B_i is the budget declared by bidder i. The decision variables for the problem are $p(S)$, the price of package S and $y_i(S) \in \{0, 1\}$, where $y_i(S)$ is 1 if the bidder i is allocated package S and is zero otherwise. A bidder can be allocated any number of packages that she bids on. The core allocation and pricing problem for a general budget constrained combinatorial auction is described below.

$$\max \quad \sum_{i=1..m} \sum_{S \subseteq N} V_i(S) y_i(S) \tag{5.23}$$

$$\text{s.t.} \quad \sum_{i=1..m} y_i(S) \leq 1; \forall S \subseteq N \tag{5.24}$$

$$\sum_{S \ni j} \sum_{i=1..m} y_i(S) \leq 1; \text{for } j = 1..n \tag{5.25}$$

$$p(S) \leq \sum_{i=1..m} V_i(S) y_i(S) + M(1 - \sum_{i=1..m} y_i(S)); \forall S \subseteq N \tag{5.26}$$

$$\sum_{S \subseteq N} p(S) y_i(S) \leq B_i; \text{for } i = 1..m \tag{5.27}$$

$$\sum_{S \subseteq N} [V_i(S) - p(S)] y_i(S) \geq \max\{\sum_{S \subseteq N} (V_i(S) - p(S)), \sum_{S \subseteq N} V_i(S) - B_i\}; \text{for } i = 1..m \tag{5.28}$$

$$y_i(S) \in \{0, 1\}, p(S) \geq 0; \text{for } i = 1..m, \forall S \subseteq N \tag{5.29}$$

The objective function is maximizing social surplus. Constraint(5.24) and (5.25) are feasible allocation constraints. Constraint (5.24) indicates that no package can be allocated to more than one bidder and (5.25) corresponds to feasibility of items over the allocated set of packages. Constraint (5.26) corresponds to the set of individual rationality constraints with respect to the declared bidder valuations. The constant M is a very large value and we can use the maximum bids and budgets disclosed by the bidders to arrive at an appropriate value for M. We use the big M method in constraint (5.26) because we do not want unallocated packages to have their price constrained i.e., unallocated packages need not have zero core prices. The budget constraint for each bidder is imposed by constraint set (5.27). Constraint (5.28) defines the core and specifies that the surplus generated by the allocated package to the bidder at a price $p(S)$ is at least as large as the surplus generated by any other package at the current set of prices. Note that the constraint sets (5.27) and (5.28) are of a non-linear nature.

To linearize the above formulation, we define a variable $t_i(S)$, where $t_i(S)$ is the fraction of bidder $i's$ budget used for obtaining the package S. The MIP formulation for the previous formulation is -

$$\max \quad \sum_{i=1..m} \sum_{S \subseteq N} V_i(S) y_i(S) \tag{5.30}$$

$$\text{s.t.} \quad \sum_{i=1..m} y_i(S) \leq 1; \forall S \subseteq N \tag{5.31}$$

$$\sum_{S \ni j} \sum_{i=1..m} y_i(S) \leq 1; \text{for } j = 1..n \tag{5.32}$$

$$\sum_{i=1..m} B_i t_i(S) \leq \sum_{i=1..m} V_i(S) y_i(S); \forall S \subseteq N \tag{5.33}$$

$$t_i(S) \leq y_i(S); \text{for } i = 1..m, \forall S \subseteq N \tag{5.34}$$

$$\sum_{S \subseteq N} (V_i(S) y_i(S) - B_i t_i(S)) \geq \sum_{S \subseteq N} (V_i(S) - p(S)); \quad \text{for } i = 1..m \tag{5.35}$$

$$\sum_{S \subseteq N} (V_i(S) y_i(S) - B_i t_i(S)) \geq \sum_{S \subseteq N} (V_i(S)) - B_i; \quad \text{for } i = 1..m \tag{5.36}$$

$$p(S) \leq \sum_{i=1..m} V_i(S) y_i(S) + M(1 - \sum_{i=1..m} y_i(S)); \forall S \subseteq N \tag{5.37}$$

$$\sum_{S \subseteq N} t_i(S) \leq 1; \text{for } i = 1..m \tag{5.38}$$

$$p(S) \geq \sum_{i=1..m} B_i t_i(S); \forall S \subseteq N \tag{5.39}$$

$$y_i(S) \in \{0, 1\}, p(S) \geq 0, t_i(S) \geq 0; \text{for } i = 1..m, \forall S \subseteq N \tag{5.40}$$

We will use a constraint generation approach to specify the core constraints. Hence, we solve for equations (5.30)-(5.34) and (5.37)-(5.40) and check if the optimal allocation and price violate any of the constraints specified by (5.35) or (5.36). If none of the constraints are being violated, then we have determined the optimal core allocation and package prices. If, however, any of the constraints in (5.35) or (5.36) are being violated, then we need to determine the most violated one for each bidder. This can be obtained by solving the separation problem $SEP2(i)$, for each bidder, given below.

