Computationally Manageable Combinational Auctions

Michael H. Rothkopf • Aleksandar Pekeč • Ronald M. Harstad
Faculty of Management and Rutgers Center for Operations Research, Rutgers University, New Brunswick, New Jersey 08903-5062
*BRICS, Department of Computer Science, University of Aarhus, Denmark**
Faculty of Management and Rutgers Center for Operations Research, Rutgers University, New Brunswick, New Jersey 08903-5062

There is interest in designing simultaneous auctions for situations such as the recent FCC radio spectrum auctions, in which the value of assets to a bidder depends on which other assets he or she wins. In such auctions, bidders may wish to submit bids for combinations of assets. When this is allowed, the problem of determining the revenue maximizing set of nonconflicting bids can be difficult. We analyze this problem, identifying several different structures of permitted combinational bids for which computational tractability is constructively demonstrated and some structures for which computational tractability cannot be guaranteed.
(*Spectrum Auctions; Combinatorial Auctions; Multi-Item Simultaneous Auctions; Bidding with Synergies; Computational Complexity*)

1. Introduction

Some auctions sell many assets simultaneously. Often these assets, like U.S. treasury bills, are interchangeable. However, sometimes the assets and the bids for them are distinct. This happens frequently, as in the U.S. Department of the Interior's simultaneous sales of off-shore oil leases, in some private farmland auctions, and in the Federal Communications Commission's recent multi-billion-dollar sales of the rights to use radio spectrum. It also happened in the post-World-War-II divestiture of synthetic rubber plants by the U.S. government. In such situations, the value of an asset to a bidder may depend strongly on which other assets he or she wins. In off-shore oil-lease bidding, this dependency often takes the form of diseconomies of scale.[1] However, in each of the other examples there are clearly situations in which the value of an asset is increased if another asset or group of assets is won. For example, in the radio spectrum auctions, a license for the Philadelphia region may be much more valuable to a company if that company also has licenses for the New York and / or the Washington / Baltimore regions.[2]

Because of the possibility of such synergy or superadditivity in values, the designers of simultaneous sales have reason to consider allowing bids not just for individual assets, but also single bids for combinations of assets. Off-shore oil lease sales have not allowed such bids. Some farmland sales do (Schackmann 1989), and the rubber divestiture ultimately did.[3] Recently, a new

* Now at Fuqua School of Business, Duke University, Durham, North Carolina 27708-0120.

[1] The diseconomies of scale are due to risks associated with the extreme variability of returns and the large amounts of money involved. See Rothkopf (1977) for a discussion of bidding in simultaneous auctions with a constraint on exposure—i.e., the total of all bids. Such constraints are imposed on off-shore oil lease bidding teams by their managements to control risks.

[2] Efficiencies in marketing to adjacent regions and in the cost of providing "roaming privileges" may be involved as well as general economies of scale.

[3] As described by McCurdy (1981) and Rothkopf (1983), the divestiture process for synthetic rubber plant units was originally crafted by Congress without such bids, but when Shell Chemical Company under McCurdy submitted a single bid for three units that exceeded the sum of bids from others for those units, Congress voted to accept the Shell bid.

0025-1909/98/4408/1131$05.00

exchange for Southern California pollution rights run by Sholtz and Associates and bidding in the United Kingdom for school bus contracts (Mercer and Tielin 1996) have allowed combinational bids.

Combinational bids (occasionally called package bids and sometimes given the misleading name combinatorial bids) have the potential advantage of allowing bidders to express their synergistic values. Bykowski et al. (1995) present some artificial but suggestive examples in which allowing combinational bids increases both revenue and efficiency. In a specialized model, Harstad and Rothkopf (1995) find that efficiency requires permitting combinational bids. A somewhat different special model by Branco (1995) finds that both a seller seeking to maximize revenue and a public-sector agency seeking an efficient outcome must typically allow combinational bids.

There are two potential disadvantages to allowing combinational bids. The first of these, a "threshold" problem (inaptly called the "free-rider" problem in Milgrom and Wilson (1993)), can occur when diseconomies of scale predominate. For example, consider four bidders competing for two assets: A and B. Suppose that each bidder's valuation is private information and that no bidder values any asset except as follows: A is worth \$100 to bidder 1; B is worth \$100 to bidder 2; and the combination AB is worth \$150 to bidder 3 and \$110 to bidder 4. In this situation, the most economic allocation is to give bidder 1 asset A and bidder 2 asset B. However, if the bids for individual assets A and B are each at \$1, neither of these bidders acting unilaterally can afford the \$110 necessary to match a bid of, say, \$111 for the combination AB by bidder 3. It is worth noting that the threshold problem has an opposing force, the "exposure" problem. This problem is that an unsuccessful attempt to acquire a collection of assets, when combinational bidding is not allowed, may lead to paying more for some individual assets than they are worth. Alternatively, a bidder unwilling to risk bidding above his individual valuations on individual assets may not be able to obtain a combination for which synergies make him the efficient recipient. This arises in the above example with a single change that makes AB worth \$250 to bidder 3. If synergies are important, the efficiency consequences of the exposure problem are likely to outweigh those of the threshold problem. Both problems can be magnified when many assets are sold and large combinations are sought.

The second potential difficulty with allowing combinational bids is the computational difficulty of finding the best (i.e., revenue-maximizing) set of winning bids. This issue was raised in the FCC auction design debate by McAfee (1993). In the worst case, the bid-taker offering n assets could receive bids on 2^n-1 different combinations of assets. Clearly, bid evaluation could present a computational problem when n is large. For examples, the FCC has sold, each time in a single simultaneous auction, 99 radio spectrum licenses, 493 licenses, 1,020 licenses, and recently (August 1996–January 1997) 1,472 licenses.

This paper considers the computational problem of evaluating combinational bids. A complete analysis of the desirability of certain structures of allowable combinational bids would have to go further than this paper. Manageability of winner determination is, however, the logical first step. Consideration of the revenue impact of permitting some combined bids depends on an analysis of how allowing combinational bidding affects bidders' behavior. A necessary first step toward modeling bidder behavior is determining how large a bid on a particular combination must be to avoid losing, given a current set of competing bids for overlapping combinations. We show below that the manageability of this problem of determining a minimal improving (i.e., revenue-increasing) combinational bid is essentially equivalent to the bid-taker's problem that is our focus.

1.1. Background—The FCC Auction

There is a great deal of material describing the FCC auction design and the debate over it. An interested reader should consult McMillan (1994) and the extensive material in the FCC's Docket No. 93.-253.[4] The FCC

[4] Among those contributing were Paul Milgrom and Robert Wilson for Pacific Bell and Nevada Bell, Robert Weber for TDS, R. Preston McAfee for Airtouch, Robert Harris and Michael Katz for Nynex, Barry J. Nalebuff and Jeremy I. Bulow for Bell Atlantic, Peter Cramton for MCI, and R. Mark Isaac for CTIA. There were also sophisticated rounds of replies to the responses, three conferences to discuss issues in the auction design, and pilot experiments conducted at the California Institute of Technology. McMillan (1994) describes many of the issues. Continuing issues arise in later dockets, including 96-228, but most of the original auction design debate is in 93-253.

adopted a series of general policies with respect to the kinds of auctions it would hold. These policies dictate different sales procedures for different situations. What follows is a general description of the process the FCC adopted for the first major (i.e., multi-billion-dollar) auction ("Broadband MTA"). Eleven auctions have been concluded as of this writing and twelve more tentatively scheduled for 1997; the auction process has so far been adjusted only in minor details.

The FCC has sold a given set of licenses in a simultaneous progressive auction with discrete rounds. In these auctions, no sale of any license takes place until the bidding is concluded on all licenses. Bids on individual licenses are binding. (A bidder withdrawing a bid is still subject to covering the FCC's reduced revenue from the withdrawal, if any.) When there is a round of bidding with no bids that increase the price on any license, each of the licenses is sold to the bidder currently holding the high bid on it. In order to keep bidders from holding back, there are "activity rules." Early in the auction, these are relatively lenient, but by the third and final phase of the auction, they are stringent. A bidder is considered active on a license in a round if he or she held the high bid for the license going into the round or raises the bid on the license by at least the minimum amount required for the round.[5] The Broadband MTA ("A/B block") auction began with daily rounds but primarily proceeded at a two-rounds-per-day pace. It ended after 112 rounds, selling 99 licenses for over $7 billion. A follow-up Broadband BTA ("C block," smaller regions of license coverage) auction, concluded after 184 rounds, built up to eight rounds per day as it became less active. Another set of Broadband BTA licenses ("$D/E/F$ block") recently required 276 rounds to complete.

