Combinatorial Auctions: A Survey

Sven de Vries* & Rakesh V. Vohra†

October 25, 2001

Abstract

Many auctions involve the sale of a variety of distinct assets. Examples are airport time slots, delivery routes, network routing and furniture. Because of complementarities or substitution effects between the different assets, bidders have preferences not just for particular items but for sets of items. For this reason, economic efficiency is enhanced if bidders are allowed to bid on bundles or combinations of different assets. This paper surveys the state of knowledge about the design of combinatorial auctions and presents some new insights.

(*Auctions; Combinatorial Optimization*)

*Zentrum Mathematik, TU München, D-80290 München, Germany.
e-mail: devries@ma.tum.de

†Department of Managerial Economics and Decision Sciences, Kellogg School of Management, Northwestern University, Evanston IL 60208. e-mail: r-vohra@nwu.edu

1. Introduction

Many auctions involve the sale of a variety of distinct assets. Examples are the FCC spectrum auction (`http://www.fcc.gov/wtb/auctions/`) and auctions for airport time slots (Rassenti *et al.* 1982), railroad segments (Brewer 1999), delivery routes (Caplice 1996) and network routing (Hershberger and Suri 2001). Because of complementarities or substitution effects between different assets, bidders have preferences not just for particular items but for sets of items, sometimes called bundles.

To illustrate, suppose you must auction off a dining room set consisting of four chairs and a table. Would you wish to auction off the entire set or run five separate auctions for each piece? The answer depends, of course, on what bidders care about. If every bidder is interested in the dining room set and nothing less, the first option is preferable. If some bidders are interested in the set but others are interested only in a chair or two it is not obvious what to do. If you believe that you can raise more by selling off the chairs separately than the set, the second option is preferable. Notice, deciding requires a knowledge of just how much bidders value different parts of the ensemble. For this reason, economic efficiency is enhanced if bidders are allowed to bid directly on *combinations* of different assets instead of bidding only on individual items. Auctions where bidders are allowed to submit bids on combinations of items are usually called combinatorial auctions. 'Combinational auctions' is more accurate, but in this survey we will comply with convention.

Auctions where bidders submit bids on combinations have recently received much attention. See for example Caplice (1996), Rothkopf *et al.* (1998), Fujishima *et al.* (1999), and Sandholm (1999). However such auctions were proposed as early as 1976 (Jackson 1976) for radio spectrum rights.[1] Rassenti *et al.* (1982), a little later, propose such auctions to allocate airport time slots. Srinivasan *et al.* (1998) have proposed a mechanism for trading financial securities that allows buyers and sellers to offer bundles of financial instruments; their mechanism treats financial securities as divisible. Increases in computing power have made combinatorial auctions more attractive to implement.

Perhaps the best known auction of heterogenous objects, has been the 1994 FCC auction of spectrum rights. Here bidders were interested in different collections of spectrum licences. The FCC decided against an auction in which bidders would bid on subsets of licences as it was thought, at the time, that such an auction would be cumbersome to run. Instead, the FCC used a separate auction for each licence. However, the auctions were run in parallel and bidders were allowed to participate in as many of them as possible. A more detailed description of these auctions can be found in Cramton (2001). In the fall of 2001 the FCC plans to run its first auction (Auction #31) in which bidders will be allowed to bid on combinations of spectrum licences.

In contrast to the FCC, a number of large firms have actively embraced combinatorial auctions to procure logistics services. Olson *et al.* (2000) describes the design and use of a combinatorial auction that was employed by Sears in 1993 to select carriers. Here the objects bid upon were delivery routes (called lanes). Since a truck must return to its depot, it was more profitable for bidders to have their trucks full on the return journey. Being allowed to bid on bundles gave bidders the opportunity to construct routes

[1] We thank Dr. Jackson for bringing the reference to our attention.

that utilized their trucks as efficiently as possible. In fact a number of logistics consulting firms tout software to implement combinatorial auctions. SAITECH-INC, for example, offers a software product called SBIDS that allows trucking companies to bid on bundles of lanes. Logistics.com's system is called OptiBid$^{\text{TM}}$. Logistics.com claims that more than $5 billion in transportation contracts have been bid by January 2000 using OptiBid$^{\text{TM}}$ by Ford Motor Company, Wal-Mart and K-Mart.[2]

Graves *et al.* (1993) describes the auction of seats in a course that is executed regularly at the University of Chicago's Business School. Strevell and Chong (1985) describe the use of an auction to allocate vacation time slots. Banks *et al.* (1989) propose a combinatorial auction for selecting projects on the space shuttle. It was tested experimentally but never implemented for political reasons.

Procurement auctions where bidders are asked to submit a collection of price-quantity pairs, for example, $4 a unit for 100 units; $3.95 a unit for 200 units etc. are also examples of combinatorial auctions. Here each price-quantity pair corresponds to a bundle of homogenous goods and a bid. If the goods are endowed with attributes like payment terms, delivery and quality guarantees they become bundles of heterogenous objects. Davenport and Kalagnanam (2001) describe a combinatorial auction for such a context that is used by a large food manufacturer. Ausubel and Cramton (1998) and Bikhchandani and Huang (1993) describe the auction for Treasury Securities

[2] Yet another firm called InterTrans Logistics Solutions, offers a software product called Carrier Bid Optimizer that allows trucking companies to bid on bundles of lanes over several bidding rounds. They appear to have been acquired by $i2$ and we have not been able to find anything more about them. Two more companies have been formed to provide software for combinatorial auctions. They are CombineNet and Trade Extensions.

that is actually used by the US Department of Treasury and compare it with other mechanisms.

In 1998 OptiMark Technologies (`http://www.optimark.com/markets.html`) offered an automated trading system that allows bidders to submit price-quantity-stock triples (along with a priority list). The Securities and Exchange Commission (SEC) approved a proposal by the Pacific Stock Exchange to implement this electronic trading system. That same year the NASDAQ market announced plans to introduce this technology to its dealers and investors trading stocks listed on it. The system was adopted by the Osaka Securities exchange, but suspended in June of 2001.

The designer of a combinatorial auction faces a surfeit of choices, some of which we list below.

1. Should the collection of bundles on which bids are allowed be restricted? If so, to what?

2. Should the auction involve a single round of bidding? If so, how should the bundles be allocated as a function of the bids and what should the payment rules be?

3. If the auction is to involve multiple rounds (call such auctions iterative), what information should be revealed to bidders from one round to the next?

The choice depends on the objectives of the auctioneer. For example, is it to maximize revenue or economic efficiency? Other considerations also matter: Speed, practicality, bidders preferences, the need to discourage collusion and encourage competition amongst the bidders.

Nevertheless, no matter how one chooses there are three problems that every auction designer must resolve. The first has to do with bid expression. The second is how to allocate bundles amongst bidders so as to optimize some criterion. Third, what are the incentive implications of the solutions offered to the first two. We discuss the first two issues in Sections 2 and 3. In Section 4 we will disucss iterative auctions and conclude with a discussion of the third major problem —incentive issues— in Section 5.

In the interests of space, a number of issues relevant to auction design in general are omitted. These are interdependent values (Jehiel and Molodvanu 2001), privacy in bidding (Naor, Pinkas and Sumner 1999) and 'false name' bidding (Yokoo, Sakurai and Matsubara 2001).

2. Bid Expression

The first and most obvious difficulty an auction which allows bidders to bid on combinations faces is that each bidder may have to determine a bid for *every* bundle he is interested in. The second problem is how to transmit this bidding function in a succinct way to the auctioneer.

In theory a bidder could be interested in every combination of items possible. In practice resource constraints on the part of bidders will limit the number of combinations they will submit bids on. For example, in the auction of spectrum, estimating the value of a bundle of spectrum requires putting together a business plan. Having decided on which combinations to place a bid on, the next step is to communicate this to the auctioneer.

The difficulty now is to communicate this list, if it is particularly large, in a way that will be computationally useful to the auctioneer. One approach, not much explored, is to rely on an 'oracle'. An oracle is a program (black box) that, for example, given a bidder and a subset computes the bid for it.

Thus bidders submit oracles rather than bids. The auctioneer can simply invoke the relevant oracle at any stage to determine the bid for a particular subset.[3] Effectiveness of this approach depends on the computational efficiency of the oracle.

Alternatively, the auctioneer may specify a bidding language that all bidders must use to encode their preferences. A discussion of various ways in which bids can be restricted and their consequences can be found in Nisan (1999). In that paper Nisan asks, given a language for expressing bids, what preferences over subsets of objects can be correctly represented by the language. What seems clear is that a computationally efficient oracle or language relies on restricting the preferences of bidders, or combinations that bidders can bid on.

Another way to overcome the complexity of communicating bids and determining the winning bidders is to restrict the collection of bundles on which bidders might bid. Different scenarios along this idea are developed by Rothkopf *et al.* (1998), see also Subsection 3.4.

Even if this problem is resolved (in a non-trivial way) to the satisfaction of the parties involved, it still leaves open the problem of deciding which collection of bids to accept.

3. Winner Determination

The problem of identifying which set of bids to accept has usually been dubbed the winner determination problem. The precise formulation will depend on the objectives of the auctioneer. Here we focus on the formulation

[3] Sandholm (1999) points out that another advantage of oracles is that bidders need not be present. Their application does rely on the probity of the auctioneer.

described in Rothkopf *et al.* (1998) and by Sandholm (1999). To distinguish it from other possible formulations we call it the Combinatorial Auction Problem (CAP).[4] CAP can be formulated as an Integer Program. We will survey what is known about the CAP. It assumes a knowledge of linear programming and familiarity with basic graph theoretic terminology.

3.1 The CAP

To formulate CAP as an integer program let N be the set of bidders and M the set of m distinct objects. For every subset S of M let $b^j(S)$ be the bid that agent $j \in N$ has announced he is willing to pay for S.[5] From the formulation it will be clear that bids with $b^j(S) < 0$ will never be selected. So, without loss of generality we can assume that $b^j(S) \geq 0$. Let $y(S, j) = 1$ if the bundle $S \subseteq M$ is allocated to $j \in N$ and zero otherwise.

$$\max \sum_{j \in N} \sum_{S \subseteq M} b^j(S) y(S, j)$$

$$\text{s.t.} \sum_{S \ni i} \sum_{j \in N} y(S, j) \leq 1 \ \forall i \in M$$

$$\sum_{S \subseteq M} y(S, j) \leq 1 \ \forall j \in N$$

$$y(S, j) = 0, 1 \ \forall S \subseteq M, \ j \in N$$

The first constraint ensures that overlapping sets of goods are never assigned. The second ensures that no bidder receives more than one subset. Call this formulation CAP1. Problem CAP as formulated here is an instance of what is known as the Set Packing Problem (SPP) which is described below.

[4] We assume that the auctioneer is a seller and bidders are buyers.

[5] Implicit is the assumption that bidders care only about the combinations they receive and not on what other bidders receives.

When bid functions are superadditive, a more succinct formulation is possible. Let $b(S) = \max_{j \in N} b^j(S)$ and set $x_S = 1$ if the highest bid on the set S is to be accepted and zero otherwise. Then CAP can be formulated as:

$$\begin{aligned} \max \sum_{S \subset M} b(S)x_S \\ \text{s.t.} \quad \sum_{S \ni i} x_S \leq 1 \ \forall i \in M \\ x_S = 0, 1 \ \forall S \subset M \end{aligned}$$

Here the constraint $\sum_{S \ni i} x_S \leq 1 \ \forall i \in M$ ensures that no object in M is assigned to more than one bidder. Call this formulation CAP2. It is also an instance of the SPP.[6]

There is another interpretation of the CAP possible. If the bids submitted are the actual values that bidders have for various combinations, then the solution to the CAP is the economically efficient allocation of indivisible objects in an exchange economy.

We have formulated CAP1 under the assumption that there is at most one copy of each object. It is an easy matter to extend the formulation to the case when there are multiple copies of the same object and bidders may want more than one copy of the same unit. Such extensions, called multi-unit combinatorial auctions, are investigated by Leyton-Brown *et al.* (2000) as well as by Gonen and Lehman (2000).[7]

[6] In the absence of superadditivity, one must impose the additional constraints of CAP1 that prevent any bidder receiving more than one bundle in an optimal solution.

[7] If the number of units of each type is large, then one could approximate the problem of selecting the winning set of bids using a linear program. The relevant decision variables would be the percentage of each type allocated to a bidder.