SEP2(i)

$$
\begin{aligned}
Z^i_{sep2} = \max \quad & \sum_{S \subseteq N} [V_i(S) - \overline{p}(S)]x(S) \\
& - \sum_{S \subseteq N} [V_i(S)\overline{y_i}(S) - B_i\overline{t_i}(S)] && (5.41) \\
\text{s.t.} \quad & \sum_{S \subseteq N} (\sum_{i=1..m} B_i.\overline{t_i}(S))x(S) \leq B_i && (5.42) \\
& x(S) \in \{0,1\}; \forall S \subseteq N && (5.43)
\end{aligned}
$$

If the value for equation (5.41) is greater than zero, then the $\hat{S}$ for which $x(\hat{S}) = 1$ is the package corresponding to the most violated constraint for the specific bidder. Thus, we need to add the constraint $\sum_{S \subseteq M} V_i(S)y_i(S) - p(S) \geq V_i(\hat{S}) - \sum_{i=1..m} B_i t_i(\hat{S})$ to (5.30)-(5.34) and (5.37)-(5.40), and resolve the problem.

From a computational perspective, the amount of computation required to find the most violated constraint in each iteration is equivalent to finding the package which gives the maximum surplus to a bidder in any iteration of the Ausubel-Milgrom auction. However, since our solution is not dependent on any step size for the item prices, unlike the Ausubel & Milgrom auction, we consider all possible core points and will not face convergence issues related to the choice of step size. Thus, from a practical standpoint, our constraint generation approach can be applied to any combinatorial auction problem where the Ausubel & Milgrom ascending auction is thought of as a feasible format.

A few researchers have proposed that in the case of package auctions a budget constrained core allocation can be obtained by a minor modification to the Ausubel & Milgrom [AM02] algorithm or the method proposed by Day & Raghavan [DR07]. The modification that the researchers propose is to truncate the valuations at the bidder budget for each package that the bidders are interested in. The core allocation algorithms are run on these packages with truncated values. It is, thus, assumed that the increased computational complexity of the approach described earlier in this section is not warranted. In section 5.2, we showed an example where there exists an optimal allocation that does not maximize social welfare when we truncate at bidder budgets and apply the AM algorithm. This occurs due to the information loss that is inherent when the valuations are truncated. The method that we propose always takes the true valuation (assuming bidders bid truthfully) of the bidders into account and thus, would determine the allocation within core that maximizes social welfare, i.e., an efficiency maximizing allocation within the core.

5.7 Conclusion

In this chapter, we have demonstrated a MIP based approach to determine individual item level core allocation and prices under hard budget constraints, when the bidder valuations are additive. We provide a computational approach that maintains *allocation feasibility* throughout. This is unlike the Ausubel & Milgrom auction, where none of the interim allocations are feasible. Also, we provide an approach to deal with the practical impediments in choosing appropriate step size for the same. Further, we describe a method to determine the bidder-pareto optimal outcome by refining the core determination formulation.

We extend this approach to combinatorial auctions with general bidder valuations and show that finding violated constraints in each iteration is computationally equivalent to finding the maximum surplus package determination by the Ausubel & Milgrom ascending auction. We have run computational tests to provide support for practical implementations of such an IP based approach in the case of an additive valuation setting. We believe that this chapter addresses some unanswered research questions for core determination in a budget constrained setting and represents a first attempt to define a framework and demonstrate an implementation to simultaneously determine core item prices and allocation.

Chapter 6

Conclusion

In this thesis, we have attempted to model budget constrained combinatorial auctions in the context of sponsored search and other multi-unit allocation settings. The last decade has seen an explosive growth in application of auction theory to resource allocation problems in many online as well as offline settings. Practical applications have included online advertising, supply chain coordination, procurement auctions, airport slot allocation, wireless spectrum allocation and fantasy sports. Researchers from diverse fields such as computer science, economics and operations research have contributed significantly to literature that examines economic properties and computational issues of these auction formats. An issue that has received very limited attention from the research community is the presence of hard budget constraints. Hard budget constraints introduce a non-linearity effect in bidder valuations, which results in mechanisms like the VCG losing their economic properties. Further, the presence of budget constraints makes the resource allocation and pricing problem significantly more difficult from a computational standpoint as compared to algorithms for the unconstrained case. We provide various frameworks to address multi-unit, budget constrained auctions from the perspective of the auctioneer as well as the bidders.