In our opinion, the FCC chose this auction process over one that would permit combinational bids for several reasons. Foremost was a concern with the political consequences of failing to conduct a smoothly flowing auction: it was critical to be able legitimately to dismiss complaints from bidders as only the sour grapes of those who were outbid.[6] This concern led to disallowing combinational bids because the economists' briefs, particularly McAfee (1993), implied that the only choice was between completely disallowing or else permitting all possible combinational bids, and the latter option was fraught with worst-case scenarios of being unable to compute which bids ought to win.

In addition, the FCC had a legitimate concern that bidders might find selecting combinational bids too confusing; even responding to rivals' combinational bids might become a frightful burden for a bidder who had eschewed contemplating his own combinational submissions.[7] Finally, several firms that had brought in influential economists to explain and critique proposed designs may have perceived an incentive to prevent combinational bids, in order to tilt the playing field in their favor. These economists emphasized the threshold problem and downplayed the exposure problem.

1.2. Background—The Auction Design Problem

Only a few scholarly papers consider the auction design problems introduced by value interdependencies among different items for sale. Rothkopf (1977) considers the bidder's problem in a simultaneous auction without combinational bids when there is an interdependency introduced by a constraint on the total of his or her bids. Smith and Rothkopf (1985) consider the bidder's problem in a simultaneous auction without com-

[5] Bidders are allotted five "waivers," which can be used in an individual round to preserve their immediately prior activity status until the next round. Otherwise, in the first stage of the MTA auction, bidders had to remain active on licenses covering a population base that is at least 33% of the total population base for which they wished to remain eligible to bid. In the second stage, they had to remain active on licenses covering at least 67% of their population base. In the final stage, they had to remain active on licenses covering 100% of their eligibility. That is, in the 100% eligibility stage, if a bidder ceases competing for a license for Philadelphia, for example, he or she can immediately switch to bidding on another license or set of licenses, but only if the other license(s) cover a smaller total population. This "one-way street" phenomenon is part of the way the activity rule forces the bidders to bid seriously. It leads bidders to look for ways to "park" eligibility by bidding on licenses they do not want but are confident they will not win. The particular percentages attached to eligibility rules have been tweaked from auction to auction, most notably backing away from 100% in stage 3 to 95–98%.

[6] The FCC's Director of Plans and Policy, Robert Pepper, emphasized "I don't want to be a beta test site." (Nonetheless, he and his staff showed considerable political courage in moving so far from established auction procedures.)

[7] This was not a farfetched concern for anyone who observed bidders' representatives reacting to the demonstration of a continuous-time bidding system, with all combinational bids permitted, at the California Institute of Technology in January 1994.

binational bids when there is a fixed charge incurred if any bid succeeds. Rassenti et al. (1982), Banks et al. (1989) and McCabe et al. (1991) describe laboratory experiments with continuous-time auction mechanisms for simultaneous auctions with interdependent values.[8] Ledyard et al. (1994) describe a combinational bidding mechanism for allocating a space mission's resources to experiments with varying requirements. Bykowsky et al. (1995) discuss "local equilibria" of simple and complex auction mechanisms. Krishna and Rosenthal (1996) consider a simple model where "local" bidders compete for only one of two assets, while "global" bidders compete for both (without permitting combined bids), with a synergy should they win both. Harstad and Rothkopf (1995) demonstrate the advantages of combinational bids in Krishna and Rosenthal's model. Campbell (1996) considers the general question of when incentive-compatible mechanisms can achieve efficiency in environments with synergies. Aside from these papers and the material already cited, we are not aware of any scholarly literature on this topic.

If frictionless aftermarkets existed for the assets being sold, the seller concerned only about efficient allocation would not have to worry about auction form. Almost any form would do since, whatever the initial allocation, the aftermarket would costlessly reallocate the assets efficiently. However, in the common situation in which there are significant transaction costs (and perhaps costly delays) associated with aftermarket transactions, the choice of auction form can affect economic efficiency. In addition, one of the important roles of auctions is to provide legitimacy by being demonstrably fair. The choice of mechanism may matter from that perspective as well.

The auction designer must choose between sequential sale and simultaneous sale, between a progressive process and one-time sealed bids, and between independent sales and allowing combinational bids. If the assets are sold independently, then bid evaluation presents no problem. If the assets are sold using one-time sealed bids, then there is a good deal of time available for determining the winning set of bids. Furthermore, the bid-taker can resort to what we call a "political" solution of the bid selection problem—one that guarantees fairness and is likely to be rather effective as well. This "political" solution involves the bid-taker finding the best feasible solution it can in a reasonable amount of time and then announcing it. All interested parties are then given an opportunity to report a feasible solution with higher revenue. Makers of tentatively losing bids will have an incentive to explore bid combinations including their bids. If each of them is unable to find such a combination in a reasonable amount of time, no one is in a position to challenge the fairness of the bid acceptance process.

If the bidding is progressive, it can either be continuous or involve discrete rounds. If the bidding is continuous, then each new bid can be compared with high bids it must displace. If it exceeds all of them, then it becomes the current leading bid, and the other bids are displaced. This is a simple calculation. However, with combinational bids there is an auction design choice. If unsuccessful bids are not kept available for use in the evaluation, the threshold problem can become quite serious. If they are, then there is a computational problem. In *AUSM* (Adaptive User Selection Mechanism), a computerized combinational auction procedure developed at the California Institute of Technology, this computational problem falls on the bidders. They are given a list of unwithdrawn, currently unsuccessful bids. If they wish, they may incorporate one or more of the bids in this list into their bid.

An example might make this clearer. Suppose that there are four assets, *A*, *B*, *C*, and *D*, for sale; that the current leading bids are $100 for *AB* and $200 for *CD*; and that there are unwithdrawn losing bids of $50 for *A* and $75 for *D*. A bidder for *BC* could combine these two losing bids with its bid of $180 for *BC* to make a winning combination. The auctioneer would accept this bid because it increases the total revenue from $300 to $305. These bids would become the leading bids, and the formerly leading bids for *AB* and *CD* would, until they are withdrawn, be available to other bidders for making combinations of their own. In some variants of the Cal Tech procedure, bidders may submit unsuccessful bids in hopes that others may choose to use them in combination with their own bids. The only limit on the number of such submissions is practical rather than formal.

If there are discrete rounds in the auction, as in the current radio spectrum auctions, then the computa-

[8] Rassenti et al. (1982) also state the integer programming problem that must be solved to select the set of nonconflicting combinational bids that maximizes revenue from the auction.

tional problem of finding the winning bids must be solved by the bid-taker at each stage. It must be solved much more rapidly (in about 10 minutes under the FCC's eight-rounds-per-day bidding schedule), so the "political" solution described above is not available. In this context, the concern of the FCC about allowing combinational bids is understandable. However, the potential computational problems with allowing arbitrary combinational bids do not exist with certain structures of permitted combinational bids. The work presented in this paper is devoted to defining such structures and exploring their limits.

1.3. This Paper: Computability with Limitations on Permitted Combinational Bids

Throughout this paper, the bid-taker's objective is assumed to be revenue maximization. If so, the optimization to be performed when a list of bids is submitted is an integer programming problem, formalized in the next section. If the submitted bids can include a combinational bid on any combination, then this computation becomes unmanageable, at least in worst-case scenarios.

Evaluation of classes of computational problems in terms of worst-case instances is the standard practice in the literature on computational complexity. It is the appropriate way to consider computational problems in auction design if retaining legitimacy of the transaction is the sort of key concern it was with the FCC. This standard practice labels a class of computational problems computationally manageable if an upper bound on computation time for all such problems can be expressed as a polynomial function of the size of the input. Appendix A briefly reviews computational complexity and defines the terminology used in the formal presentation below.

Our principal point is this: While allowing all combinational bids yields a potentially unmanageable (NP-complete) revenue-maximization problem, disallowing combinational bids completely is not the only computationally manageable alternative. We provide several examples of structures of permitted combinational bids and show constructively that they are computationally manageable. For each, the structure is on the border of computational manageability, in the sense that a natural next step reverts to an NP-complete problem. In addition, the problem of finding a minimal winning bid for a bidder who would like to raise a currently losing bid on a particular combination enough to include that combination in an optimal outcome is shown to be computationally manageable whenever the revenue-maximization problem is, and potentially unmanageable whenever that also describes the revenue-maximization problem.

Section 3 considers nested structures. When the set of permitted combinational bids is nested and thus forms a tree structure, a straightforward algorithm for "rolling back" the tree is both fast and transparent. An example demonstrates that rolling back multiple trees of permitted combinational bids can fail to find the revenue-maximal outcome.