The formulation for winner determination just given is not flexible enough to encompass some of the variations that have been considered in the literature. Here is a more comprehensive formulation:

$$\max \sum_{j \in N} \sum_{S \in \Omega_j} b_j(q) y(q,j)$$

$$\text{s.t.} \sum_{j \in N} \sum_{S \in \Omega_j} y(q,j) q_i \leq m_i \ \forall i \in M \tag{GCAP$_1$}$$

$$y^j \in P_j^A \ \forall j \in N \tag{GCAP$_2$}$$

$$y \in P^A \tag{GCAP$_3$}$$

$$y^j \in P_j^B \ \forall j \in N \tag{GCAP$_4$}$$

$$y(q,j) = 0,1 \ \forall q \in \Omega_j \ \forall j \in N \tag{GCAP$_5$}$$

Here m_i is the number of units of object i available and q is an integral vector whose i^{th} component represents the number of units of object i demanded. If $y(q,j) = 1$ this means agent j is allocated the bundle represented by the vector q.

The sets $\Omega_j \subseteq \mathbb{N}^M \cap [0, m_1] \times [0, m_2] \times \cdots \times [0, m_m]$ model restrictions on what bidders can bid on. They can be fixed by the auctioneer or she might permit bidders to specify them (subject to some constraints as the FCC-restrictions for auction #31 on the number of packages a bidder might bid on).

The constraints (GCAP$_1$) ensure that no more of an item is allocated than the available supply. The constraints (GCAP$_2$) are imposed by the auctioneer and enforce capacity constraints on the bidders; for example no bidder is supposed to win more than two items, no bidder is supposed to win more than 40% of the total business, etc. Here P_j^A denotes the polyhedron of feasible solutions to these constraints.[8]

Constraints (GCAP$_3$) permit the auctioneer to restrict the overall allocation. For example, the allocation must be edges that form a path or a tree. Here P_A denotes the polyhedron of feasible solutions to these restrictions. [9]

Constraints (GCAP$_4$) allow each bidder to restrict the allocations he might win. The feasible solutions satisfying these bidder imposed restrictions are represented by the polyhedron P_i^B. For example, if he has a subadditive valuation, he might put $P_i^B = \{y \in \mathbf{R}^{\Omega_j} | \sum_{S\in\Omega_j} y(S,j) \leq 1\}$ to ensure that he does not pay more than what he bid.

Finally, (GCAP$_5$) ensures that we end with an integral allocation.

3.2 The Set Packing Problem

The SPP is a well studied integer program. Given a ground set M of elements and a collection V of subsets with non-negative weights, find the largest weight collection of subsets that are pairwise disjoint. To formulate this problem as an integer program, let $x_j = 1$ if the j^{th} set in V with weight c_j is selected and $x_j = 0$, otherwise. Define a_{ij} to be 1 if the j^{th} set in V contains element $i \in M$. Given this, the SPP can be formulated as:

$$\max \sum_{j\in V} c_j x_j$$

$$\text{s.t.} \quad \sum_{j\in V} a_{ij} x_j \leq 1 \ \forall i \in M$$

$$x_j = 0, 1 \ \forall j \in V$$

[8] The auctioneer could choose P_j^A that in effect restrict Ω_j. To avoid this one could require that $P_j^A \cap \mathbf{N}^{\Omega_j}$ to be full dimensional.

[9] We assume that P^A can not be described as the cartesian product of an interval on the coordinate axis and a lower-dimensional polytope.

As noted by Rothkopf *et al.* (1998) and Sandholm (1999) the CAP is an instance of the SPP. Just take M to be the set of objects and V the set of all subsets of M.

Before continuing with a discussion of the SPP we mention two of its close relatives. The first is called the **set partitioning problem** (SPA) and the second is called the **set covering problem** (SCP). Both would be relevant had we cast the auction problem in procurement rather than selling terms. The auctions used in the transport industry are of this set covering type. In that setting, objects are origin-destination pairs, called lanes. Bidders submit bids on bundles of lanes that represent how much they must be offered to undertake the deliveries on the specified lanes. The auctioneer wishes to choose a collection of bids of lowest cost such that all lanes are served.[10]

While SPA and SPC are cosmetically similar to the SPP they have different computational and structural properties. The survey by Balas and Padberg [1976] contains a bibliography on applications of the SPP, SCP and SPA.

3.3 Complexity of the SPP

How hard is the SPP to solve? By enumerating all possible 0-1 solutions we can find an optimal solution in a finite number of steps. If $|V|$ is the number of variables, then the number of solutions to check would be $2^{|V|}$, clearly impractical for all but small values of $|V|$. For the instances of SPP that arise in the CAP, the cardinality of V is the number of bids; possibly a large number.

[10] In fact, one must specify not only lanes but volume as well, so this problem constitutes an instance of a multi-unit combinatorial auction.

No polynomial time algorithm for the SPP is known and is unlikely to exist because SPP is NP-hard.[11]

For the CAP, this discussion of complexity may have little relevance. To see why, suppose one takes the number of bids as a measure of the size of the input and this number is exponential in $|M|$. Any algorithm for CAP that is polynomial in the number of bids but exponential in the number of items would, formally, be polynomial in the input size but impractial for $|M|$ large. Thus effective solution procedures for the CAP must rely on two things. The first is that the number of distinct bids is not large and is structured in computationally useful ways. The second is that the underlying SPP can be solved reasonably quickly.

3.4 Solvable Instances of the SPP

The usual way in which instances of the SPP can be solved by a polynomial algorithm is when the extreme points of the polyhedron $P(A) = \{x : \sum_{j \in V} a_{ij} x_j \leq 1 \; \forall i \in M; \; x_j \geq 0 \; \forall j \in V\}$ are all integral, i.e. 0-1. In these cases we can simply drop the integrality requirement from the SPP and solve it as a linear program. Linear programs can be solved in polynomial time. It turns out that in most of these cases, because of the special structure of these problems, algorithms more efficient than linear programming ones exist. Nevertheless, the connection to linear programming is important because it allows one to interpret dual variables as prices for the objects being auctioned. We will say more about this later in the paper.

A polyhedron with all integral extreme points is called **integral**. Identifying sufficient conditions for when a polyhedron is integral has been a

[11] More precisely, the recognition version of SPP is NP-complete.

cottage industry in integer programming. These sufficient conditions involve restrictions on the constraint matrix, which in this case amount to restrictions on the kinds of subsets for which bids are submitted. We list the most important ones here.

Rothkopf *et al.* (1998) covers the same ground but organizes the solvable instances differently as well as suggesting auction contexts in which they may be salient. An example of one such context is given below.

3.4.1 Total Unimodularity

The most well known of these sufficient conditions is **total unimodularity**, sometimes abbreviated to 'TU'. A matrix is said to be TU if the determinant of every square submatrix is 0, 1 or -1. If the matrix $A = \{a_{ij}\}_{i \in M, j \in V}$ is TU then all extreme points of the polyhedron $P(A)$ are integral (see Nemhauser and Wolsey 1988).

A special case of TU matrices are those with the consecutive ones property (Nemhauser and Wolsey 1988). A 0-1 matrix has the *consecutive ones property* if the non-zero entries in each column occur consecutively. Rothkopf *et al.* (1998) offer the following to motivate the consecutive ones property in the auction context. Suppose the objects to be auctioned are parcels of land along a shore line. The shore line is important as it imposes a linear order on the parcels. In this case it is easy to imagine that the most interesting combinations (in the bidders eyes) would be contiguous. If this were true it would have two computational consequences. The first is that the number of distinct bids would be limited (to intervals of various length) by a polynomial in the number of objects. Second, the constraint matrix A of CAP2 would have the consecutive ones property in the columns.[12]

[12] If the valuation of each bidder is additive over sets of nonadjacent intervals and super-

3.4.2 Balanced Matrices

A 0-1 matrix B is **balanced** if it has no square submatrix of odd order with exactly two 1's in each row and column. If the matrix B is balanced then (see Schrijver 1986) the following linear program:

$$\max\left\{\sum_j c_j x_j : \sum_j b_{ij} x_j \leq 1 \ \forall i,\ x_j \geq 0 \ \forall j\right\}$$

has an integral optimal solution whenever the c_j's are integral.

One instance of balancedness that may be relevant to the CAP. Consider a tree T with a distance function d. For each vertex v in T let $N(v,r)$ denote the set of all vertices in T that are within distance r of v. If you like, the vertices represent parcels of land connected by a road network with no cycles. Bidders can bid for subsets of parcels but the subsets are constrained to be of the form $N(v,r)$ for some vertex v and some number r. Now the constraint matrix of the corresponding SPP will have one column for each set of the form $N(v,r)$ and one row for each vertex of T. This constraint matrix is balanced. See Nemhauser and Wolsey (1988) for a proof as well as efficient algorithms. In the case when the underlying tree T is a path the constraint matrix reduces to having the consecutive ones property. If the underlying network were not a tree then the corresponding version of SPP becomes NP-hard.

additive over sets of adjacent intervals, then CAP2 models the situation correctly and the problem is polynomially solvable. Otherwise one has to use CAP1 which adds constraints that violate the consecutive ones property. Müller pointed out that it is a consequence of Keil (1992) that this problem becomes NP-hard in general. For more efficiently solvable subcases see van Hoesel and Müller (2000).

3.4.3 Perfect Matrices

More generally, if the constraint matrix A can be identified with the vertex-clique adjacency matrix of what is known as a perfect graph, then SPP can be solved in polynomial time. The interested reader should consult Chapter 9 of the book by Grötschel *et al.* (1988) for more details. The algorithm, while polynomial, is impractical.

We now describe one instance of perfection that may be relevant to the CAP. It is related to the example on balancedness. Consider a tree T. As before imagine the vertices represent parcels of land connected by a road network with no cycles. Bidders can bid for *any* connected subset of parcels. Now the constraint matrix of the corresponding SPP will have one column for each connected subset of T and one row for each vertex of T. This constraint matrix is perfect (Nemhauser and Wolsey 1988).

3.4.4 Graph Theoretic Methods

There are situations where $P(A)$ is not integral yet the SPP can be solved in polynomial time because the constraint matrix of A admits a graph theoretic interpretation in terms of an *easy* problem. The most well known instance of this is when each column of the matrix A contains at most two 1's. In this case the SPP becomes an instance of the maximum weight matching problem in a graph which can be solved in polynomial time.

Each row (object) corresponds to a vertex in a graph. Each column (bid) corresponds to an edge. The identification of columns of A with edges comes from the fact that each column contains two non-zero entries. It is well known that $P(A)$ contains fractional extreme points. Consider for example a graph which is a cycle on three vertices. A comprehensive discussion of the matching problem can be found in the book by Lovász and Plummer (1986). The subclass of SPP where each column has at most $K \geq 3$ non-zero entries is NP-hard.

It is natural to ask what happens if one restricts the number of 1's in each row rather than column. The subclass of SPP with at most two non-zero entries per row of A is NP-hard. These instances correspond to what is called the stable set problem in graphs, a notoriously difficult problem.[13]

Another case is when the matrix A has the circular ones property. A 0-1 matrix has the **circular ones** property if the non-zero entries in each column (row) are consecutive; first and last entries in each column (row) are treated consecutively. Notice the resemblance to the consecutive ones property. In this case the constraint matrix can be identified with what is known as the vertex-clique adjacency matrix of a circular arc graph.[14] The SPP then becomes the maximum weight independent set problem for a circular arc graph. This problem can also be solved in polynomial time, see Golumbic *et al.* (1988). Following the parcels of land on the seashore example, the circular ones structure makes sense when the land parcels lie on the shores of an island or lake.

[13] The instance of CAP produced by the radio spectrum auction in Jackson (1976) reduces to just such a problem.

[14] Take a circle and a collection of arcs of the circle. To each arc associate a vertex. Two vertices will be adjacent if the corresponding arcs overlap. The consecutive ones property also bears a graph theoretic interpretation. Take intervals of the real line and associate them with vertices. Two vertices are adjacent if the corresponding intervals overlap. Such graphs are called interval graphs.

3.4.5 Using Preferences

The solvable instances above work by restricting the sets of objects over which preferences can be expressed. Another approach would be to study the implications of restrictions in the preference orderings of the bidders themselves.

One common restriction that is placed on $b^j(\cdot)$ is that it be non-decreasing (that is, $b^j(S) \leq b^j(T)$ for $S \subseteq T$) and supermodular (that is, $b^j(S)+b^j(T) \leq b^j(S \cup T) + b^j(S \cap T)$). Suppose now that bidders come in two types. The type one bidders have $b^j(\cdot) = g^1(\cdot)$ and those of type two have $b^j(\cdot) = g^2(\cdot)$ where $g^r(\cdot)$ are non-decreasing, integer valued supermodular functions. Let N^r be the set of type r bidders.