Auction mechanisms have become the de-facto standard for pricing and allocation of ad space for many online advertising channels. The significant growth in revenues that has occurred for firms such as Google, Yahoo and Facebook over the last few years can be mostly attributed to the application of auctions to online advertising. Auctions are prevalent across online advertising channels such as search engine marketing, graphical display ads, social media ads, contextual text ads and mobile advertising. These online advertising channels have the channel operator (e.g. Google, Yahoo etc.) creating a platform for advertisers to set bids for various biddable units and budgets for campaigns, while the channel operator solves an allocation and pricing problem with the bidder inputs. Currently, these mechanisms account for more than $30 billion dollars of advertising spend online. The significant economic value, the pervasive use of these auctions and the lack of research addressing computational implications of budget constraints in these auctions motivated us to address in this thesis the following research questions.

1. Search engine platforms currently use generalized second price (GSP) auctions for pricing and allocation. The GSP auction is not incentive compatible. Can we formulate the auction problem in the incentive compatible VCG framework and use optimization methods to solve optimally in real time typical problem sizes that the search engine operators face? Further, can we enhance the expressiveness of the auction for the bidders and still solve the auction in real time?

2. The large advertisers bid on hundreds of thousands of keywords on a daily basis and have hard budget constraints to deal with. Are there computational methods that can help advertisers determine optimal bids to place on the various advertising channels when dealing with tens of millions of decision variables?

3. The auctioneer is typically interested in revenue maximization. In case of bidders with hard budget constraints and additive valuations over items, can we use the decomposible constraint structure of the revenue maximization problem to efficiently solve the problem to optimality?

4. When the auctioneer is aiming to achieve social value maximization and the bidders have hard budget constraints along with additive valuations over items, are there computational methods to help the auctioneer arrive at an optimal allocation and price vector (specifically individual prices for items) such that neither the auctioneer nor any of the bidders have an incentive to deviate from the optimal solution? Further, can we extend our methodology to address general package auctions?

Chapter 2 attempts to address research question 1 posed above. Instead of specifying a single bid for a keyword, the auction format we consider provides the bidders the ability to express different bids for various ad slots. Here, the allocation and pricing problem faced by the auctioneer is formulated as an assignment problem. Using the Hungarian algorithm for assignment problem, we demonstrate how the auctioneer can solve the allocation and pricing problem in real time, even with this increased expressiveness. Consequently, this also provides VCG prices for the auction. We also extend the model to solve for the option of allocating multiple, contiguous slots to advertisers (which is appropriate for graphical ads). Currently, Facebook uses a VCG mechanism for its biddable advertising marketplace [Heg10]. Advertisers bid to place ads for specific targets (e.g., age range, likes & interests, geo etc.) and Facebook uses a VCG mechanism to determine allocation and pricing. Our approach provides an algorithm that can determine the optimal solution for their auction in polynomial time.

In chapter 3, we formulate the problem that budget constrained advertisers face while attempting to determine optimal bids for each keyword of interest. We formulate the problem as an integer program and show that the linear relaxation to the problem has a "near-integral" property. We use column generation approach to solve the linear relaxation to optimality for large scale problems (problems that involve upto 24 million decision variables). Further, we demonstrate the application of a branch and price algorithm to solve for the IP formulation. To our knowledge, this is the first time that such methods have been applied to solve large-scale bid determination problems in online advertising from the bidder's perspective. Chapter 3, thus, addresses research question 2.

To address research question 3, we use a MIP formulation to determine optimal prices and allocation for a revenue maximization auctioneer objective and additive bidder valuations with hard budget constraints. Chapter 4 uses the Benders decomposition framework to recast the MIP formulation as an iterative auction mechanism. Although the Benders formulation does not provide any improvement over the OPL results (OPL is a commercial software to solve MIPs), it does provide a framework to arrive at the optimal solution with limited information revelation by the bidders.

Chapter 5 deals with the determination of a core solution for a social welfare maximizing auctioneer with the bidders having hard budget constraints. Chapter 5 seeks to address research question 4. First we deal with the case when bidder valuations are assumed to be additive. In this case, we use a constraint generation method identical to that in Day & Raghavan [DR07] to arrive at a core solution. However, our model has to incorporate the budget constraints and thus, we develop a method to determine core prices and allocation simultaneously. Further, our method provides individual item prices for the additive case. To the best of our knowledge, this is the first time that core solution determination for budget constrained auctions with individual item price feedback has been addressed algorithmically in the research community. We, also, extend our approach to demonstrate how general package auctions with budget constraints can be formulated as MIPs and extend the use of the constraint generation technique to determine core prices and allocation.