Section 4 considers cardinality-based structures—that is, situations in which bids are permitted on combinations meeting size restrictions. Sufficiently strong superadditivity in values may imply that large combinations would be the principal ones for which bidders would wish to submit single bids. Fast and transparent algorithms readily permit bids on large combinations of assets. Allowing bids on arbitrary doubletons reduces to what is essentially the hardest problem for which there is a polynomial-time algorithm; computational manageability would be attained, but not transparency. Allowing combinational bids on tripletons reverts to NP-completeness. Auctions of airport landing and takeoff slots are an example of a setting in which bidding on doubletons may achieve most of the advantages available from combinational bidding.

Section 5 considers geometric structures. For the first case, in which the assets are linearly ordered, like selling cellular telephone rights in Chile, a transparent dynamic algorithm can maximize revenue efficiently when bids are permitted on any interval of consecutive assets.

The situation is more complicated when a representation of the assets requires two dimensions (such as geographic location). We show that permitting bids on arbitrary rectangular subsets might revert to computational unmanageability.

However, some restrictions of 2-dimensional grids of assets can allow useful combinational bidding. For example, if the set of auctioned assets can be organized in a matrix, permitting combinational bids on any row or any column, as would be natural in coin auctions, works well. We generalize this idea in a k-dimensional case (by presenting a set intersection property behind this generalization) and illustrate its limits.

To ease the exposition, longer proofs and formal statements of algorithms are relegated to Appendix B.

2. Formulation of the Problem

The bid-taker's revenue maximization problem can be formulated as follows. Let A denote the set of all individual assets being auctioned. We assume that there are n assets, i.e., $|A| = n$. Any $C \subseteq A$ represents a combination of assets. Auction rules can specify which $C \subseteq A$ are **permitted combinations,** that is, combinations of assets for which bidders may submit a bid. The family of all permitted combinations is denoted by $\mathcal{P}$; that is, $\mathcal{P} = \{C \subseteq A : C \text{ is a permitted combination}\}$. Obviously, $|\mathcal{P}| \leq 2^n$.

In any outcome of a simultaneous auction, the winning combinations must be disjoint since no single asset can be sold more than once. Formally, let $\Omega_{\mathcal{P}}$ denote the set of **outcomes:**

$$\Omega_{\mathcal{P}} := \{\mathcal{W} \subseteq \mathcal{P} : C, C' \in \mathcal{W} \Rightarrow C \cap C' = \emptyset\}. \quad (1)$$

We analyze a single round of an auction. Computational ease in determining winning bids in this case readily leads to a manageable multiround progressive auction by repeating the designed single-round model.

Recall that, throughout this paper, the bid-taker's goal is assumed to be revenue maximization.[9] Also, note that we do not require that every asset be sold.

Let $b(C)$ be the largest bid for the combination C.[10] If there is no bid for C, we set $b(C) = 0$.[11] For any outcome $\mathcal{W}$ (that is, a collection of pairwise disjoint permitted combinations), we define $\text{rev}(\mathcal{W}) := \Sigma_{C\in\mathcal{W}} b(C)$. In other words, $\text{rev}(\mathcal{W})$ is the revenue that the bid-taker would collect if the assets were sold to bidders who submitted the largest bids for the combinations in $\mathcal{W}$.

Our principal focus in this paper is on the bid-taker's problem ($\text{REV}_{\mathcal{P}}$) of determining revenue maximizing outcome $\mathcal{W}_{\text{OPT}}$, that is, the problem of determining $\mathcal{W}_{\text{OPT}} \in \Omega_{\mathcal{P}}$ such that

$$\text{rev}(\mathcal{W}_{\text{OPT}}) = \max\{\text{rev}(\mathcal{W}) : \mathcal{W} \in \Omega_{\mathcal{P}}\}.$$

We call $\mathcal{W}_{\text{OPT}}$ an optimal outcome, and we call any combination $C \in \mathcal{W}_{\text{OPT}}$ a **winning combination.**

The problem of finding an optimal outcome ($\text{REV}_{\mathcal{P}}$) can be formulated as an integer programming problem:

$$\max \sum_{C\in\mathcal{P}} b(C)x_C$$

with the constraints:

$$\forall C \in \mathcal{P} : x_C \in \{0, 1\} \quad \text{and} \quad \forall i \in A: \sum_{C\ni i} x_C \leq \mathbf{1},$$

or, in a more compact form,

$$\max\{\mathbf{b}^T\mathbf{x}: M\mathbf{x} \leq \mathbf{1}, \mathbf{x} \in \{0, 1\}^{|\mathcal{P}|}\}, \quad (2)$$

where M is a 0-1-matrix defined by $m_{i,C} = 1$ if and only if $i \in C$, $\mathbf{b}$ is the vector whose coordinates are $b(C)$, $C \in \mathcal{P}$, and $\mathbf{1}$ is the all ones vector.

The problem of finding an optimal outcome is equivalent to a set-packing problem on a hypergraph $\mathcal{P}$ with weights $b(C)$ for every $C \in \mathcal{P}$. This problem is known to be NP-complete (Karp 1972). However, there are numerous special cases for which the problem is computationally manageable.[12] For example, whenever the matrix M from (2) is totally unimodular, the problem can be solved in polynomial time (see, for example, Schrijver 1986). It is not our intention to list the various special cases of problem (2) that are solvable in polynomial time. This paper concentrates on special classes of hypergraphs that seem to have potential application in practice. For such classes of hypergraphs, we discuss the complexity of the problem of finding an optimal outcome and present efficient algorithms or demonstrate the NP-hardness of the problem. For most classes, we can find a borderline between computational manageability and

[9] This is inessential for the mathematics, which only requires a continuous, quasi-concave objective function, and can readily incorporate political constraints, so long as they can be expressed as computable intersections of half-spaces.

[10] It is computationally trivial to parse a list of submitted bids, removing a bid if it is below the specified minimum price (if any), or if there is a higher bid submitted for the identical combination. We assume the bid-taker has adopted some tie-breaking rule, such as accepting the chronologically earlier bid (the rule used by the FCC).

[11] This can be viewed as the bid-taker submitting a zero bid for every permitted combination C. Hence, when viewed in this way, the bids cover the market.

[12] In other words, for certain families $\mathcal{P}$, there exist a polynomial time algorithm for finding an optimal outcome $\mathcal{W}_{\text{OPT}}$.

potential unmanageability. Our purpose is not to provide new mathematics; most theorems presented here have exact antecedents or can be deduced from antecedents in a straightforward manner. Auction designers should have available, among other tools, a selection of limited structures of permitted combinational bids, without facing the specter of computational nightmares. When an auction designer learns or infers some information about which combinations are likely to yield the principal synergies, it would be useful to attempt finding a structure to permit bidding on these combinations.

The other principal computational problems affected by the choice of the set $\mathcal{P}$ of permitted combinations are those facing bidders. Optimal bidding strategy, like optimal auction design, lies beyond the scope of this paper. However, understanding computational manageability of the problem (REV$_{\mathcal{P}}$) of finding revenue maximizing outcome $\mathcal{W}_{\mathrm{OPT}}$ is a logical precursor to sensibly addressing auction design issues. Both auction design and bidding strategy issues cannot be addressed without understanding the computational complexity of the **minimal winning bid** problem (WIN$_{\mathcal{P}}$): given a permitted combination $C \in \mathcal{P}$, what is the minimal $b_{\mathrm{win}}(C)$ so that C becomes (remains) a winning combination provided that all other bids $b(C')$, $C' \neq C \in \mathcal{P}$ remain unchanged? Even specifying the options to consider in deciding bidding strategy requires understanding the minimal winning bid problem.

The following Proposition (proved in Appendix B) shows that the minimal winning bid problem for C is equivalent to the bid-taker's problem of finding revenue maximizing outcome for a related auction.

PROPOSITION 1. *Let $C_L \notin \mathcal{W}_{\mathrm{OPT}}$ and let $\mathcal{W}^*_{\mathrm{OPT}}$ be an optimal outcome for the auction of $A^* := A \backslash C_L$ with permitted combinations $\mathcal{P}^* := \{C \in \mathcal{P} : C \subseteq A^*\}$. Then*

$$b_{\mathrm{win}}(C_L) = \mathrm{rev}(\mathcal{W}_{\mathrm{OPT}}) - \mathrm{rev}(\mathcal{W}^*_{\mathrm{OPT}}).$$

Similarly, let $C_W \in \mathcal{W}_{\mathrm{OPT}}$ and let $(\mathcal{W}^{**}_{\mathrm{OPT}})$ be an optimal outcome for the auction of $A^{**} := A$ with permitted combinations $\mathcal{P}^{**} := \mathcal{P} \backslash \{C^*\}$. Then

$$b_{\mathrm{win}}(C_W) = b(C_W) - \mathrm{rev}(\mathcal{W}_{\mathrm{OPT}}) + \mathrm{rev}(\mathcal{W}^{**}_{\mathrm{OPT}}).$$

An important consequence of Proposition 1 is the following Corollary.[13]

[13] The proof (see Appendix B) is based on the observation that an algorithm for solving the problem (REV$_{\mathcal{P}}$) can be used to solve the problem (WIN$_{\mathcal{P}}$).