Consider now the dual to the linear programming relaxation of CAP1:

$$\min \sum_{i \in M} p_i + \sum_{j \in N} q_j$$

$$\begin{aligned} \text{s.t. } & \sum_{i \in S} p_i + q_j \geq g^1(S) \ \forall S \subseteq M, \ j \in N^1 \\ & \sum_{i \in S} p_i + q_j \geq g^2(S) \ \forall S \subseteq M, \ j \in N^2 \\ & p_i, \ q_j \geq 0 \ \forall i \in M, \ j \in N \end{aligned}$$

This problem is an instance of the polymatroid intersection problem and is polynomially solvable; see Theorem 10.1.13 in the book by Grötschel *et al.* (1988). More importantly it has the property of being **totally dual integral**, which means that its linear programming dual, the linear relaxation of the original primal problem, has an integer optimal solution. This last observation is used in Bikhchandani and Mamer (1997) to establish the existence of competitive equilibria in exchange economies with indivisibilities. Utilizing the method to solve problems with three or more types of bidders is not possible because it is known in those cases that the dual problem above admits fractional extreme points. In fact the problem of finding an integer optimal solution for the intersection of three or more polymatroids is NP-hard.

Another restriction on bids/preferences that has been studied is the gross substitutes property (Kelso and Crawford 1982). To describe it let the value that bidder j assigns to the set $S \subset M$ of objects be $v_j(S)$. Given a vector of prices p on objects let the collection of subsets of objects that maximize bidder j's utility be denoted $D_j(p)$, defined as follows.

$$D_j(p) = \{S \subset M : v_j(S) - \sum_{i \in S} p_i \geq v_j(T) - \sum_{i \in T} p_i \quad \forall T \subset M\}.$$

The gross substitutes condition requires that for all price vectors p, p' such that $p' \geq p$, and all $A \in D_j(p)$, there exists $B \in D_j(p')$ such that $\{i \in A : p_i = p'_i\} \subset B$. A special case of the gross substitutes condition is when bidders are interested in multiple units of the same item and have diminishing marginal utilities.

In the case when each of the $b^j(\cdot)$ have the gross substitutes property, the linear programming relaxation of CAP1 has an optimal integer solution. This is proved in Kelso and Crawford (1982) as well as Gul and Stachetti (1997). In both cases a primal-dual algorithm for the linear relaxation of CAP1 is offered and interpreted as an auction.

3.5 Exact Methods

An exact method for solving the SPP and the CAP is one that is guaranteed to return a solution that is both feasible and optimal. They come in three varieties: branch and bound, cutting planes and a hybrid called branch and cut. Fast exact approaches to solving the SPP require algorithms that generate both good lower and upper bounds on the maximum objective function value of the instance. In general, the upper bound on the optimal solution value is obtained by solving a **relaxation** of the optimization problem. There are two standard relaxations for SPP: Lagrangean relaxation (where the feasible set is usually required to maintain 0-1 feasibility, but many if not all of the constraints are moved to the objective function with a penalty term) and the linear programming relaxation (where only the integrality constraints are relaxed—the objective function remains the original function). Lagrangean relaxation will be discussed in greater detail in Section 5 on iterative auctions.

Because even small instances of the CAP1 may involve a huge number of columns (bids) the techniques described above need to be augmented with another method known as column generation. Introduced by Gilmore and Gomory (1961) it works by generating a column when needed rather than all at once. An overview of such methods can be found in the article by Barnhart *et al.* (1994). Later in this paper we illustrate how this idea could be implemented in an auction.

One sign of how successful exact approaches are can be found in the article by Hoffman and Padberg (1993). They report being able to find an optimal solution to an instance of SPA with 1,053,137 variables and 145 constraints in under 25 minutes. In auction terms this corresponds to a problem with 145 items and 1,053,137 bids. A major impetus behind the desire to solve large instances of SPA (and SPC) quickly has been the airline industry. The problem of assigning crews to routes can be formulated as an SPA. The rows of the SPA correspond to flight legs and the columns to a sequence of flight legs that would be assigned to a crew. Like the CAP, in this problem the number of columns grows exponentially with the number of rows.[15] For the SPP, the large instances that have been studied have usually arisen from relaxations of SPA's. Given the above we believe that established integer programming methods will prove quite successful when applied to the solution of CAP.

Logistics.com's OptiBid$^{\mathrm{TM}}$ software has been used in situations where the number of bidders is between 12 to 350 with the average being around 120. The number of lanes (objects) has ranged between 500 and 10,000. Additionally, each lane bid can contain a capacity constraint as well as a budget capacity constraint covering multiple lanes. The typical number of lanes is 3000. OptiBid$^{\mathrm{TM}}$ does not limit the number of distinct subsets that bidders bid on or the number of items allowed within a package. OptiBid$^{\mathrm{TM}}$ is based on an integer program solver with a series of proprietary formulations and starting heuristic algorithms.[16]

SAITECH-INC's bidding software, SBID, is also based on integer programming. They report being able to handle problems of similar size as OptiBid$^{\mathrm{TM}}$.[17]

Exact methods for CAP2 have been proposed by Rothkopf *et al.* (1998), Fujishima *et al.* (1999), Sandholm (1999) and Andersson *et al.* (2000). The first uses straight dynamic programming, while the second and third use refinements by substantially pruning the search tree and introducing additional bounding heuristics. Andersson *et al.* use integer programming. In the second, the method is tested on randomly generated instances the

[15] However, these crew scheduling problems give rise to instances of SPA that have a large number of duplicate columns in the constraint matrix. In some cases as many as 60% of them. We thank Dr. Márta Eső for alerting us to this.

[16] We thank Dr. Christopher Caplice of Logistics.com for providing this information.

[17] We thank Dr. Yoshiro Ikura of SAITECH-INC for providing us with this information.

largest of which involved 500 items (rows) and 20,000 bids (variables). The third also tests the method on randomly generated instances, the largest of which involved 400 items (rows) and 2000 bids (variables). In these tests the number of bids examined is far less than the number of subsets of objects. The last uses integer programming methods on the test problems generated by the second and third.

By comparison, a straightforward implementation on a commercially available code for solving linear integer programs (called CPLEX) runs into difficulties for instances of CAP involving more than 19 items if one puts nonzero bids on *all* subsets. There will be more than 2^{19} variables. This already requires one giga-byte of memory to store. CPLEX can handle in this straight forward approach on the order of 2^{19} variables and 19 constraints before running out of resident memory. Notice that this is large enough to handle the test problems considered by Sandholm (1999), Fujishima *et al.* (1999).

Andersson *et al.* (2000)[18] point out that CPLEX dominates (in terms of run times) the algorithms of Sandholm (1999) and appear competitive with Fujishima *et al.* (1999). As pointed out by Andersson *et al.* (2000) and de Vries and Vohra (2001), solution times can be sensitive to problem structure. For this reason Leyton-Brown *et al.* (2000a) are compiling a suite of test problems.

3.6 Approximate Methods

One way of dealing with hard integer programs is to give up on finding the optimal solution. Rather one seeks a feasible solution fast and hopes that it

[18] We reported in earlier versions of this survey about our own experiments. This part became now an independent report, see de Vries and Vohra (2001).

is near optimal. This raises the obvious question of how close to optimal the solution is. There have traditionally been three ways to assess the accuracy of an approximate solution. The first is by worst-case analysis, the second by probabilistic analysis and the third empirically.

Before proceedings it is important to say that probably every heuristic approach for solving general integer programming problems has been applied to the SPP. Unfortunately, there has not been a comparative testing across such methods to determine under what circumstances a specific method might perform best. We think it safe to say that anything one can think of for approximating the SPP has probably been thought of. In addition, one can embed approximation algorithms within exact algorithms so that one is attempting to get a sharp approximation to the lower bound for the problem at the same time that one iteratively tightens the upper bound.

3.6.1 Worst-Case Analysis

The SPP is difficult to approximate in a worst-case sense. A major result by Håstad (1999) is that unless ZPP = NP (this assumption is actually weaker than P=NP, but as well believed to be unlikely), there is no polynomial time algorithm for the SPP that can deliver a worst case ratio larger than $n^{\epsilon-1}$ for any $\epsilon > 0$.[19] On the positive side polynomial algorithms that have a worst case ratio of $O(n/(\log n)^2)$, see Boppana and Halldórsson (1992) are known.[20] Bounds that are a function of the data of the underlying input problem are also known. A recent example of this motivated by CAP1 is given by Akcoglu *et al.* (2000). The reader interested in a full account

[19] Recall that in CAP1, n would be the number of bids.

[20] In contrast, SPC can be approximated to within a factor of $\log n$. In this sense SPC is ‘easier’ than SPP.

of what is known about approximating the SPP should consult Crescenzi *et al.* (1998) where an updated list of what is known about the worst-case approximation ratio of a whole range of optimization problems is given.

When interpreting these worst case results it should be remembered that they shed little light on the 'typical' accuracy of an approximation algorithm.

3.6.2 Probabilistic Analysis

Probabilistic analysis is an attempt to characterize the typical behavior of an approximation algorithm. A probability distribution over problem instances is specified. This induces a distribution over the value of the optimal as well as approximate solution. The goal is to understand how close these two distributions might be. Since the results are asymptotic in nature, attention must be paid to the convergence results when interpreting the results. A problematic feature is that the distributions over instances that are chosen (because of ease of analysis) do not necessarily coincide with the distributions that actual instances will be drawn from.[21] This issue arises also in the empirical testing of approximation algorithms.

3.6.3 Empirical Testing

Many approximation algorithms will be elaborate enough to defy theoretical analysis. For this reason it is common to resort to empirical testing. Further empirical testing allows one to consider issues not easily treated analytically.

A good guide to the consumption of an empirical study of approximation algorithms is given by Ball and Magazine (1981). They list the following evaluation criteria:

[21] Results that assert that 'typical' instances are hard are very rare.

1. Proximity to the optimal solution.

2. Ease of implementation (coding and data requirements).

3. Flexibility; ability to handle changes in the model.

4. Robustness; ability to provide sensitivity analysis and bounds.

This is not the forum for an extensive discussion of the issues associated with the empirical testing of heuristics. However, some points are worth highlighting.

The most obvious is the choice of test problems. Are they realistic? Do they exhibit the features that one thinks one will find in the environment? Interestingly, probabilistic analysis has a role to play here in eliminating some schemes for randomly generating test problems. For example it is known that certain generation schemes give rise to problems that are easy to solve; for example, a randomly generated solution is with high probability close to optimal. Success on a collection of problems generated in this way conveys no information. Is the accuracy due to the approximation algorithm or the structure of the test problems?

Some approximation algorithms involve a number of parameters that need to be fine tuned. Comparing their performance with heuristics whose parameters are not fine tuned becomes difficult because it is not clear whether one should include the overhead in the tuning stage in the comparison.

4. Iterative Auctions

Iterative auctions come in two varieties (with hybrids possible). In the first, bidders submit, in each round, prices on various allocations. The auctioneer makes a provisional allocation of the items that depends on the submitted prices. Bidders are allowed to adjust their price offers from the previous rounds and the auction continues. Such auctions come equipped with rules to ensure rapid progress and encourage competition. Iterative auctions of this type seem to be most prevalent in practice.

In the second type, the auctioneer sets the price and bidders announce which bundles they want at the posted prices. The auctioneer observes the requests and adjusts the prices. The price adjustment is usually governed by the need to balance demand with supply.

Call auctions of the first type *quantity setting*, because the auctioneer sets the allocation or quantity in response to the prices/bids set by bidders. Call the second *price setting* because the auctioneer sets the price. Quantity setting auctions are harder to analyze because of the freedom they give to bidders. Each bidder determines the list of bundles as well as prices on same to announce. In price setting auctions, each bidder is limited to announcing which bundles meet their needs at the announced prices.

In many simple environments price setting and quantity setting auctions can be viewed as being 'dual' to one another. The simplest example is the auction of a single object. The popular English ascending auction is an example of a quantity setting auction. Bidders submit prices in succession, with the object tentatively assigned to the current highest bidder. The auction terminates when no one is prepared to top the current high bid. The 'dual' version to this auction has the auctioneer continuously raising the price. Bidders signal their willingness to buy at the current price by keeping their hands raised. The auction terminates the instant a single bidder remains with his hand raised. In fact this dual version of the English auction is used as a stylized model of the English auction itself for the purposes of analysis (see for example Klemperer 2000). We believe that price setting auctions are useful stylized models of quantity setting auctions and that insights from one apply to the other.