Bibliography

[AGM06] G. Aggarwal, A. Goel, and R. Motwani. Truthful auctions for pricing keywords. In *7th ACM Conference on Electronic Commerce*, pages 1–7, 2006.

[AM02] Larry Ausubel and Paul Milgrom. Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1(1):1–42, 2002.

[AM04] N. Andelman and Y. Mansour. Auctions with budget constraints. In *9th Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 26–38, 2004.

[ARV05] A. Animesh, V. Ramachandran, and S. Viswanathan. Online advertisers' bidding strategies for search, experience, and credence goods: An empirical investigation. In *Workshop on Sponsored Search Auctions, 6th ACM conference on Electronic commerce*, 2005.

[Asd06] K. Asdemir. Dynamics of bidding in sponsored search auctions: An analytical investigation. In *Workshop on Sponsored Search Auctions, 7th ACM conference on Electronic commerce*, 2006.

[BCI+05] C. Borgs, J. Chayes, N. Immorlica, M. Mahdian, and A. Saberi. Multi-unit auctions with budget-constrained bidders. In *6th ACM conference on Electronic commerce*, pages 44–51, 2005.

[BDH06] M.O. Ball, G.L. Donahue, and K. Hoffman. *Auctions for Safe, Efficient and Equitable Allocation of Airspace System Resources*, chapter 20, pages 507–538. Combinatorial Auctions. MIT Press, 2006.

[BDHK06] M. Bichler, A. Davenport, G. Hohner, and J. Kalagnanam. *Industrial Procurement Auctions*, chapter 23, pages 593–611. Combinatorial Auctions. MIT Press, 2006.

[BdVSV02] S. Bikhchandani, S. de Vries, J. Schummer, and R. Vohra. *Linear Programming and Vickrey Auctions*, pages 75–115. Mathematics of the Internet: E-auctions and Markets. Springer Verlag, 2002.

[BK01] J.P. Benoit and Vijay Krishna. Multiple object auctions with budget constrained bidders. *Review of Economic Studies*, 68(1):155–179, 2001.

[Cra06] P. Cramton. *Simultaneous Ascending Auctions*, chapter 4, pages 99–114. Combinatorial Auctions. MIT Press, 2006.

[CS06] C. Caplice and Y. Sheffi. *Combinatorial Auctions for Truckload Transportation*, chapter 21, pages 539–572. Combinatorial Auctions. MIT Press, 2006.

[DGS86] G. Demange, D. Gale, and M. Sotomayor. Multi-item auctions. *The Journal of Political Economy*, 94(4):863–872, 1986.

[DLN08] S. Dobzinski, R. Lavi, and N. Nisan. Multi-unit auctions with budget limits. In *FOCS*, pages 260–269, 2008.

[DR07] Robert Day and S. Raghavan. Fair payments for efficient allocations in public sector combinatorial auctions. *Management Science*, 53(9):1389–1406, 2007.

[DR09] Robert Day and S. Raghavan. Matrix bidding in combinatorial auctions. *Operations Research*, 57(4):916–933, 2009.

[Dye84] M.E. Dyer. An O(n) algorithm for the multiple-choice knapsack problem. *Mathematical Programming*, 29:57–63, 1984.

[EK03] W. Elmaghraby and P. Keskinocak. Combinatorial auctions in procurement. *The Practice of Supply Chain Management*, pages 245–258, 2003.

[EO05] B. Edelman and M. Ostrovsky. Strategic bidder behavior in sponsored search auctions. In *Workshop on Sponsored Search Auctions, 6th ACM conference on Electronic commerce*, 2005.

[EOM07] B. Edelman, M. Ostrovsky, and M.Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259, 2007.

[FBP07] J. Feng, H.K. Bhargava, and D.M. Pennock. Implementing sponsored search in web search engines: Computational evaluation of alternative mechanisms. *Informs Journal on Computing*, 19(1):137–148, 2007.

[FMPS07] J. Feldman, S. Muthukrishnan, M. Pal, and C. Stein. Budget optimization in search-based advertising auctions. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 40–49, 2007.

[Goo05] J. Goodman. Pay-per-percentage of impressions: An advertising method that is highly robust to fraud. In *Workshop on Sponsored Search Auctions, 6th ACM conference on Electronic commerce*, 2005.

[GY08] A. Ghose and S. Yang. An empirical analysis of sponsored search performance in search engine advertising. In *Proceeding of the international conference on Web search and web data mining*, pages 241–250, 2008.