COROLLARY 2. *If the problem of finding revenue maximizing outcome (REV$_{\mathcal{P}}$) is computationally manageable, then, given $C \in \mathcal{P}$, the minimum winning bid problem (WIN$_{\mathcal{P}}$) is also computationally manageable.*

If the number of assets being auctioned, n, is small, determining winning combinations is manageable, but if n is large, this problem for arbitrary $\mathcal{P}$ (e.g., $\mathcal{P} = 2^A$) may be unsolvable for practical purposes. However, for some special classes of $\mathcal{P}$ there exist fast and easy algorithms for finding an optimal outcome of the auction. Even in the most general cases, some combinations need not to be considered as candidates for winning combinations. The following Observation shows that superadditivity plays a straightforward role in determining winning bids. Combination C^* will never be a winning combination if some of the assets in C can be sold for a larger total amount.

OBSERVATION 3. *Let $\mathcal{W}_{\mathrm{OPT}}$ be an optimal outcome and let $C^* \in \mathcal{W}_{\mathrm{OPT}}$ be a winning combination. Let $C^* \supseteq C_1 \cup C_2 \cup \cdots \cup C_k$ where C_i's are pairwise disjoint permitted combinations ($C_1, C_2, \ldots, C_k \in \mathcal{P}$ and $C_i \cap C_j = \emptyset$ for all $1 \leq i < j \leq k$). Then, $b(C^*) \geq \Sigma_{i=1}^{k} b(C_i)$.*

One way to determine an optimal outcome is to evaluate all possible outcomes. In Appendix B, we present a dynamic algorithm (Algorithm 1), which uses Observation 3 to determine an optimal outcome $\mathcal{W}_{\mathrm{OPT}}$ in $O(3^n)$ steps. This algorithm is presented only for completeness; it is only useful when the number of assets is sufficiently small that 3^n operations can be carried out in a reasonable amount of time. If bids on all combinations were permitted, and the number of combinations for which bids were submitted grew exponentially with n, the algorithm would be polynomial in the size of the input. Of course, if the size of the input grew exponentially in n, then even the problem of recording all the input data would not be computationally manageable.

As we just mentioned, there is no way of being sure that the simultaneous auction procedure is manageable for large n if there are no restrictions on the set $\mathcal{P}$, not only because the problem is NP-complete, but also because the size of the input itself might become unmanageable.

As will be demonstrated in the later sections, the manageability of a simultaneous auction depends upon the structure of $\mathcal{P}$ rather than the size of $\mathcal{P}$ (the number of permitted combinations). Therefore, *merely* limiting the number of bids each bidder is allowed to submit will not, in general, guarantee that the bid evaluation problem is computationally manageable.[14]

3. Nested Structures

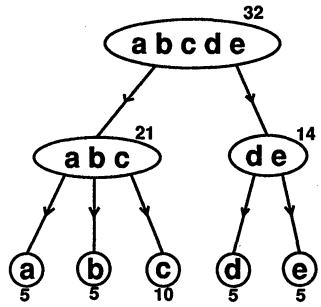
Perhaps the simplest example of this sort of structure is the following. Suppose all assets to be auctioned are either on the East Coast or the West Coast, and the underlying economics augurs against any synergies from combinations mixing East- and West-Coast assets, perhaps excepting the grand combination (all assets). Accordingly, suppose the auctioneer limits permitted combinations to the grand combination and combinations involving assets from only one coast. Then, the optimal outcome can be determined by finding the optimal outcomes for the East Coast and West Coast separately, and then comparing the revenue from these outcomes to the best bid for all assets.

Whenever the set of assets can be decomposed into k disjoint parts such that each permitted combination except the grand combination is contained within one of these parts, an optimal outcome of the auction can be found by simply comparing the union of optimal outcomes for the parts with the bid for all assets. Thus, the problem of finding an optimal outcome reduces to k smaller problems, which can reduce computational burdens by orders of magnitude.

Obviously, it might be possible to use this observation recursively to further simplify the problem of finding $\mathcal{W}_{\mathrm{OPT}}$. For certain structures of families of permitted combinations, this not only will simplify but indeed will solve the problem. A family of sets $\mathcal{P}$ forms a **tree structure** if for all $C, C' \in \mathcal{P}$, $(C \cap C')$ is one of $\emptyset, C, C'$. That is, every two sets in the family are disjoint or one is a subset of the other.

EXAMPLE. Let $A = \{a, b, c, d, e\}$ and suppose that permitted combinations are all singletons and $\{a, b, c\}$, $\{d, e\}$, and A. Figure 1 shows all permitted combinations represented as a tree. Suppose that numbers adjacent to $C \in \mathcal{P}$ in Figure 1 represent $b(C)$.

Figure 1 **Example of a Tree Structure**



On noting that the bids for singletons $\{a\}$, $\{b\}$, and $\{c\}$ sum to 20, less than the highest bid for $\{a, b, c\}$ (21), an algorithm need no longer consider $\{a\}$, $\{b\}$, and $\{c\}$. Similarly, $\{d\}$, and $\{e\}$ will never be winning combinations. Comparing to the grand combination, $\mathcal{W}_{\mathrm{OPT}} = \{\{a, b, c\}, \{d, e\}\}$ (because $21 + 14 > 32$).

The steps taken to solve this simple example generalize to handle arbitrarily complex tree structures $\mathcal{P}$. $\mathcal{W}_{\mathrm{OPT}}$ can be obtained by recursively simplifying the problem via splitting $\mathcal{P}$ into several smaller subfamilies of permitted combinations. Since $\mathcal{P}$ forms a tree structure, any $\mathcal{P}' \subseteq \mathcal{P}$ either is a singleton or permits further decomposition as described above. Although the intuition behind computational manageability when permitted combinations form a tree structure is recursive, it is not hard to develop an efficient nonrecursive algorithm. It is clear that a polynomial time algorithm must exist since the matrix M from (2) is totally unimodular, given that $\mathcal{P}$ forms a tree structure.[15] An example is Algorithm 2, presented in Appendix B, which computes $\mathcal{W}_{\mathrm{OPT}}$ in $O(n^2)$ time. This algorithm "builds up" $\mathcal{W}_{\mathrm{OPT}}$, starting with smaller combinations and then gradually considering larger ones.

The following theorem summarizes the observations above.

THEOREM 4. *Let $\mathcal{P}$ form a tree structure. Then an optimal outcome can be determined in $O(n^2)$ time.*

Auctioneers might find it advantageous to restrict permitted combinational bids to form a tree structure in situations where the underlying economics pointed to a particular superset of some set of assets as being the set at which the next synergistic valuation step

[14] Of course, to be computationally manageable, any bidding plan must limit the number of bids itself (i.e., the size of the input of the bid evaluation problem) to a manageable level.

[15] An equivalent problem is an exercise in Murty (1992).

occurred. If there were some sense in which synergies that were unattainable within a metropolitan area did not become attainable until the coverage extended to well-defined regions (e.g., the Federal Reserve Bank regions), then permitting bids on metropolitan areas, these regions, and the grand combination might allow bidders to submit bids on the efficient outcome.

One might hope that Algorithm 2 could be used to find $\mathcal{W}_{\mathrm{OPT}}$ when all permitted combinations can be represented by two or more tree structures. A heuristic approach would be to find an optimal outcome for each of the tree structures and then choose the best one among them. Unfortunately, in the case of the multiple tree structures, this heuristic approach could fail to determine an optimal outcome.

EXAMPLE. Suppose that, in addition to permitted combinations from the previous example, we also allow bids on combinations $\{a, b\}$ and $\{c, d, e\}$. Figure 2 shows two tree structures representing all permitted combinations.

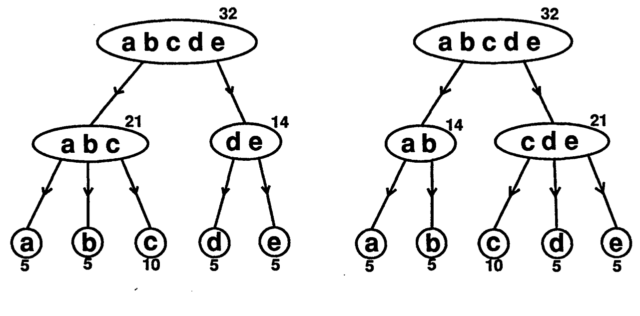
As indicated on the figure, we suppose that $b(\{c, d, e\}) = 21$ and $b(\{a, b\}) = 14$. An optimal outcome is $\mathcal{W}_{\mathrm{OPT}} = \{\{a, b\}, \{c\}, \{d, e\}\}$ with $\mathrm{rev}(\mathcal{W}_{\mathrm{OPT}}) = 38$. However, the revenue maximizing outcomes on each of the tree structures would lead to total revenue of 35.