Our discussion of iterative auctions is motivated by this 'duality'. We will point out that price setting auctions can be viewed as primal-dual algorithms for the underlying winner determination problem. The reverse will also be true. Primal dual algorithms for CAP1 (or CAP2) can be given a price setting auction interpretation. Dantzig (1963), specifically offers an auction interpretation for the decomposition algorithm for linear programming. A more recent example is Bertsekas (1991), who has proposed a collection of dual based algorithms for the class of linear network optimization problems. These algorithms he dubs auction algorithms. Auction interpretations of algorithms for optimization problems go back at least as far as Walras (see Chapter 17H of the book by Mas-Collel *et al.* 1995) and all have the same flavor. Dual variables are interpreted as prices and the updates on their value that are executed in these algorithms can be interpreted as a form of myopic best response on the part of bidders.

What are the advantages of iterative auctions over, say, single round sealed bid auctions? The first is that they save bidders from specifying their bids for every possible combination in advance. Second, such methods can be adapted to dynamic environments where bidders and objects arrive and depart at different times. Third, in settings where bidders have private information that is relevant to other bidders, such auctions (with appropriate feedback) allow that information to be revealed.

Examples of iterative approaches for solving the CAP are given by Fujishima *et al.* (1999), Rassenti *et al.* (1982), Parkes (2000a), and Bikhchandani *et al.* (2001). In the same spirit, Brewer (1999), Wellman *et al.* (1998)

and Kutanoglu and Wu (1998) propose decentralized scheduling procedures in different contexts. In their set up the auctioneer chooses a feasible solution and 'bidders' are asked to submit improvements to the solution. In return for these improvements, the auctioneer agrees to share a portion of the revenue gain with the bidder.

In order to understand the behavior of price setting auctions it is important to identify what properties prices must have in order to produce an allocation that solves CAP1 (or CAP2). Such an understanding can be derived from the duality theory of integer programs.

4.1 Duality in Integer Programming

To describe the dual to SPP let $\mathbf{1}$ denote the m-vector of all 1's and a^j the j^{th} column of the constraint matrix A. The (superadditive) dual to SPP is the problem of finding a superadditive, non-decreasing function $F : \mathbb{R}^m \to \mathbb{R}^1$ that solves

$$\min F(\mathbf{1})$$

$$\text{s.t. } F(a^j) \geq c_j \ \forall j \in V$$

$$F(\mathbf{0}) = 0$$

We can think of F as being a non-linear price function that assigns a price to each bundle of goods (see Wolsey 1981).

If the primal integer program has the integrality property, there is an optimal integer solution to its linear programming relaxation, the dual function F will be linear i.e. $F(u) = \sum_i y_i u_i$ for some y and all $u \in \mathbb{R}^m$. The dual becomes:

$$\min \sum_i y_i$$

$$\text{s.t.} \sum_i a_{ij} y_i \geq c_j \ \forall j \in V$$
$$y_i \geq 0 \ \forall i \in M$$

That is, the superadditive dual reduces to the dual of the linear programming relaxation of SPP. In this case we can interpret each y_i to be the price of object i. Thus an optimal allocation given by a solution to the CAP can be supported by prices on individual objects.

Optimal objective function values of SPP and its dual coincide (when both are well defined). There is also a complementary slackness condition:

Theorem 4.1 (Nemhauser and Wolsey 1988) *If x is an optimal solution to SPP and F an optimal solution to the superadditive dual then*

$$(F(a^j) - c_j)x_j = 0 \ \forall j.$$

Solving the superadditive dual problem is as hard as solving the original primal problem. It is possible to reformulate the superadditive dual problem as a linear program (the number of variables in the formulation is exponential in the size of the original problem). For small or specially structured problems this can provide some insight. The interested reader is referred to Nemhauser and Wolsey (1988) for more details. In general one relies on the solution to the linear programming dual and uses its optimal value to guide the search for an optimal solution to the original primal integer program. One way to do it is with a technique known as Lagrangean Relaxation.

4.2 Lagrangean Relaxation

The basic idea is to 'relax' some of the constraints of the original problem by moving them into the objective function with a penalty term. That is infeasible solutions to the original problem are allowed, but they are penalized in the objective function in proportion to the amount of infeasibility. The constraints that are chosen to be relaxed, are selected so that the optimization problem over the remaining set of constraints is in some sense easy. We describe the bare bones of the method first and then give a 'market' interpretation of it.

Recall the SPP:

$$Z = \max \sum_{j \in V} c_j x_j$$

$$\text{s.t.} \quad \sum_{j \in V} a_{ij} x_j \leq 1 \ \forall i \in M$$

$$x_j = 0, 1 \ \forall j \in V$$

Let Z_{LP} denote the optimal objective function value to the linear programming relaxation of SPP. Note that $Z \leq Z_{LP}$. Consider now the following relaxed problem:

$$Z(\lambda) = \max \sum_{j \in V} c_j x_j + \sum_{i \in M} \lambda_i (1 - \sum_{j \in V} a_{ij} x_j)$$

$$\text{s.t. } 1 \geq x_j \geq 0 \ \forall j \in V$$

For a given λ, computing $Z(\lambda)$ is easy. To see why note that

$$\sum_{j \in V} c_j x_j + \sum_{i \in M} \lambda_i (1 - \sum_{j \in V} a_{ij} x_j) = \sum_{j \in V} (c_j - \sum_{i \in M} \lambda_i a_{ij}) x_j + \sum_{i \in M} \lambda_i.$$

Thus, to find $Z(\lambda)$, simply set $x_j = 1$ if $(c_j - \sum_{i \in M} \lambda_i a_{ij}) > 0$ and zero otherwise. It is also easy to see that $Z(\lambda)$ is piecewise linear and convex. A basic result that follows from the duality theorem of Linear Programming is:

Theorem 4.2

$$Z_{LP} = \min_{\lambda \geq 0} Z(\lambda).$$

Evaluating $Z(\lambda)$ for each λ is a snap, if one can find a fast way to determine the λ that solves $\min_{\lambda \geq 0} Z(\lambda)$ one would have a fast procedure to find Z_{LP}. The resulting solution (values of the x variables) while integral need not be feasible. However it may not be 'too infeasible' and so could be fudged into a feasible solution without a great reduction in objective function value.

Finding the λ that solves $\min_{\lambda \geq 0} Z(\lambda)$ can be accomplished using the subgradient algorithm. Suppose the value of the lagrange multiplier λ at iteration t is λ^t. Choose any subgradient of $Z(\lambda^t)$ and call it s^t. Choose the lagrange multiplier for iteration $t+1$ to be $\lambda^t + \theta_t s^t$, where θ_t is a positive number called the step size. In fact if x^t is the optimal solution associated with $Z(\lambda^t)$,

$$\lambda^{t+1} = \lambda^t + \theta_t(Ax^t - \mathbf{1}).$$

Notice that $\lambda_i^{t+1} > \lambda_i^t$ for any i such that $\sum_j a_{ij} x_j^t > 1$. The penalty term is increased on any constraint currently being violated.

For an appropriate choice of step size at each iteration, this procedure can be shown to converge to the optimal solution. Specifically, $\theta_t \rightarrow 0$ as $t \rightarrow \infty$ but $\sum_t \theta_t$ diverges. Ygge (1999) describes some heuristics for determining the multipliers in the context of winner determination.

Here is the auction interpretation. The auctioneer chooses a price vector λ for the individual objects and bidders submit bids. If the highest bid, c_j, for the j^{th} bundle exceeds $\sum_{i \in M} a_{ij}\lambda_i$, this bundle is tentatively assigned to that bidder. Notice that the auctioneer need not know what c_j is ahead of time. This is supplied by the bidders after λ is announced. In fact, the bidders need not announce bids, they could simply state which individual objects are acceptable to them at the announced prices. The auctioneer can randomly assign objects to bidders in case of ties. If there is a conflict in the assignments, the auctioneer uses the subgradient algorithm to adjust prices and repeats the process.

Now let us compare this auction interpretation of Lagrangean relaxation with the simultaneous ascending auction (SAA) proposed by P. Milgrom, R. Wilson and P. McAfee (see Milgrom 1995). In the SAA, bidders bid on individual items simultaneously in rounds. To stay in the auction for an item, bids must be increased by a specified minimum from one round to the next just like the step size. Winning bidders pay their bids. The only difference between this and Lagrangean relaxation, is that the bidders through their bids adjust their prices rather than the auctioneer. The adjustment is along a subgradient. Bids increase on those items for which there are two or more bidders competing.

One byproduct of the SAA is called the *exposure problem.* Bidders pay too much for individual items or bidders with preferences for certain bundles drop out early to limit losses. As an illustration consider an extreme example of a bidder who values the bundle of goods i and j at \$100 but each separately at \$0. In the SAA, this bidder may have to submit high bids on i and j to be able to secure them. Suppose that it loses the bidding on i. Then it is left standing with a high bid j which it values at zero. The presence of such a problem is easily seen within the Lagrangean relaxation framework. While Lagrangean relaxation will yield the optimal objective function value for the *linear relaxation* of the underlying integer program, it is not guaranteed to produce a feasible solution. Thus the solution generated may not satisfy the complementary slackness conditions. The violation of complementary slackness is the *exposure problem* associated with this auction scheme. To see why, notice that a violation of complementary slackness means

$$\sum_{i\in M} a_{ij}\lambda_i > c_j \text{ and } x_j = 1.$$

Hence the sum of prices exceeds the value of the bundle that the agent receives. Notice that any auction scheme that relies on prices for individual items alone will face this problem.

In contrast to the SAA outlined above is the Adaptive User Selection Mechanism (AUSM) proposed by Banks *et al.* (1989). AUSM is asynchronous in that bids on subsets can be submitted at any time and so is difficult to connect to the Lagrangean ideas just described. An important feature of AUSM is an arena which allows bidders to aggregate bids to exploit synergies. DeMartini *et al.* (1999) propose an iterative auction scheme that is a hybrid of the SAA and AUSM that is easier to connect to the Lagrangean framework. In this scheme, bidders submit bids on packages rather than on individual items. Like the SAA, bids on packages must be increased by a specified amount from one round to the next. This minimum increment is a function of the bids submitted in the previous round. In addition, the number of items that a bidder may bid on in each round is limited by the number of items s/he bid on in previous rounds. The particular implementation of this scheme advanced by DeMartini *et al.* (1999) can also be given a Lagrangean interpretation. They choose the multipliers (which can be interpreted as prices on individual items) so as to try to satisfy the complementary slackness conditions of linear programming. Given the bids in each round, they allocate the objects so as to maximize revenue. Then they solve a linear program (that is essentially the dual to CAP1) that finds a set of prices/multipliers that approximately satisfy the complementary slackness conditions associated with the allocation.

Wurman and Wellman (2000) propose an iterative auction that allows bids on subsets but uses anonymous, non-linear prices to 'direct' the auction. Bidders submits bids on bundles and using these bids, an instance of CAP2 is formulated and solved. Then, another program is solved to impute prices to the bundles allocated that will satisfy a complementary slackness condition. In the next round, bidders must submit a bid that is at least as large the imputed price of the bundles.

Kelly and Steinberg (2000) also propose an iterative scheme for combinatorial auctions.[22] The auction has two phases. The first phase is an SAA where bidders bid on individual items. In the second phase an AUSM-like mechanism is used. The important difference is that each bidder submits a (temporary) suggestion for an assignment of all the items in the auction. Here a temporary assignment is comprised of previous bids of other players plus new bids of his own.[23]

In Parkes (1999) an iterative auction, called *i*Bundle, that allows bidders to bid on combinations of items and uses non-linear prices is proposed. Bidders submit bids for subsets of items. At each iteration the auctioneer announces prices for those subsets of items that receive unsuccessful bids from agents. For a bid on a subset to be 'legal' it must exceed the price posted by the auctioneer. Given the bids, the auctioneer solves an instance of CAP1 and tentatively assigns the objects. For the next iteration, the prices on each subset are either kept the same or adjusted upwards. The upward adjustment is determined by the highest losing bid for the subset in

[22] The description is tailored to the auction for assigning carrier of last resort rights in telecommunications.