[Heg10] J. Hegeman. Facebook's ad auction. In *Workshop on Sponsored Search Auctions, 11th ACM conference on Electronic commerce*, 2010.

[HMvdHW06] K. Hoffman, D. Menon, S. van den Heever, and T. Wilson. *Observations and near direct implementations of the ascending proxy auctions*, chapter 17, pages 415–450. Combinatorial Auctions. MIT Press, 2006.

[IA06] G. Iyengar and A.Kumar. Characterizing optimal keyword auctions. In *Workshop on Sponsored Search Auctions, 7th ACM conference on Electronic commerce*, 2006.

[JR05] B.J. Jansen and M. Resnick. Examining searcher perceptions of and interactions with sponsored results. In *Workshop on Sponsored Search Auctions, 6th ACM conference on Electronic commerce*, 2005.

[KDM03] S. Kumar, M. Dawande, and V. Mookerjee. Revenue maximization in web advertisement. In *13th Annual Workshop on Information Technologies and Systems (WITS)*, pages 121–126, 2003.

[KL04] B. Kitts and B. Leblanc. Optimal bidding on keyword auctions. *Electronic Markets*, 14(3):186–201, 2004.

[LBNS06] K. Leyton-Brown, E. Nudelman, and Y. Shoham. *Empirical Hardness Models for Combinatorial Auctions*, chapter 19, pages 479–504. Combinatorial Auctions. MIT Press, 2006.

[Leo83] H.B. Leonard. Elicitation of honest preferences for the assignment of individuals to positions. *The Journal of Political Economy*, 91(3):461–479, 1983.

[LJA06] D. Liu, J.Chen, and A.B.Whinston. Weighted unit price auctions. In *Workshop on Sponsored Search Auctions, 7th ACM conference on Electronic commerce*, 2006.

[LMS06] D. Lehmann, R. Mueller, and T. Sandholm. *The Winner Determination Problem*, chapter 12, pages 297–318. Combinatorial Auctions. MIT Press, 2006.

[LT05] W.S. Lim and C.S. Tang. An auction model arising from internet search service providers. *European Journal of Operational Research*, 172(3):956–970, 2005.

[MSVV07] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *Journal of the ACM*, 54(5):1–19, 2007.

[MW81] T.L. Magnanti and R.T. Wong. Accelerating benders decomposition: Algorithmic enhancement and model selection. *Operations Research*, 29(3):464–484, 1981.

[Pis95] D. Pisinger. *Algorithms for Knapsack Problems.* PhD thesis, University of Copenhagen, 1995.

[PS05] D.C. Parkes and T. Sandholm. Optimize-and-dispatch architecture for expressive ad auctions. In *Workshop on Sponsored Search Auctions, 6th ACM conference on Electronic commerce*, 2005.

[PV08] M. Pai and R. Vohra. Optimal auctions with financially constrained bidders. *Working Paper*, 2008.

[RP03] E. Rolland and R.S. Patterson. Classification in online pay-for-performance advertising. In *13th Annual Workshop on Information Technologies and Systems (WITS)*, pages 219–224, 2003.

[RW06] P. Rusmevichientong and D. Williamson. An adaptive algorithm for selecting profitable keywords for search-based advertising services. In *7th ACM conference on Electronic commerce*, pages 260–269, 2006.

[SZ79] A. Sinha and A.A. Zoltners. The multiple-choice knapsack problem. *Operations Research*, 27(3):503–515, 1979.

[Var07] Hal Varian. Position auctions. *International Journal of Industrial Organization*, 25(6):1163–1178, 2007.

[Wol98] L.A. Wolsey. *Integer Programming*, pages 57–62. Interscience Series in Discrete Mathematics and Optimization. Wiley, 1998.

[WZ07] T.A. Weber and Z.E. Zheng. A model of search intermediaries and paid referrals. *Information Systems Research*, 18(4):414–436, 2007.

[YG00] Y.K.Che and Ian Gale. The optimal mechanism for selling to a budget-constrained buyer. *Journal of Economic Theory*, 92:198–233, 2000.

[Zem84] E. Zemel. An O(n) algorithm for the linear multiple-choice knapsack problem and related problems. *Information Processing Letters*, 18:123–128, 1984.

[ZR06] Y. Zhou and R.Lukose. Vindictive bidding in keyword auctions. In *Workshop on Sponsored Search Auctions, 7th ACM conference on Electronic commerce*, 2006.

[ZSF09] R.L. Zhan, Z.M. Shen, and J. Feng. Ranked item auctions and online advertisement. *Production and Operations Management*, 16(4):510–522, 2009.