Particular situations representable as multiple-tree structures can remain computationally manageable; one example is presented in Section 5. In general, however, computational manageability for multiple-tree structures cannot be guaranteed.

4. Cardinality-Based Structures

In a variety of situations, the combinations that are critical to attaining the key synergies may not allow nesting

Figure 2 Example of a Multiple Tree Structure



but may be structured by cardinality—that is, by the number of assets in the permitted combinations. An auction designer who can predict how large are the combinations on which bidders might wish to submit combinational bids may be able to take advantage of the tools presented in this section.

EXAMPLE. Suppose a building of differentiated apartments is converted into condominiums, which are then auctioned. It is unlikely that important synergies result from purchasing two or three units, but purchasing a large enough block to gain voting control over changes in the condominium charter may be worth more than the sum of the values of the units in the block.

Permitting large combinations and singletons yields a computationally manageable auction. For example, if for every nonsingleton combination $C \in \mathcal{P}$, $|C| > n/2$, then there can be at most one such combination in any optimal outcome. (This is because every two sets in $\mathcal{W}$ must be disjoint.) More generally, if there exists $k > 0$ such that every $C \in \mathcal{P}$ is either a singleton or $|C| > n/k$, then there can be at most $k - 1$ nonsingletons in any outcome $\mathcal{W}$. This idea can be generalized for any measure of sets in A, not just the number of elements of the set. For example, the FCC auctions could have allowed bids on any combination of licenses covering more than one-third of the U.S. population. In such cases, $\mathcal{W}_{\mathrm{OPT}}$ can be determined by simply evaluating $\mathrm{rev}(\mathcal{W})$ for all outcomes $\mathcal{W}$. Since there can be at most $k - 1$ nonsingletons in any outcome, the number of outcomes is not large.

THEOREM 5. *Let μ be a given finite measure on 2^A.*[16] *Let $\mathcal{P} = \{C \subseteq A : |C| = 1$ or $\mu(C) > \mu(A)/k\}$. Let $S := \{C \in \mathcal{P} : \mu(C) > \mu(A)/k$ and $b(C) > 0\}$. Then $\mathcal{W}_{\mathrm{OPT}}$ can be determined in $O(n|S|^{k-1})$ time.*

For example, if $k = 2$ in the Theorem, then we have at most $|S| + 1$ candidates for $\mathcal{W}_{\mathrm{OPT}}$: every $C \in S$ and $C = \emptyset$ defines an outcome, $\mathcal{W}_C$, consisting of C and all singletons disjoint from C ($\mathcal{W}_C := \{C\} \cup \{\{x\} : x \notin C\}$).

A simple special case that is covered by Theorem 5 is when the finite measure puts more weight on one asset

[16] We assume that there are no computational difficulties involved with evaluation of $\mu(C)$. If the complexity of evaluating μ is known, our results have a straightforward generalization.

than on all of the others combined. In this case, any combination involving that asset can be allowed. For example, an auction of a mainframe computer and various peripheral devices could allow bids on any combination involving the mainframe.

Now we turn our attention to the obverse situations, in which combinations capturing synergies are small sets. A trivial combinational auction is an auction where $\mathcal{P} = \{\{x\} : x \in A\}$. In such an auction, bids can be submitted for single assets only, and an optimal outcome is $\mathcal{W}_{\mathrm{OPT}} = \mathcal{P}$ where the winning bids are the highest submitted bids for particular assets. But there are also nontrivial possibilities.

EXAMPLE. Consider auctions of takeoff and landing rights. One of the key concerns in designing such auctions is the interrelatedness of the rights. Such rights often make economic sense only in pairs. If rights at multiple airports are being sold simultaneously, then a flight between two of the airports involved would require a takeoff right at one and a landing right at the other. For example, a flight from La Guardia to O'Hare would need a takeoff right at La Guardia and a landing right a given time later at O'Hare.

Suppose combinational bids are not permitted on any sets of size 3 or larger, but are permitted on all doubletons. Then, it is possible to determine the winning bids in polynomial time. This is because the problem of finding an optimal outcome reduces to finding a maximal weighted matching in a graph (and we show the reduction in the proof of Theorem 6). But the latter problem is well studied and there exist polynomial-time algorithms for solving it (the first one was given by Edmonds (1965); more about matchings can be found in Pulleyblank (1995)).

Algorithms for finding maximum-weight matching are not particularly transparent and are considered to be among the most complicated polynomial-time algorithms. Therefore, it should not be surprising that allowing bids on any combination of size three risks serious computational problems.

THEOREM 6. *(a) If $\mathcal{P} \subseteq \{C \subset A : |C| \le 2\}$, then $\mathcal{W}_{\mathrm{OPT}}$ can be determined in $O(n^3)$ time. (b) If $\mathcal{P} = \{C \subset A : |C| \le 3\}$, then finding $\mathcal{W}_{\mathrm{OPT}}$ is NP-complete.*

Without significantly increasing computational difficulty, combinations of size two and large combinations can be allowed simultaneously. The following result compiles results from this section:

THEOREM 7. *Let μ be a given finite measure on 2^A. Let $\mathcal{P} = \{C \subseteq A : |C| \le 2$ or $\mu(C) > \mu(A)/k\}$. Let $S := \{C \in \mathcal{P} : \mu(C) > \mu(A)/k$ and $b(C) > 0\}$. Then $\mathcal{W}_{\mathrm{OPT}}$ can be determined in $O(n^3|S|^{k-1})$ time.*

5. Geometry-Based Structures

Often there exists some physical connection between the assets. For example, a group of neighboring assets may be more valuable as a combination than some otherwise equivalent random group of assets.

EXAMPLE. Some years ago California sold oil leases in Southern California that formed a single swath of offshore waters. In such a situation, most of the interesting combinations are groups of neighboring assets.

Suppose that there exists a total ordering among the assets. In other words, for every two assets $i, j \in A$: $i < j$ or $j < i$. For example, licenses for radio frequencies in the cities on the East Coast could be ordered according to a North-South geographical ordering of the cities. Suppose that we allow combined bids for any set of consecutive assets $[i, j] := \{x \in A : i \le x \le j\}$ ($C_{i,j}$ is an **interval**). Note that if the assets can be put in the total order, then we can label them in a way that $A = \{1, 2, \cdots n\}$ with the ordinary relation $<$. Hence, $A = [1, n]$.

For example, suppose that the licenses for Boston, New York, Philadelphia, Baltimore, and Washington, D.C., are numbered 1, 2, 3, 4, and 5, respectively. Then a combined bid for New York, Philadelphia, and Baltimore licenses is $[2, 4] = \{2, 3, 4\}$. Note that, if only bids for sets of consecutive licenses are permitted, then, for example, it is not possible submit a combinational bid for only the New York and Washington, D.C. licenses ($C = \{2, 5\}$) because any permitted combination containing both of these licenses must contain all licenses between them—that is, the Philadelphia and Baltimore licenses.

It should not be surprising that there exist efficient algorithms for determining $\mathcal{W}_{\mathrm{OPT}}$ when only bids for combinations of consecutive assets are permitted. It is easy to see that in this case the matrix M in (2) is totally unimodular since it has so-called "consecutive ones" property (see Murty 1992). Network flow algorithms, as well as dynamic programming approach can be used to solve (2) in this case (see Ahuja et al. 1993). Here we present a simple dynamic programming algorithm (Algorithm 3 in Appendix B) that produces $\mathcal{W}_{\mathrm{OPT}}$ when $A = [1, n]$ and the permitted combinations are sets of consecutive assets (i.e., $\mathcal{P} := \{[i, j] : 1 \le i \le j \le n\}$). The main idea of the algorithm is that the value of the first k assets can be evaluated by evaluating the value of first l assets and adding $b([l + 1, k])$ $(l = 0, \ldots, k - 1)$.

Instead of considering intervals (sets of consecutive assets) on the line (totally ordered set) we can consider intervals on the circle[17] (for example, offshore tracts surrounding an island might define such a structure) and determine $\mathcal{W}_{\mathrm{OPT}}$ by repeated use of any algorithm for determining $\mathcal{W}_{\mathrm{OPT}}$ for auctions of consecutive assets (for example, Algorithm 3). The algorithm gets applied n times, taking each of n elements to be the first one in the linear order defined by the cyclic order. For example, the k-th run of the algorithm considers the order: k, $k + 1, \ldots, n, 1, 2, \ldots, k - 1$. At the end, $\mathcal{W}_{\mathrm{OPT}}$ is chosen by comparing these n outcomes.