[23] We thank Professor Steinberg for alerting us to an earlier inaccuracy in the description of the procedure.

question plus a user specified increment. The auction terminates when the bids from one round to the next do not show sufficient change. The scheme can be given a Lagrangian interpretation as well, however the underlying formulation is different from CAP1 or CAP2. We discuss the underlying formulation in Section 4.4.

By relaxing on a *subset* of the constraints as opposed to all of them we get different relaxations, some of which give upper bounds on Z that are smaller than Z_{LP}. Details can be found in the book by Nemhauser and Wolsey (1988). Needless to say there have been many applications of Lagrangean relaxation to SPP, SPA and SPC and hybrids with exact methods have also been investigated. See Balas and Carrera (1996) and Beasley (1990) for recent examples.

4.3 Column Generation

Column generation is a technique for solving linear programs with an exceedingly large number of variables. Each variable gives rise to a column in the constraint matrix, hence the name column generation. A naive implementation of a simplex type algorithm for linear programming would require recording and storing every column of the constraint matrix. However, only a small fraction of those columns would ever make it into an optimal basic feasible solution to the linear program. Further, of those columns not in the current basis, one only cares about the ones whose reduced cost will be of the appropriate sign. Column generation exploits this observation in the following way. First an optimal solution is found using a subset of the columns/variables. Next, given the dual variable implied by this preliminary solution, an optimization problem is solved to find a non-basic column/variable that has a reduced cost of appropriate sign. The trick is to design an optimization problem to find this non-basic column without listing all non-basic columns.

Here we propose that the column generation idea can be implemented in an auction setting. In the first step the auctioneer chooses an extreme point solution to the CAP. It does not matter which one, any one will do. Note that this initial solution could involve fractional allocations of objects.

This extreme point solution together with the reduced costs is reported to all bidders. Each bidder, looking only at how they value the allocation proposes a column/variable/subset to enter the basis (along with its value to the bidder). The proposed column and its valuation must satisfy the appropriate reduced cost criterion for inclusion in the basis. In effect each bidder is being used as a subroutine to execute the column generation step. In the worst case, the bidder's pricing problem might be NP-complete however the the bidder knows how his valuation is structured. So to the bidder it might be computationally simpler to solve the pricing problem knowing the underlying structure than for the auctioneer who does not. Further, this eases the communication requirements between bidder and auctioneer and permits bidders to reveal only as little of their valuation as is necessary to determine whether they win anything.

The auctioneer now gathers up the proposed columns (along with their valuations) and using these columns and the columns from the initial basis only (and possibly previously generated nonbasic columns), solves a linear program to find a revenue maximizing (possibly fractional) allocation. The new extreme point solution generated is handed out to the bidders who are asked to each identify a new column (if any) to be added to the new basis that meets the reduced cost criterion for inclusion. The process is then repeated until an extreme point solution is identified that no bidder wishes to modify. To avoid cycling, the auctioneer can always implement one of the standard anti-cycling rules for linear programming.

This auction procedure eliminates the need to transmit and process long lists of subsets and their bids. Bids and subsets are generated only as needed. Second, the bidders are provided an opportunity to challenge an allocation provided they propose an alternative that increases the revenue to the seller. If the bids might lead to a nonintegral allocation, then this column generation has to be imbedded into a branch and cut/price scheme to produce an integer solution.[24]

Notice that the ellipsoid method provides a way to solve the fractional CAP to optimality in polynomial time while generating only a polynomially bounded number of columns. So if the fractional CAP turns out to be integral, CAP itself can be solved in polynomial time. On the other hand, Nisan (2001) showed, that even for submodular valuations the computation of the efficient outcome requires exponential communication.

4.4 Cuts, Extended Formulations and Non-linear Prices

The decentralized methods described above work by conveying 'price' information to the bidders. Given a set of bids and an allocation, prices for individual items that 'support' or are 'consistent' with the bids and allocations are derived and communicated to the bidders. Such prices, because they are linear cannot hope to fully capture the interactions between the parties. Here we show, with an example, how cutting plane methods can be used to generate prices that more closely reflect the interactions between

[24] We thank Dr. Márta Eső for suggesting this last refinement. See Eső (1999) for an example of such a branch and cut scheme.

bids on different sets of objects.

In the example we have 6 objects with highest bids on various subsets of objects shown below; subsets with bids of zero are not shown:

$$b(\{1,2\}) = b(\{2,3\}) = b(\{3,4\}) = b(\{4,5\}) = b(\{1,5,6\}) = 2,\ b(\{6\}) = 1.$$

Formulation CAP2 for this example (ignoring the integrality constraints) is:

$$\begin{array}{llllllll}
\max & 2x_{12}+ & 2x_{23}+ & 2x_{34}+ & 2x_{45}+ & 2x_{156}+ & x_6 & \\
s.t. & x_{12}+ & & & & x_{156} & & \leq 1 \\
 & x_{12}+ & x_{23} & & & & & \leq 1 \\
 & & x_{23}+ & x_{34} & & & & \leq 1 \\
 & & & x_{34}+ & x_{45} & & & \leq 1 \\
 & & & & x_{45}+ & x_{156} & & \leq 1 \\
 & & & & & x_{156}+ & x_6 & \leq 1 \\
 & x_{12}, & x_{23}, & x_{34}, & x_{45}, & x_{156}, & x_6 & \geq 0
\end{array}$$

The optimal *fractional* solution is to set all variables equal to a half. The optimal dual variables are $y_i = 1/2$ for $i = 1, \ldots, 5$ and $y_6 = 1$. So, for example, the imputed price of the set $\{1,2\}$ is $y_1 + y_2 = 1$.

Consider now the following inequality:

$$x_{12} + x_{23} + x_{34} + x_{45} + x_{156} \leq 2.$$

Every feasible *integer* solution to the formulation above satisfies this inequality but not all fractional solutions do. In particular the optimal fractional solution above does not satisfy this inequality. This inequality is an example of a cut. Classes of cuts for the SPP are known, the one above belongs to the class of odd-cycle cuts.

Now append this cut to our original formulation:

$$\begin{array}{llllllll}
\max & 2x_{12}+ & 2x_{23}+ & 2x_{34}+ & 2x_{45}+ & 2x_{156}+ & x_6 & \\
s.t. & x_{12}+ & & & & x_{156} & & \leq 1 \\
 & x_{12}+ & x_{23} & & & & & \leq 1 \\
 & & x_{23}+ & x_{34} & & & & \leq 1 \\
 & & & x_{34}+ & x_{45} & & & \leq 1 \\
 & & & & x_{45}+ & x_{156} & & \leq 1 \\
 & & & & & x_{156}+ & x_6 & \leq 1 \\
 & x_{12}+ & x_{23}+ & x_{34}+ & x_{45}+ & x_{156} & & \leq 2 \\
 & x_{12}, & x_{23}, & x_{34}, & x_{45}, & x_{156}, & x_6 & \geq 0
\end{array}$$

The optimal solution to this linear program is *integral.* It is $x_{12} = 1, x_{34} = 1$ and $x_6 = 1$. There are now 7 dual variables. One for each of the six objects (y_i) and one more for the cut (μ). One optimal dual solution is $y_1 = y_5 = y_6 = 0$, $y_2 = y_3 = y_4 = 1$ and $\mu = 1$. The imputed price for the set $\{1, 2\}$ is now $y_1 + y_2 + \mu = 2$. In general the price of a set S will be the sum of the item prices, $\sum_{i \in S} y_i$ plus μ if the 'x' variable associated with the set S appears with coefficient 1 in the cut. Notice that pricing sets of objects in this way means that the price function will be superadditive.

It is instructive to compare the imputed price of the set $\{1, 2\}$ in the two formulations. The first formulation assigns a price of one to the set. The second a higher price. The first formulation ignores the fact that if the set $\{1, 2\}$ is assigned to a bidder, the sets $\{1, 5, 6\}$ and $\{2, 3\}$ cannot be assigned to anyone else. This fact is captured by the cut. The dual variable associated with the cut can be interpreted as the associated opportunity cost of assigning the set $\{1, 2\}$ to a bidder. Thus the actual price of the set $\{1, 2\}$ is the sum of the prices of the objects in it plus the opportunity cost associated with its sale.

Cuts can be derived in one of two ways. The first is by purely combinatorial reasoning (see Padberg (1973, 1975 and 1979), Cornuejols and Sassano (1989) and Sassano (1989)) and the other through an algebraic technique introduced by Ralph Gomory (see Nemhauser and Wolsey (1988) for the details). For CAP1 or CAP2, given a fractional extreme point, one can use the Gomory method to generate a cut involving only the variables that are basic in the current extreme point. This is useful for computational purposes as one does not have to lug all variables around to identify a cut. Second, the new inequality will be a non-negative linear combination of the current basic rows less equal than a non-negative number. Thus the dual variable associated with this new constraint will have an additive effect on the prices of various subsets as in the example.

The reader will notice that by picking an extreme point dual solution, the imputed prices for some sets are zero. Since there is some flexibility in the choice of dual variables, one can choose an interior (to the feasible region) dual solution. Other choices are possible. Incentive considerations suggest choosing the one that minimizes the prices bidders pay (see for example Bikhchandani and Ostroy 1998).

Yet another way to get non-linear prices is by starting with a stronger formulation of the underlying optimization problem. One formulation is stronger than another if its set of feasible (fractional) solutions is strictly contained in the other. In the example above, the second formulation is stronger than the first. Both formulations share the same set of integer solutions, but not fractional solutions. The set of fractional solutions to the second formulation is a strict subset of the fractional solutions to the first one.

Stronger formulations can be obtained, as shown above, by the addition of inequalities. Yet another, standard way, of obtaining stronger formulations is through the use of additional or auxiliary variables, typically a large number of them. Geometrically, one is treating the problem formulated in the original set of variables as the projection of a higher dimensional but structurally simpler polyhedron. Formulations involving such additional variables are called extended formulations and developing these extended formulations is called lifting. Using lifting one can develop a hierarchy of successively stronger formulations of the underlying integer program.

There is a close connection between lifting, extended formulations and cutting planes. Perhaps the most accessible introduction to these matters is Balas *et al.* (1993).

In the auction context, Bikhchandani and Ostroy (1998), propose a number of extended formulations for the problem of selecting the winning set of bids. To describe the first of their extended formulations, let Π be the set of all possible partitions of the objects in the set M. If π is an element of Π, we write $S \in \pi$ to mean that the set $S \subset M$ is a part of the partition π. Let $z_\pi = 1$ if the partition π is selected and zero otherwise. These are the auxiliary variables. Using them Bikhchandani and Ostroy (1998) can reformulate CAP1 as follows:

$$\max \sum_{j \in N} \sum_{S \subseteq M} b^j(S) y(S, j)$$

$$\text{s.t.} \sum_{S \subseteq M} y(S, j) \leq 1 \; \forall j \in N$$

$$\sum_{j \in N} y(S, j) \leq \sum_{\pi \ni S} z_\pi \; \forall S \subset M$$

$$\sum_{\pi \in \Pi} z_\pi \leq 1$$

$$y(S, j) = 0, 1 \;\forall S \subseteq M,\; j \in N$$

$$z_\pi = 0, 1 \;\forall \pi \in \Pi$$

Call this formulation CAP3. In words, CAP3 chooses a partition of M and then assigns the sets of the partition to bidders in such a way as to maximize revenue. It is easy to see that this formulation is stronger than CAP1 or CAP2. Fix an $i \in M$ and add over all $S \ni i$ the inequalities

$$\sum_{j \in N} y(S, j) \le \sum_{\pi \ni S} z_\pi \;\forall S \subset M$$

to obtain:

$$\sum_{S \ni i} \sum_{j \in N} y(S, j) \le 1 \;\forall i \in M$$

which are the inequalities that appear in CAP1. While stronger than CAP1, formulation CAP3 still admits fractional extreme points (Bikhchandani and Ostroy 1998).

The dual of the linear relaxation of CAP3 involves one variable for every constraint of the form:

$$\sum_{S \subseteq M} y(S, j) \le 1 \;\forall j \in N,$$

call it s_j, which can be interpreted as the surplus that bidder j obtains. The dual involves one variable for every constraint of the form:

$$\sum_{j \in N} y(S, j) \le \sum_{\pi \ni S} z_\pi \;\forall S \subset M$$

which we will denote p_S. It can be interpreted as the price of the subset S. In fact the dual will be:

$$\min \sum_{j \in N} s_j + \mu$$

$$\text{s.t. } s_j \ge b^j(S) - p_S \;\forall j \; S \subset M$$

$$\mu \geq \sum_{S \in \pi} p_S \ \forall \pi \in \Pi$$

$$s_j, p_S, \mu \geq 0$$

and has the obvious interpretation: minimizing the bidders surplus plus μ. Thus one can obtain non-linear prices from the extended formulation. These prices do not support the optimal allocation since CAP3 is not integral. Further they do not depend on the bidders, that is all bidders pay the same price for a given subset. The catch of course is this formulation involves many more variables than CAP1 or CAP2.