Analysis of Algorithm 3 yields the following theorem:

THEOREM 8. *Let $A = [1, n]$. (a) If $\mathcal{P} \subseteq \{[i, j] : 1 \le i \le j \le n\}$, then $\mathcal{W}_{\mathrm{OPT}}$ can be determined in $O(n^2)$ time.*

(b) If $\mathcal{P} = \{[i, j] : 1 \le i, j \le n\}$ where $[i, j] := \{i, i + 1, \ldots, j\}$ if $i \le j$ and $[i, j] := \{i, i + 1, \ldots, n, 1, \cdots j\}$ if $i > j$, then $\mathcal{W}_{\mathrm{OPT}}$ can be determined in $O(n^3)$ time.

In many cases, assets can be identified as elements of a direct product of two linear orders. Geographic location of an asset (position on a map) is an example. Clearly, X can be represented as $A = [1, m] \times [1, n]$. The two dimensional analogues of the intervals are rectangles $[a, b] \times [c, d] := \{(x, y) \in A : a \le x \le b \text{ and } c \le y \le d\}$. Unfortunately, there is no hope for finding a computationally manageable algorithm for an auction of A if combinational bids are permitted on all rectangles. In fact, as demonstrated in the proof of the following theorem, even when all two-by-two rectangles are permitted combinations, the problem of finding an optimal outcome is NP-complete.

[17] The matrix M in (2) has the so-called "circular ones" property. Bartholdi, Orlin, and Ratliff (1980) presented efficient algorithms for solving (2) when M has the circular ones property.

THEOREM 9. *Let $A = [1, m] \times [1, n]$ and let $\mathcal{P} = \{[a, b] \times [c, d] : 1 \le a \le b \le m \text{ and } 1 \le c \le d \le n\}$. Then finding a $\mathcal{W}_{\mathrm{OPT}}$ is an NP-complete problem.*

If we allow only rectangles of a specific type, there may be a computationally easy algorithm for finding $\mathcal{W}_{\mathrm{OPT}}$. For example, rows and columns of an $m \times n$ rectangular grid can be viewed as rectangles.

EXAMPLE. This might be an appropriate representation of a situation in which a set of collectible assets is to be sold where assets have two different properties of interest to different collectors, as with the years and denominations of coins. Synergies may primarily be gained when a collection comprises a complete set for a given year, or else a collection of, say, nickels for a lengthy set of consecutive years.

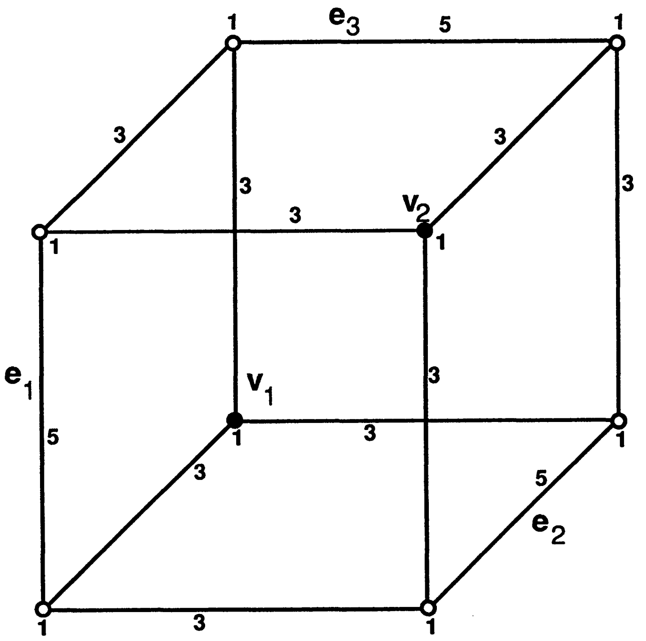
Formally, rows R_a and columns C_b of an $m \times n$ rectangular grid are rectangles of the form $R_a := [a, a] \times [1, n] = \{(a, y) \in A : y \in [1, n]\}$ $(a = 1, \ldots, m)$ and $C_b := [1, m] \times [b, b] = \{(x, b) \in A : x \in [1, m]\}$ $(b = 1, \ldots, n)$. Then, if bids on singletons, rows, and columns are permitted, $\mathcal{W}_{\mathrm{OPT}}$ can be determined in a very transparent way. Note that in any outcome $\mathcal{W}$, there cannot be a row and a column at the same time because $R_a \cap C_b = \{(a, b)\} \neq \emptyset$. Therefore, every outcome contains rows and singletons only or columns and singletons only. Since rows (columns) are pairwise disjoint, an optimal outcome containing rows (columns) is easily determined by checking whether replacing a row (column) by singletons increases $\mathrm{rev}(\mathcal{W})$. At the end, only the best of the outcomes containing rows and the best of the outcomes containing columns need to be compared.[18]

A straightforward k-dimensional generalization is the situation where the assets can be described by k different properties.

EXAMPLE. Let A be vertices of a cube and let $\mathcal{P}$ consist of vertices (singletons) and edges (some doubletons) of the cube. Let $b(C) = 1$ for every vertex, let $b(e_1) = b(e_2) = b(e_3) = 5$, and let $b(e) = 3$ for every other edge (edges e_1, e_2 and e_3 are denoted in Figure 3). Note that

[18] This is an example where repeated use of Algorithm 2 (for tree structures) produces an optimal outcome for a multiple tree structure. Note that all singletons, all rows (resp. columns), and the set of all assets A form a tree structure. Hence, all $C \in \mathcal{P}$ form two tree structures.

Figure 3 **An Example of Auctioning the Vertices and the Edges of a Cube, Showing that an Optional Outcome Can Contain Edges of all Three Orientations**



there are three types of edges in the drawing: horizontal, vertical, and "diagonal." If we were to follow the approach for the 2-dimensional case, we would find and evaluate optimal outcomes containing just one type of edge. In all three cases, we get $3 + 3 + 3 + 5 = 14$, and no singletons are involved. However, the optimal outcome $\{e_1, e_2, e_3, v_1, v_2\}$ has value $5 + 5 + 5 + 1 + 1 = 17$ and includes edges of different types. The reason why this approach doesn't work in this case is that there are edges of different types that are disjoint.

When can the problem of finding $\mathcal{W}_{\text{OPT}}$ be decomposed into several simpler problems (as in the example where bids on rows and columns are allowed)? One example is when permitted nontrivial (i.e., nonsingleton) combinations can be partitioned into k families, so that any two combinations drawn from different families intersect. If so, an optimal outcome contains combinations from only one family. More formally,

PROPOSITION 10. *Let* $\mathcal{P} = \mathcal{S} \cup \mathcal{P}_1 \cup \mathcal{P}_2 \cup \cdots \cup \mathcal{P}_k$ *where* $\mathcal{S} = \{\{x\} : x \in A\}$ *(singletons) and for any* $i \neq j$, $A \in \mathcal{P}_i$, $B \in \mathcal{P}_j \geq A \cap B = \emptyset$. *Then* $\mathcal{W}_{\text{OPT}} \subseteq \mathcal{S} \cup \mathcal{P}_i$ *for some* i.

PROOF. If $\mathcal{W}$ is an outcome, then it cannot contain nontrivial combinations from more than one family $\mathcal{P}_i$ because every two sets from $\mathcal{W}$ must be disjoint. Therefore, $\mathcal{W} \subseteq \mathcal{S} \cup \mathcal{P}_i$ for some i. □

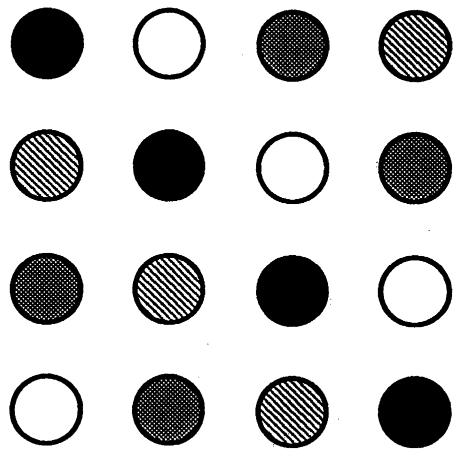
Also note that if $\mathcal{P}$ satisfies conditions of Proposition 10, then for any $C_l \in \mathcal{P}_i$, $\mathcal{P}^* = \{C \in \mathcal{P} : C \subseteq A \backslash C_l\} \subseteq \mathcal{S} \cup \mathcal{P}_i$ (i.e., $\mathcal{P}^*$ is contained in the remainder of C_l's family). Hence, determining how much a losing bid must be raised is as hard as finding an optimal outcome for $A \backslash C_l$ with permitted combinations from $\mathcal{P}^* \subset \mathcal{S} \cup \mathcal{P}_i$ only.