In Parkes and Ungar (2000) a condition on bidders preferences is identified that ensures that the linear relaxation of CAP3 has an integral solution. The condition, called bid safety, is difficult to interpret but has the effect of forcing complementary slackness to hold for an integer solution of CAP3. Under this condition any algorithm for solving CAP3's dual (or its lagrangean relaxation) will generate an optimal solution of CAP3 itself. Since many dual algorithms can be given an auction interpretation with the iterations being identified as adjustments in bids that a myopic best reply agent might execute, one can generate auction schemes that are arguably optimal. This is precisely the tack taken in Parkes and Ungar (2000) to support the adoption of the *i*Bundle auction scheme of Parkes (1999).

Bikhchandani and Ostroy (1998) introduce yet another formulation stronger than CAP3 which is integral. The idea is to use a variable that represents both a partition of the objects *and* an allocation. Essentially one variable for every solution. The dual to this formulation gives rise to non-linear prices with the twist that they are bidder specific. Different bidders pay different prices for the same subset. Bikhchandani *et al.* (2001) propose and investigate another extended integral formulation with significantly fewer variables than the one in Bikhchandani and Ostroy (1998).

5. Incentive Issues

Thus far we have focused on the problem of choosing an allocation of the objects so as to maximize the seller's revenue. The revenue depends on the bids submitted but there is no guarantee that the submitted bids approximate the actual values that bidders assign to the various subsets. To illustrate how this can happen consider three bidders, 1, 2 and 3, and two objects $\{x, y\}$. Suppose:

$$v^1(x, y) = 100,\ v^1(x) = v^1(y) = 0,\ v^2(x) = v^2(y) = 75,\ v^2(x, y) = 0,$$

$$v^3(x) = v^3(y) = 40,\ v^3(x, y) = 0.$$

Here $v^i(\cdot)$ represents the value to bidder i of a particular subset. Notice that the bid that i submits on the set S, $b^i(S)$, need not equal $v^i(S)$.

If the bidders bid truthfully, the auctioneer should award x to 2 and y to 3, say, to maximize his revenue. Notice however that bidder 2 say, under the assumption that bidder 3 continues to bid truthfully, has an incentive to shade his bid down on x and y to, say, 65. Notice that bidders 2 and 3 still win but bidder 2 pays less. This argument applies to bidder 3 as well. However, if they both shade their bids downwards they can end up losing the auction. This feature of combinatorial auctions is called the 'threshold problem' (see Bykowsky *et al.* 1995): a collection of bidders whose combined valuation for distinct portions of a subset of items exceeds the bid submitted on that subset by some other bidder. It may be difficult for them to coordinate their bids to outbid the large bidder on that subset. The basic problem is that the bidders 2 and 3 must decide how to divide $75 + 40 - 100$

between them. Every split can be rationalized as the equilibrium of an appropriate bargaining game. In linear programming terms, the threshold problem arises because of a multiplicity of optimal dual solutions.

In this section we describe what is known about auction mechanisms that give bidders the incentive to truthfully reveal their valuations. [25]

To discuss incentive issues we need a model of bidders preferences. The simplest conceptual model endows bidder $j \in N$ with a list $\{v^j(S)\}_{S \subseteq M}$ with $v^j(\emptyset) = 0$, abbreviated to v^j, that specifies how she values (monetarily) each subset of objects. Thus $v^j(S)$ represents how much bidder j values the subset S of objects.[26]

The auction scheme chosen and the bids submitted will be a function of the beliefs that seller and bidders have about each other. The simplest model of beliefs is the independent private values model and it is the model we will restrict ourselves to. Each bidder's v^j is assumed by seller and all bidders to be an independent draw from a commonly known distribution over a compact, convex set. Bidder j knows her v^j but not the valuations of the other bidders. Last, bidders and seller are assumed to be risk neutral.

To continue the discussion it will be useful to distinguish between two popular objectives the auctioneer may have. The first is economic efficiency and the second is revenue maximization.

[25] For more details on game theory, equilibria and mechanism design see Fudenberg and Tirole (1992).

[26] In the language of mechanism design, this list of valuations becomes the bidders **type**. In this case, since the type is not a single number it is called multi-dimensional. For an introduction to mechanism design see Chapter 7 of Fudenberg and Tirole (1992).

5.1 Economic Efficiency

An auction is economically efficient if the allocation of objects to bidders chosen by the seller solves the following:

$$\max \sum_{j \in N} \sum_{S \subseteq M} v^j(S) y(S,j)$$

$$\text{s.t.} \sum_{S \ni i} \sum_{j \in N} y(S,j) \leq 1 \ \forall i \in M$$

$$\sum_{S \subseteq M} y(S,j) \leq 1 \ \forall j \in N$$

$$y(S,j) = 0, 1 \ \forall S \subseteq M, \ j \in N$$

Notice that this is just CAP1 with b^i replaced by v^i. The optimal objective function value of this integer program is an upper bound on the revenue that the seller can achieve if no bidder bids above their valuation. The fact that the seller uses an auction that selects an allocation that solves this integer program does not imply that the seller achieves this revenue.[27]

An auctioneer interested in producing an efficient allocation has a puzzle. Since bidders valuations are private information, he must solve the optimization problem above without a knowledge of the objective function! Remarkably, there is a sealed bid auction that implements the efficient outcome. It does so because it is a weakly dominant strategy for bidders to bid truthfully in the auction. The most general class of such auctions was characterized by Clarke (1971) and Groves (1973). A special case was identified earlier by William Vickrey (1961) in an auction that bears his name. The version we describe here is sometimes known as as Vickrey-Clarke-Groves

[27] In Myerson (1981) it is shown that the revenue maximizing auction for a single good is not guaranteed to be efficient. See Jehiel and Moldovanu (1999) for a more pronounced version of the same.

(VCG) scheme. It is proved in Krishna and Perry (1997) (see also Williams (1999) for the same result under slightly different assumptions) that in the independent private values model, amongst all auctions that implement the efficient allocation, the VCG scheme maximizes the revenue to the seller.

It works as follows:

1. Agent j reports v^j. There is nothing to prevent agent j from misrepresenting themselves. However, given the rules of the auction, it is a weakly dominant strategy to bid truthfully.

2. The seller chooses the allocation that solves:

$$V = \max \sum_{j \in N} \sum_{S \subseteq M} v^j(S) y(S,j)$$

$$\text{s.t. } \sum_{S \ni i} \sum_{j \in N} y(S,j) \leq 1 \ \forall i \in M$$

$$\sum_{S \subseteq M} y(S,j) \leq 1 \ \forall j \in N$$

$$y(S,j) = 0,1 \ \forall S \subseteq M, \ j \in N$$

Call this optimal allocation y^*

3. To compute the payment that each bidder must make let, for each $k \in N$,

$$V^{-k} = \max \sum_{j \in N \setminus k} \sum_{S \subseteq M} v^j(S) y(S,j)$$

$$\text{s.t. } \sum_{S \ni i} \sum_{j \in N \setminus k} y(S,j) \leq 1 \ \forall i \in M$$

$$\sum_{S \subseteq M} y(S,j) \leq 1 \ \forall j \in N \setminus k$$

$$y(S,j) = 0,1 \ \forall S \subseteq M, \ j \in N \setminus k$$

Denote by y^k the optimal solution to this integer program. Thus y^k is the efficient allocation when bidder k is excluded.

4. The payment that bidder k makes is equal to

$$V^{-k} - [V - \sum_{S \subseteq M} v^k(S) y^*(S,k)].$$

Thus bidder k's payment is the difference in 'welfare' of the other bidders without him and the welfare of others when he is included in the allocation. Notice that the payment made by each bidder to the auctioneer is non-negative.

If a seller were to adopt the VCG scheme her total revenue would be

$$\sum_{k \in N} V^{-k} - \sum_{k \in N} [V - \sum_{S \subseteq M} v^k(S) y^*(S,k)]$$
$$= \sum_{k \in N} \sum_{S \subseteq N} v^k(S) y^*(S,k) + \sum_{k \in N} (V^{-k} - V)$$
$$= V + \sum_{k \in N} (V^{-k} - V).$$

If there were a large number of agents then no single agent can have a significant effect, i.e., one would expect that, on average, V is very close in value to V^{-k}. Thus the revenue to the seller would be close to V, the largest possible revenue that any auction could extract. To solidify this intuition we need that for all agents k that their valuation v^k is superadditive, i.e. $v^k(A) + v^k(B) \leq v^k(A \cup B)$ for all $k \in N$ and $A, B \subset M$ such that $A \cap B = \emptyset$. With this assumption we can find the efficient allocation using CAP2. Thus:

$$V = \max \sum_{S \subset M} \{\max_{j \in N} v^j(S)\} x_S$$

$$\text{s.t.} \sum_{S \ni i} x_S \leq 1 \ \forall i \in M$$

$$x_S = 0, 1 \ \forall S \subset M$$

and

$$V^{-k} = \max \sum_{S \subset M} \{ \max_{j \in N \setminus k} v^j(S) \} x_S$$

$$\text{s.t.} \ \sum_{S \ni i} x_S \leq 1 \ \forall i \in M$$

$$x_S = 0, 1 \ \forall S \subset M$$

Notice now that if the number $|N|$ of bidders is large and given that the v^j's live in a compact set, the random variable $\max_{j \in N} v^j(S)$ is very close on average to $\max_{j \in N \setminus k} v^j(S)$.[28] Hence the objective function of the program that defines V is essentially the same as the objective function of the integer program that defines V^{-k}. This argument is made precise in Monderer and Tennenholtz (1999), where it is shown in the model used here that the VCG scheme generates a revenue for the seller that is asymptotically close to the revenue from the optimal auction.

The VCG scheme is, in general, impractical to implement, if the number of bidders is very large. However in some circumstances this difficulty can be avoided. Hershberger and Suri (2001), for example, show that in the network routing context at most two optimization problems must be solved to compute Vickrey payments. Bikhchandani *et al.* (2001), show that in a wide range of situations, the problem of finding the efficient allocation can be formulated as a linear program. What is more, optimal dual variables in this linear program coincide with the Vickrey payments. In these instances, two —and in many instances even only one— optimization problems must be solved to compute the Vickrey payments.

[28] In fact the difference of the two is essentially the difference between the first and second order statistic of a large collection of independent random numbers from a compact set.

Another way to overcome the computational difficulties is to replace y^* and y^k for all $k \in N$ with approximately optimal solutions. Such a modification in the scheme does not in general preserve incentive compatibility (see Nisan and Ronen 2000). In Lehmann *et al.* (1999) such a direction is taken. They solve the embedded optimization problems using a greedy type algorithm and show that the resulting scheme is not incentive compatible. However if one is willing to restrict bidders valuations drastically it is possible to generate schemes based on the greedy algorithm that are incentive compatible. Lehmann *et al.* (1999) call this restriction "single mindedness." Each bidder values only one subset and no other.

Even if one is willing to relax incentive compatibility, an approximate solution to the underlying optimization problems in the VCG can lead to other problems. There can be many different solutions to an optimization problem whose objective function values are within a specified tolerance of the optimal objective function value. The payments specified by the VCG scheme are very sensitive to the choice of solution. Thus the choice of approximate solution can have a significant impact on the payments made by bidders. This issue is discussed by Johnson *et al.* (1997) in the context of an electricity auction used to decide the scheduling of short term electricity needs. Through simulations they show that variations in near-optimal schedules that have negligible effect on total system cost can have significant consequences on the total payments by bidders.

The VCG scheme as described is a single round sealed bid auction. In simple settings the VCG auction has an iterative counterpart that implements the efficient outcome in the sense that it is a Nash equilibrium for bidders to bid truthfully in each round. Such iterative auctions reduce the cognitive burden on bidders to list their valuations for all bundles. Also, they reduce the amount of information that bidders must reveal to the auctioneer. An example of such can be found in Ausubel (2000) where an ascending auction for indivisible heterogenous objects under the gross substitutes on preferences assumption is proposed that implements the efficient outcome. Bikhchandani and Ostroy (1998) and Bikhchandani *et al.* (2001) show that in many environments the problem of finding the efficient allocation can be formulated as a linear program in such a way that dual variables correspond to Vickrey payments.[29] The primal-dual algorithm for these linear programs produce an iterative auction that implements the outcome of the VCG auction. That is, bidding truthfully in each round is a Nash equilibrium.