The 2-dimensional example satisfies the conditions of Proposition 10 with $\mathcal{P}_1$ being the set of all rows and $\mathcal{P}_2$ being the set of all columns. As long as finding an optimal outcome where permitted combinations are unions of rows (resp. columns) is computationally manageable, the problem of finding $\mathcal{W}_{\text{OPT}}$ is computationally manageable also. For example, we may allow bids on any combination of consecutive rows or columns (i.e., rectangles of type $[a, b] \times [1, n]$ and of type $[1, m] \times [c, d]$ are permitted combinations) and use Algorithm 3 twice to determine $\mathcal{W}_{\text{OPT}}$.

In fact, in addition to rows and columns, one can also allow bidding on any other family of combinations whose elements intersect every row and every column. For example, in the $n \times n$ case we can always allow bids on n "diagonal" combinations also. Figure 4 shows four diagonals in the 4×4 case.

This idea can be generalized. If A can be represented as a k-dimensional vector space over a finite field, then

Figure 4 **Four Diagonals (Black, White, Gray, and Shaded) Intersect Every Row and Every Column**



we can allow bids on any combination representing a hyperplane (i.e., a $(k-1)$-dimensional plane). Note that we can partition all hyperplanes of a k-dimensional vector space into families of parallel hyperplanes. Then, any pair of hyperplanes drawn from different families intersects. (Note that the cube in Figure 3 is just a 3-dimensional vector space over the field of two elements, and the edges are 1-dimensional planes. Since 1-dimensional planes are not hyperplanes in a 3-dimensional vector space, we could find disjoint edges from different families.) For example, the previous coin example might fit this generalization if potential synergies in values resulted from three aspects of coins: year, denomination, and origin (Philadelphia, Denver, or San Francisco Mint).

6. Concluding Remarks

Some auction markets sell a considerable number of assets simultaneously. Often, the value of an asset to a bidder depends on which other assets he or she also wins. In such situations, allowing bidding on combinations of assets may offer a way of increasing the efficiency of the allocation of assets, but it can raise computational problems.[19]

Those who determine the rules for simultaneous auctions must determine for which combinations to allow bids. In sales with many assets, "all combinations" may not be a workable answer, and "no combinations" may not be a desirable one. In this paper, we have considered restricted sets of combinations for which combinational bidding presents a provably manageable computational burden. We hope that this will allow auction designers to design workable auctions that are more efficient.

[19] This paper has explored extensively the effects on our ability to guarantee computational tractability in determining the revenue-maximizing set of bids by restricting the kinds of combinations upon which bidders are allowed to bid. We have also commented on the ineffectiveness, in general, of merely restricting the number of combinational bids a bidder may submit. However, we have not considered the computational implications of combined policies in which both the number of combinational bids and the types of combinational bids are jointly restricted. It is possible that when the number of bidders is small, there are potentially useful forms of restriction that allow each bidder one or, possibly, a few bids on more general combinations than can be handled with unrestricted numbers of bids, and that such restrictions lead to guaranteed computational manageability. We leave the exploration of this possibility to further research.

Deciding for which combinations to allow bids puts a responsibility on auction designers—a responsibility that some of them may find politically risky to fulfill. We note that computational impossibility does not protect against such responsibility, and that "no combinations" is only one choice among what we have now established are many feasible possibilities. Responsible auction designers will try to determine the kinds of combinations of greatest economic significance and will attempt to allow bids on them in the auction—at least when there is reason to believe that economies of scale exist. The politically astute among them may well try to involve the potential bidders in that determination process. Perhaps, the Department of the Interior's use of a nomination process in deciding which offshore tracts to offer for sale at a given time could serve as a model.[20]

[20] The authors thank Richard Engelbrecht-Wiggans, Eric Greenleaf, Karla Hoffman, Edward Kaplan, and two referees for helpful comments, and gratefully acknowledge the National Science Foundation for its support under grant number SBR 93-09333 to Rutgers University.

Appendix A

Additional Background—Computational Complexity

The difficulty of computational problems does depend on particular details of the problem and the computing capabilities that are to be applied to solve it. It is useful, however, for scholars to have ways of discussing computational complexity for broad classes of problems, and for these discussions to be meaningful independent of problem details or computer size and speed. The salient and unavoidable fact is that solutions become more difficult to compute as the problems grow in the size of the input (principally the numbers of control and state variables, and constraints).

Mathematicians have focused on how time needed to obtain a solution (measured in the number of computations) grows as a function of the size of the input. If there exists an algorithm for which the number of steps needed to compute the solution to any problem in that problem class (i.e., for any data instance of that problem) can be expressed as a polynomial function of input size, then the problem is considered to be relatively tractable computationally. Such problems fall within the polynomial class (P), and are then further classified by the degree of the bounding polynomial function. If the bound on the number of steps needed to compute the solution is a superpolynomial function of input size (i.e., cannot be bounded by any polynomial function; any exponential function is an example) then the algorithm is considered to be inefficient since solving every data instance of the problem using such an algorithm cannot be guaranteed even for problems with relatively small input sizes.

While the size of problems that can be solved grows with advances in both algorithms and affordable computing power, this distinction is still quite useful. An interesting class of problems, for which no polynomial time algorithm for finding a solution to every data instance is known, are NP-complete problems (the general integer programming problem, for example, is NP-complete). Existence of a polynomial time algorithm for solving any NP-complete problem would imply that there are also polynomial time algorithms for solving every other NP-complete problem. It is widely believed that no polynomial time algorithm exists for any such problem, although this is still an open question in complexity theory. Thus, if the problem is shown to be NP-complete, it is currently (and probably always will be) impossible to guarantee that it can be solved in polynomial time; it is in this sense that it is computationally intractable. We note, however, that it is quite possible that there exist a polynomial time algorithm for solving a particular instance (or a set of instances) of an NP-complete problem. In fact, many problems arising in industry that are NP-complete are routinely solved to proven optimality because there exist efficient algorithms for solving the problem instances arising in practice.

In our analysis of the computational complexity of finding an optimal outcome for a variety of combinational auctions, we heavily use the standard asymptotic notation for growth of functions (for example, see Cormen et al. 1990). Let f and g be two functions from $\mathbf{N}$ to $\mathbf{N}$ (where $\mathbf{N}$ denotes the set of natural numbers, i.e., $\mathbf{N} = \{1, 2, 3, \cdots\}$).

If there exist $c > 0$ and n_0 such that $f(n) \geq cg(n)$ for all $n \geq n_0$, then we say that g is an asymptotic lower bound (within a constant) for f and use the notation $f = \Omega(g)$.

Similarly, if there exist $c > 0$ and n_0 such that $f(n) \leq cg(n)$ for all $n \geq n_0$, then we say that g is an asymptotic upper bound (within a constant) for f and use the notation $f = O(g)$. If, in addition, for any $c > 0$ we can find n_0, then we say that $f = o(g)$ (in other words, $f = o(g)$ if and only if $\lim_{n\to\infty} f(n)/g(n) = 0$).

In this notation, the problem is in the class P if there is an algorithm that finds the solution in $O(n^k)$ steps ($O(n^k)$ time) for some fixed k. An example of a superpolynomial algorithm would be an algorithm that needs $\Omega(2^n)$ time to find a solution. More on computational complexity can be found in Papadimitriou (1994), and in Garey and Johnson (1979).