Experience with the VCG scheme in field settings is limited. Isaac and James (1998) report on an experiment using the VCG scheme for a combinatorial auction involving three bidders and two objects. On the basis of their results they argue that the VCG scheme can be operationalized and, in their words, 'achieve high allocative efficiency'. Kagel and Levin (2001) experiment with iterative auctions for the sale of multiple units of homogeneous goods. They compare the uniform price auction with an iterative auction (due to Ausubel 1997) that implements the VCG outcome. In their experiments Ausubel's auction results in outcomes close to the efficient one. Hobbs *et al.* (2000) explore the possibility that the VCG scheme is vulnerable to collusion. It is pointed out by these authors in environments with repeated interactions that not only are there many opportunities for collusion amongst bidders but incentive compatibility of the VCG scheme cannot be guaranteed. However, this is a weakness not unique to VCG.

The efficient auction when bidders values are interdependent is more

[29] In fact the envirnoments when this is possible are charaterized in terms of bidders preferences.

difficult. Jehiel *et al.* (1999) and Jehiel and Moldovanu (2000) discuss the underlying mathematical problems. Eső and Maskin (1999) introduce a restriction on preferences which they call partition preferences. The collection of subsets that a bidder assigns positive value to form a partition of the set of objects.[30] For this restricted case they derive results about the form of the efficient auction.

5.2 Revenue Maximization

The problem of designing an auction that maximizes the auctioneers revenue (optimal auction) is more difficult. Our discussion of this problem will be restricted to the case of independent private values. For simplicty we will suppose that each bidder assumes that the other bidders value functions are independent draws from a *finite* set V of value functions with commonly known distribution F. The probability that agent j has value function u will be denoted $F(u)$. The probability that the n-tuple $\mathbf{v} = (v^1, \ldots, v^n) \in V^n$ is realized is denoted $\Pi_{j=1}^n F(v^j)$. For convenience this will be abbreviated to $F(\mathbf{v})$.

One can imagine a variety of elaborate and involved auction protocols that must be considered. However, the revelation principle (see Myerson 1981) allows one to restrict attention to a **direct revelation** scheme. In such a scheme the auctioneer announces how he will allocate the objects amongst the bidders and the payments he will extract from each as a function of the announced value functions. Then bidders are asked to announce their value functions. The auctioneer is constrained to choose an allocation rule and payment function that satisfy two constraints:

[30] Note that problem CAP is still NP-hard under this restriction.

1. Induce bidders to reveal their actual valuations (**incentive compatibility**).

2. Each bidder's expected payoff is non-negative.

An **allocation rule** is a mapping from an n-tuple of value functions to an integer solution to CAP1. If A is an allocation rule and $\mathbf{v} = (v^1, v^2, \ldots, v^n)$ an n-tuple of value functions, we will write $A(\mathbf{v})$ to mean the allocation selected by the rule A.

A **payment rule** is a mapping from an n-tuple of value functions to an n-tuple of payments, one for each bidder. If P is a payment rule and $\mathbf{v} = (v^1, v^2, \ldots, v^n)$ an n-tuple of value functions, we will write $P_j(\mathbf{v})$ to mean the payment that bidder j must make.

With these variables one can formulate the problem of finding a revenue maximzing auction as a mathematical program.[31]

There is one decision variable that represents the choice of allocation rule and another to represent the choice of payment rule. The objective is

$$\max_{A,P} \sum_{\mathbf{v} \in V^n} F(\mathbf{v})[\sum_{j \in N} P_j(\mathbf{v})].$$

Incentive compatibility requires that for each $j \in N$ with value function v^j and all $u \neq v^j$:

$$\sum_{\mathbf{v^{-j}} \in V^{n-1}} [v^j(A(v^j, \mathbf{v^{-j}}) - P_j(v^j, \mathbf{v^{-j}})]F(\mathbf{v^{-j}}) \geq \sum_{\mathbf{v^{-j}} \in V^{n-1}} [v^j(A(u, \mathbf{v^{-j}}) - P_j(u, \mathbf{v^{-j}})]F(\mathbf{v^{-j}}).$$

[31] This is just a a specialization of the usual formulation for optimal auctions with multidimensional types. The survey paper by Rochet and Stole (2000) has more details on this subject. No closed form solution to the problem of optimal auctions with multidimensional types is known and it is unlikely that any exists.

That is, the expected utility from truthfully reporting ones value function should exceed the expected utility from reporting some other value function. Given incentive compatibility we can write the individual rationality constraint as

$$\sum_{\mathbf{v}^{-\mathbf{j}} \in V^{n-1}} [v^j(A(v^j, \mathbf{v}^{-\mathbf{j}})) - P_j(v^j, \mathbf{v}^{-\mathbf{j}})] F(\mathbf{v}^{-\mathbf{j}}) \geq 0$$

for all $j \in N$ and value functions v^j.

How might one solve this? First fix a choice for allocation rule, A. Let

$$\nu^j(A, u) = \sum_{\mathbf{v}^{-\mathbf{j}} \in V^{n-1}} v^j(A(u, \mathbf{v}^{-\mathbf{j}})) F(\mathbf{v}^{-\mathbf{j}})$$

and

$$\rho^j(u) = \sum_{\mathbf{v}^{-\mathbf{j}} \in V^{n-1}} P_j(v^j, \mathbf{v}^{-\mathbf{j}}) F(\mathbf{v}^{-\mathbf{j}}).$$

Then we can rewrite the incentive compatibility and individual rationality constraints as follows:

$$\rho^j(v^j) - \rho^j(u) \leq \nu^j(A, v^j), -\nu^j(A, u),$$

and

$$\rho^j(v^j) \leq \nu^j(A, v^j).$$

The objective function can be written as:

$$\sum_{\mathbf{v} \in V^n} F(\mathbf{v})[\sum_{j \in N} P_j(\mathbf{v})] = \sum_{j \in N} \sum_{v \in V} F(v) \rho^j(u).$$

Thus the optimization problem becomes:

$$\max \sum_{j \in N} \sum_{v \in V} F(v) \rho^j(v)$$

$$\text{s.t. } \rho^j(v) - \rho^j(u) \leq \nu^j(A, v) - \nu^j(A, u) \ \forall v, u \in V$$

$$\rho^j(v) \leq \nu^j(A, v) \; \forall v \in V.$$

Since the allocation rule A is fixed, the only variables are the ρ's. There are at most two of them per constraint with coefficients of +1 and -1. Hence, given A, the problem of finding a payment rule that enforces incentive compatibility and individual rationality is a network flow problem (or more precisely the dual to one).[32] For each possible choice of allocation rule A, we can determine via the duality theorem of linear programming whether a payment rule that is incentive compatible exists.

To determine the optimal auction one has to solve this constrained optimization problem for each possible allocation rule. Then pick the allocation rule that yields the largest revenue. Thus the design of the optimal auction may be extremely sensitive to both the choice of bidders value function as well as distribution of type. Under fairly stringent conditions some results have been derived as well as comparisons of revenue between auctions of different kind. Rochet and Stole (2000) summarize these results.

Levin (1997) identifies the optimal auction under a more restrictive setting. Specifically, all objects, for all bidders, complement each other and bidders are perfectly symmetric. In this case, the revenue maximizing auction is to simply bundle all the objects together and auction the bundle off using an optimal single item auction.

Krishna and Rosenthal (1996), again with a simplified model of preferences, attempt to make revenue comparisions between different auction schemes. They consider auctions involving two items where bidders are of two kinds. One kind, that they call local, are interested in receiving a single

[32] Introduce one vertex for each value function in V. For each ordered pair (v, u), introduce an arc directed from v to u with length $\nu^j(A, v), -\nu^j(A, u)$. The dual problem is to find the shortest path tree rooted at the source.

item. The other, called global, have valuations for the entire set that are superadditive. In this paper they identify equilibria for a two item sequential second-price auction, (a particular) simultaneous second-price auction, and a combinatorial second-price auction. The revenues obtained from each auction (under the identified equilibria) are compared numerically. They observe that if the synergy is small then the sequential second-price auction generates the largest revenue. For larger synergies the simultaneous second-price auction generates the largest revenue. In their simulations the combinatorial second-price auction always generates the smallest revenue. It is not known whether this relation is a function of the equilibrium found or the specification of the simultaneous auction.

Cybernomics, Inc. (2000) went a step further by comparing a particular simultaneous multi-round auction with a particular multi-round combinatorial auction. They performed experiments for additive values, and for valuations with synergies of small, medium or high intensity. Their experiments indicate, that the combinatorial multi-round auction was always superior with respect to efficency but the revenue was smaller and it took more rounds of longer duration to finish the auction.

6. Summary

Our survey has had three goals. The first, a pedestrian one, has been to survey the extant literature. The second, has been to point out 'classical' results that apply directly to the problem of designing combinatorial auctions. The third has been to emphasize the connections between the duality theory of optimization problems and the design of auctions.

Acknowledgements

We thank David Parkes, Sunil Chopra, Michael Rothkopf and John Ledyard and annonymous referees for many useful comments and suggestions.

REFERENCES

A. Andersson, M. Tenhunen and F. Ygge, “Integer Programming for Combinatorial Auction Winner Determination”, *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS00)*, 39–46, 2000.

K. Akcoglu, J. Aspnes, B. DasGupta and M. Kao, “An Opportunity-Cost Algorithm for Combinatorial Auctions”, manuscript, 2000.

L. M. Ausubel, “An Efficient Ascending-Bid Auction for Multiple Objects”, manuscript, University of Maryland, 1997.

L. M. Ausubel, “An Efficient Dynamic Auction for Heterogeneous Commodities”, manuscript, University of Maryland, 2000.

L. M. Ausubel, P. Cramton, “Demand Reduction and Inefficiency in Multi-Unit Auctions”, manuscript, University of Maryland, 1998.

L. M. Ausubel, P. Cramton, R. Preston McAfee and J. McMillan, “Synergies in Wireless Telephony: Evidence from the Broadband PCS Auction”, *Journal of Economics and Management Strategy*, 6, 3, 497–527, 1997.

E. Balas, S. Ceria and G. Cornuejols, “A Lift-and-Project Cutting Plane Algorithm for mixed 0-1 Programs”, *Mathematical Programming*, 58, 295–324, 1993.

E. Balas and M. C. Carrera, “A Dynamic Subgradient-based Branch-and-

Bound Procedure for Set Covering", *Operations Research*, 44, 875–890, 1996.

E. Balas and A. Ho, "Set Covering Algorithms Using Cutting Planes, Heuristics, and Subgradient Optimization: A Computational Study", *Mathematical Programming*, 12, 37–60, 1980.

E. Balas and M. W. Padberg, "Set Partitioning: A Survey", *SIAM Review*, 18, 710–760, 1976.

M. O. Ball and M. Magazine, 1981, "The Design and Analysis of Heuristics", *Networks*, 11, 215–219, 1981.

J. S. Banks, J. O. Ledyard and D. P. Porter "Allocating Uncertain and Unresponsive Resources: An experimental approach", *Rand Journal of Economics*, vol. 20, 1, 1–25, 1989.

C. Barnhart, E. D. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh and P. H. Vance "Branch and Price: Column Generation for Solving Huge Integer Programs", *Operations Research*, 46, 316–329, 1998.

J. E. Beasley, "A Lagrangian Heuristic for Set-Covering Problems", *Naval Research Logistics*, 37, 151–164, 1990.

D. P. Bertsekas, Linear Network Optimization: Algorithms and Codes, MIT press, Cambridge, MA, 1991.

S. Bikhchandani and Chi-fu Huang, "The Economics of Treasury Securities Markets", *Journal of Economic Perspectives*, 7, 117–134, 1993.

S. Bikhchandani, S. de Vries, J. Schummer and R. V. Vohra, "Linear Programming and Vickrey Auctions", to appear in, "Mathematics of the Internet: E-Auction and Markets", Springer, "The IMA Volumes in Mathematics and its Applications", Vol. ??, 2001.

S. Bikchandani and J. M. Ostroy, "The Package Assignment Model", manuscript, 1998.

S. Bikchandani and J. Mamer, "Competitive Equilibrium in an Exchange Economy with Indivisibilities", *Journal of Economic Theory*, vol. 74, no. 2, 385–413, 1997.

R. B. Boppana and M. M. Halldórsson, "Approximating maximum independent sets by excluding subgraphs", *BIT*, 32(2):180–196, 1992.

P. J. Brewer, "Decentralized Computation Procurement and Computational Robustness in a Smart Market", *Economic Theory*, 13, 41–92, 1999.

M. M. Bykowsky, R. J. Cull and J. O. Ledyard, "Mutually Destructive Bidding: The FCC Auction Design Problem", *Journal of Regulatory Economics*, 17, 205–228, 2000.

C. G. Caplice, "An Optimization Based Bidding Process: A New Framework for Shipper-Carrier Relationships", Thesis in the field of Transportation and Logistics Systems submitted to the Department of Civil and Environmental Engineering, School of Engineering, MIT, June 1996.

E. Clarke, "Multipart Pricing of Public Goods", *Public Choice*, 8, 19–33, 1971.

Cybernomics, Inc., "An Experimental Comparison of the Simultaneous Mlti-Round Auction and the CRA Combinatorial Auction", `http://combin.fcc.gov/98540193.pdf`, 2000

G. Cornuejols and A. Sassano, "On the 0-1 Facets of the Set Covering Polytope", *Mathematical Programming*, 43, 45–56, 1989.

P. Cramton, "Spectrum Auctions", **Handbook of Telecommunications Economics**, Martin Cave, Sumit Majumdar, and Ingo Vogelsang, eds., Amsterdam: Elsevier Science B.V., forthcoming, 2001.

P. Crescenzi and V. Kann, "A compendium of NP optimization problems", `http://www.nada.kth.se/theory/compendium/`, manuscript, 1998.

G. B. Dantzig, Linear Programming and Extensions, Princeton University Press, New Jersey, 1963.

A. Davenport and J. Kalagnanam, "Price negotiations for procurement of direct inputs", IBM Technical Report 98774 2001.

C. DeMartini, A. Kwasnica, J. Ledyard and D. Porter, "A New and Improved design for Multi-Object Iterative Auctions", manuscript, 1999.

S. de Vries and R. V. Vohra, "A solver for winnerdetermination for Combinatorial Auctions", manuscript, 2001.

M. Eső, "Parallel Branch and Cut for Set Partitioning", Ph.D. thesis, School of Operations Research and Industrial Engineering, Cornell University, 1999.

P. Eső and E. Maskin, "Multi-Good Efficient Auctions with Multi-Dimensional Information", manuscript, 1999.

D. Fudenberg and J. Tirole, Game Theory, MIT Press, 1992.

Y. Fujishima, K. Leyton-Brown and Y. Shoham, "Taming the Computational Complexity of Combinatorial Auctions: Optimal and Approximate Approaches", *Proceedings of IJCAI-99*, Stockholm, 1999.

R. Gonen and D. Lehmann, "Optimal Solutions for Multi-Unit Combinatorial Auctions: Branch-and-Bound Heuristics", *ACM Conference on Electronic Commerce*, 2000.

R. Graves, J. Sankaran and L. Schrage, "An Auction Method for Course

Registration", *Interfaces*, 23, 5, 1993.

M. Grötschel, L. Lovász and A. Schrijver, Geometric Algorithms and Combinatorial Optimization, Springer-Verlag, 1988.

T. Groves, "Incentives in Teams," *Econometrica*, 41, 617–631, 1973.

P. C. Gilmore and R. E. Gomory, "A Linear Programming Approach to the Cutting Stock Problem", *Operations Research*, 9, 849–859, 1961.

M. C. Golumbic and P. L. Hammer, "Stability in circular arc graphs", *J. Algorithms*, 9, 314–320, 1988.

F. Gul and E. Stachetti, "English and Double Auctions with Differentiated Commodities", *Journal of Economic theory*, 92, 66–95, 2000.

J. Håstad, "Clique is hard to approximate within $n^{1-\epsilon}$", *Acta Mathematica*, 182, 105–142, 1999.

J. Hershberger and S. Suri, "Vickrey Pricing in Network Routing: Fast Payment Computation", manuscript, 2001.

B. F. Hobbs, M. Rothkopf, L. Hyde and R. P. O'Neill, "Evaluation of a Truthful Revelation Auction in the Context of Energy markets with Nonconcave Benefits", *Journal of Regulatory Economics*, 18, 5–32, 2000.

S. van Hoesel and R. Müller, "Optimization in Electronic Markets: Examples in Combinatorial Auctions", *Netnomics*, forthcoming, 2000.

K. Hoffman and M. W. Padberg, "Solving Airline Crew Scheduling Problems by Branch and Cut", *Management Science*, 39, 657–682, 1993.

ILOG Inc. CPLEX Division, 930 Tahoe Blvd. # 802-279, Incline Village, NV 89451-9436, USA, "Using the CPLEX Callable Library" (information available at URL `http://www.cplex.com`) 1997.

R. Mark Isaac and D. James, “Robustness of the Incentive Compatible Combinatorial Auction”, manuscript, 1998.

C. Jackson, “Technology for Spectrum Markets”, Ph. D. Thesis submitted to the Department of Electrical Engineering, School of Engineering, MIT, June 1976.

P. Jehiel, B. Moldovanu and E. Stacchetti, “Multidimensional Mechanism Design for Auctions with Externalities”, *Journal of Economic Theory* 85(2), 258–294, 1999.

P. Jehiel, B. Moldovanu, “Efficient Design with Inderdependent Valuations”, to appear in Econometrica, 2001.

P, Jehiel and B. Moldovanu, “A Note on Revenue Maximization and Efficiency in Multi-Object Auctions”, discussion paper, University College London, 1999.

R. B. Johnson, S. S. Oren, and A. J. Svoboda, “Equity and Efficiency of Unit Commitment in Competitive Electricity Markets”, *Utilities Policy*, 6(1), 9–20, 1997.

J. H. Kagel and D. Levin, “Behavior in Multi-Unit Demand Auctions: Experiments with Uniform Price and Dynamic Vickrey Auctions”, *Econometrica*, 69, 2001.

J. M. Keil, “On the complexity of scheduling tasks with discrete starting times”, *Operations Research Letters*, 12, 293–295, 1992.

A. S. Kelso and V. P. Crawford, “Job matching, Coalition Formation and Gross Substitutes”, *Econometrica*, 50, 1483–1504, 1982.

F. Kelly and R. Steinberg, “A Combinatorial Auction with Multiple Winners for Universal Service”, *Management Science*, 46(4), 586–596, 2000.

V. Krishna and M. Perry, “Efficient Mechanism Design”, manuscript, 1998.

V. Krishna and R. W. Rosenthal, “Simultaneous Auctions with Synergies”, *Games and Economic Behavior*, 17, 1–31, 1996.

E. Kutanoglu and S. D. Wu, “On Combinatorial Auction and Lagrangean Relaxation for Distributed Ressource Scheduling”, *IIE Transactions*, 31, 813–826, 1999.

D. Lehmann, L. O’Callaghan and Y. Shoham, “Truth Revelation in Rapid, Approximately Efficient Combinatorial Auctions”, manuscript, 1999.

J. Levin, “An Optimal Auction for Complements”, *Games and Economic Behavior*, 18, 176–192, 1997.

K. Leyton-Brown, Y. Shoham and M. Tennenholtz, “An Algorithm for Multi-Unit Combinatorial Auctions”, *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, 2000.

K. Leyton-Brown, M. Pearson and Y. Shoham, “Towards a Universal Test suite for Combinatorial Auction Algorithms, *ACM Conference on Electronic Commerce*, 2000a.

A. Mas-Collel, M. D. Whinston and J. R. Green, Microeconomic Theory, Oxford Univ. Press, NY, 1995.

P. Milgrom, “Auctioning the Radio Spectrum” in *Auction Theory for Privatization*, Cambridge University Press, 1995.

D. Monderer and M. Tennenholtz, “Asymptotically Optimal Multi-Object Auctions for Risk-Averse Agents”, manuscript, 2000.

M. Naor, B. Pinkas and R. Sumner, “Privacy Preserving Auctions and Mechanism Design”, *ACM Conference on Electronic Commerce*, 1999.

R. Myerson, “Optimal Auction Design”, *Mathematics of Operations Research*, 6, 58–73, 1981.

G. L. Nemhauser and L. A. Wolsey, Integer and Combinatorial Optimization, John Wiley, New York, 1988.

N. Nisan, “Bidding and Allocation in Combinatorial Auctions”, *ACM Conference on Electronic Commerce*, 2000.

N. Nisan, “The Communications Complexity of Combinatorial Auctions”, manuscript, 2000.

N. Nisan and A. Ronen, “Computationally Feasible VCG Mechannisms”, *ACM Conference on Electronic Commerce*, 2000.

M. Olson, D. Porter, J. Ledyard, J. Swanson and D. Torma, “The First Use of a Combined Value Auction for Transportation Services”, Social Science Working Paper No. 1093, California Institute of Technology, March 2000.

M. W. Padberg, “On the facial structure of set packing polyhedra”, *Mathematical Programming*, 5, 199–215, 1973.

M. W. Padberg, “Perfect zero-one matrices”, *Mathematical Programming*, 6, 180–196, 1974.

M. W. Padberg “On the complexity of the set packing polyhedra”, *Annals of Discrete Mathematics* 1, 421–434, 1975.

M. W. Padberg, “Covering, Packing and Knapsack Problems”, *Annals of Discrete Mathematics*, 4, 265–287, 1979.

M. Parker and J. Ryan “A column generation algorithm for bandwidth packing”, *Telecommunications Systems*, 2, 185–196, 1994.

D. C. Parkes, "*i*Bundle: An Efficient Ascending Price Bundle Auction", *Proceedings ACM Conference on Electronic Commerce (EC-99)*, Denver, 148–157, November 1999.

D. C. Parkes and L. Ungar, "Iterative Combinatorial Auctions: Theory and Practice", *Proceedings 17th Nat. Conf. on Artificial Intelligence*, 74–81, 2000(a).

D. C. Parkes and L. Ungar, "Preventing Strategic Manipulation in Iterative Auctions: proxy Agents and Price Adjustment", *Proceedings 17th Nat. Conf. on Artificial Intelligence*, 82–89, 2000 (b).

S. J. Rassenti, V. L. Smith and R. L. Bulfin, "A Combinatorial Auction Mechanism for Airport Time Slot Allocation", *Bell Journal of Economics*, vol. 13, No. 2, 402–417, 1982.

J.-C. Rochet and L. Stole, "The Economics of Multidimensional Screening", manuscript, 2000.

M. H. Rothkopf, A. Pekec and R. M. Harstad, "Computationally Manageable Combinatorial Auctions", *Management Science*, vol. 44, No. 8, 1131–1147, 1998.

T. Sandholm, "An Algorithm for Optimal Winner Determination in Combinatorial Auctions", *Proceedings of IJCAI-99*, Stockholm, 542–547, 1999.

A. Sassano, "On the Facial Structure of the Set Covering Polytope", *Mathematical Programming*, 44, 181–202, 1973.

A. Schrijver, Theory of Linear and Integer Programming, John Wiley, New York, 1986.

S. Srinivasan, J. Stallert and A. B. Whinston, "Portfolio Trading and Electronic Networks", manuscript, 1998.

M. Strevell and P. Chong, "Gambling on Vacation", *Interfaces*, vol. 15, March-April, 63-67, 1985.

W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders", *Journal of Finance*, 16, 8–37, 1961.

M. P. Wellman, W. E. Walsh, P. R. Wurman and J. K. MacKie-Mason, "Auction Protocols for Decentralized Scheduling", *Games and Economic Behavior*, 35(1-2), 271-303, 2001.

S. R. Williams, "A Characterization of Efficient, Bayesian Incentive Compatible Mechanisms", *Economic Theory*, vol. 14, 155–180, 1999.

L. Wolsey, "Integer Programming Duality: Price Functions and Sensitivity Analysis", *Mathematical Programming*, vol. 20, 173–195, 1981.

P. R. Wurman and M. Wellman, "A Progressive, Anonymous-Price Combinatorial Auction", *Proceedings Second ACM Conference on Electronic Commerce*, 21-29, Minneapolis, October, 2000.

F. Ygge, "Improving the Computational Efficiency of Combinatorial Auction Algorithms", EnerSearch Public Report:EnS 99:08, 1999.

M. Yokoo, Y. Sakurai and S. Matsubara, "Robust Combinatorial Auction Protocol against False-name Bids", *Artificial Intelligence Journal*, to appear, 2001.