Appendix B

Algorithms and Proofs of Theorems

Proof of Proposition 1. We first consider $C_L \notin \mathcal{W}_{\text{OPT}}$. $C_L \notin \mathcal{W}_{\text{OPT}}$ becomes a winning combination if and only if one of the outcomes that contains C_L becomes a winning outcome. Hence,

$$b_{\text{win}}(C_L) - b(C_L) = \text{rev}(\mathcal{W}_{\text{OPT}}) - \max\{\text{rev}(\mathcal{W}) : \mathcal{W} \in \Omega_{\mathcal{P}}, C_L \in \mathcal{W}\}. \quad (3)$$

This is because increasing $b(C_L)$ by some amount increases $\text{rev}(\mathcal{W})$ by the same amount if and only if $C_L \in \mathcal{W}$. Also note that for any $\mathcal{W} \in \Omega_{\mathcal{P}} : C_L \in \mathcal{W}$, if and only if $\mathcal{W} \setminus \{C_L\} \in \Omega_{\mathcal{P}^*}$ (i.e., $\mathcal{W} \setminus \{C_L\}$ is an outcome for auction of A^* where permitted combinations are $\mathcal{P}^*$). Furthermore, for $\mathcal{W} \ni C_L$, $\text{rev}(\mathcal{W}) = b(C_L) + \text{rev}(\mathcal{W} \setminus \{C_L\})$. Therefore, subtracting $b(C_L)$ from both sides of (3) yields

$$b_{\text{win}}(C_L) = \text{rev}(\mathcal{W}_{\text{OPT}}) - \max\{\text{rev}(\mathcal{W}) : \mathcal{W} \in \Omega_{\mathcal{P}^*}\} = \text{rev}(\mathcal{W}_{\text{OPT}}) - \text{rev}(\mathcal{W}^*_{\text{OPT}}).$$

Next consider $C_W \in \mathcal{W}_{\text{OPT}}$. Clearly, for any outcome $\mathcal{W} \ni C_W$, decreasing $b(C_W)$ will decrease $\text{rev}(\mathcal{W})$ by the same amount. Hence, C_W will remain a winning combination as long as one of the $\mathcal{W} \ni C_W$ remains optimal. In other words,

$$b(C_W) - b_{\text{win}}(C_W) = \text{rev}(\mathcal{W}_{\text{OPT}}) - \max\{\text{rev}(\mathcal{W}) : \mathcal{W} \in \Omega_{\mathcal{P}} : C_W \notin \mathcal{W}\}$$

from which

$$b_{\text{win}}(C_W) = b(C_W) - \text{rev}(\mathcal{W}_{\text{OPT}}) + \text{rev}(\mathcal{W}^{**}_{\text{OPT}})$$

follows immediately. □

Proof of Corollary 2. It suffices to show that any algorithm with input $b(C)$, $C \in \mathcal{P}$, and output $\text{rev}(\mathcal{W}_{\text{OPT}})$, can be used to solve minimal winning bid problem for a given $C \in \mathcal{P}$. By Proposition 1, we only need to show that the same algorithm can be used to find $\text{rev}(\mathcal{W}^*_{\text{OPT}})$ and $\text{rev}(\mathcal{W}^{**}_{\text{OPT}})$. Note that $\text{rev}(\mathcal{W}^*_{\text{OPT}}) = \text{rev}(\mathcal{W}_{\text{OPT}})$ whenever $b(C) = 0$ for every $C \in \mathcal{P}$ such that $C \cap C_L \neq \emptyset$. Hence, by setting $b(C) = 0$ for every $C \in \mathcal{P} \setminus \mathcal{P}^*$, the algorithm for finding $\text{rev}(\mathcal{W}_{\text{OPT}})$ can be used to find $\text{rev}(\mathcal{W}^*_{\text{OPT}})$. Similarly, by setting $b(C_W) = 0$, the algorithm will find $\text{rev}(\mathcal{W}^{**}_{\text{OPT}})$ (since $\text{rev}(\mathcal{W}^{**}_{\text{OPT}}) = \text{rev}(\mathcal{W}_{\text{OPT}})$ whenever $b(C_W) = 0$). □

Proof of Observation 3. Define $\mathcal{W}' := (\mathcal{W}_{\text{OPT}} \setminus \{C^*\}) \cup \{C_1, C_2, \cdots C_k\}$.

$$\sum_{C \in \mathcal{W}'} b(C) = \sum_{C \in \mathcal{W}_{\text{OPT}}} b(C) - b(C^*) + b(C_1) + b(C_2) + \cdots + b(C_k).$$

Since $\Sigma_{C \in \mathcal{W}_{\text{OPT}}} b(C) = \text{rev}(\mathcal{W}_{\text{OPT}}) \geq \text{rev}(\mathcal{W}') = \Sigma_{C \in \mathcal{W}'} b(C)$ (because $\mathcal{W}_{\text{OPT}}$ is an optimal outcome), $b(C_1) + b(C_2) + \cdots + b(C_k) \leq b(C^*)$ follows. □

Algorithm 1: Dynamic Algorithm for the General Problem[21]
INPUT: $b(C)$ for all $C \subseteq A$ (if no bids are submitted or $C \notin \mathcal{P}$ then $b(C) = 0$).

1. For all $x \in A$, set $f(\{x\}) := b(\{x\})$, $\mathcal{C}(\{x\}) := \{x\}$.
2. For $i = 2$ to n, do:
   For all $C \subseteq A$ such that $|C| = i$, do:
   (a) $f(C) := \max\{f(C \setminus C') + f(C') : C' \subseteq C$ and $1 \leq |C'| \leq \frac{|C|}{2}\}$.
   (b) If $f(C) \geq b(C)$, then set $\mathcal{C}(C) := C^*$ where C^* maximizes right hand side of (a).
   (c) If $f(C) < b(C)$, then set $f(C) := b(C)$ and $\mathcal{C}(C) := C$.
3. Set $\mathcal{W}_{\text{OPT}} := \{A\}$.
4. For every $C \in \mathcal{W}_{\text{OPT}}$, do:
   If $\mathcal{C}(C) \neq C$, then
   (a) Set $\mathcal{W}_{\text{OPT}} := (\mathcal{W}_{\text{OPT}} \setminus \{C\}) \cup \{\mathcal{C}(C), C \setminus \mathcal{C}(C)\}$.

[21] It is possible that there is more than one optimal outcome. Each algorithm we present incorporates tie-breaking rules in its execution, outputting a single optimal outcome. Each can be adjusted (in a straightforward way, essentially not compromising its complexity) to output all optimal outcomes.

(b) Go to 4 and start it with the new $\mathcal{W}_{\text{OPT}}$.

There is a constant K such that this algorithm requires fewer steps than

$$n^2 + \sum_{k=1}^{n} K2^{k-1}\binom{n}{k} = O(3^n).$$

After the first two steps, $f(A)$ will be exactly $\text{rev}(\mathcal{W}_{\text{OPT}})$ (i.e., $\Sigma_{C\in \mathcal{W}_{\text{OPT}}} b(C)$). $f(C)$ is calculated using the fact that all the assets from C will be sold to a single bidder (who submitted the highest bid for C) if and only if any dividing of C into smaller combinations will not lead to the higher value collected by the bid-taker. The variable $\mathcal{C}(C)$ keeps track of the structure of $\mathcal{W}_{\text{OPT}}$, which is being determined recursively in Step 4.

Note that the algorithm doesn't need to evaluate the function f for every subset of A. It suffices to check only the sets $S \subseteq A$ in the smallest algebra $\mathcal{A} \subseteq 2^A$ of sets containing $\mathcal{P}$ (or only those $C \in \mathcal{P}$ for which a bid is submitted). (An algebra $\mathcal{A}$ of sets is a family of sets $A \in \mathcal{A} \subseteq 2^A$ closed under union and taking complements, i.e., $A \in \mathcal{A} \Rightarrow A^c \in \mathcal{A}$ and $A, A' \in \mathcal{A} \Rightarrow A \cup A' \in \mathcal{A}$.)

The value of $f(A\backslash C)$ is exactly $\text{rev}(\mathcal{W}^*_{\text{OPT}})$ for every $C \subseteq A$. A $\mathcal{W}^*_{\text{OPT}}$ can be determined by setting $\mathcal{W}_{\text{OPT}} := A\backslash C$ and then executing Step 4. In many cases, this information won't be a byproduct of an algorithm to find an optimal solution. However, the existence of an easy way to determine an optimal outcome automatically gives an easy way to determine an optimal outcome for an auction of the smaller set of assets $(A\backslash C)$, provided permitted combinations on the smaller set of assets are of the same type as permitted combinations on the original set of assets.

Algorithm 2: Finding $\mathcal{W}_{\text{OPT}}$ for Tree Structures

Before presenting the algorithm, note that the sets in a tree structure $\mathcal{P}$ define a directed tree $T(\mathcal{P})$ in a natural way: $\mathcal{P}$ is the set of vertices of $T(\mathcal{P})$, and (C, C') is an arc in $T(\mathcal{P})$ if and only if $C \supset C'$ and there is no $C'' \in \mathcal{P}$ such that $C \supset C'' \supset C'$. In other words, (C, C') is an arc in the tree $T(\mathcal{P})$ if and only if C covers C' in $\mathcal{P}$. For any arc (C, C'), we say that C is the **tail** and C' is the **head** of the arc. Precise definitions of the standard graph theoretic terms that we are using here can be found in any textbook on graph theory (for example, see Bondy and Murty 1976). Note that $C_1 \supset C_2$ if and only if there exist a directed path $P \subseteq T(\mathcal{P})$ from C_1 to C_2. The length of the directed path P from C_1 to C_2 (i.e., the number of arcs in P) is denoted by $d(C_1, C_2)$. If there is no directed path from C_1 to C_2, nor from C_2 to C_1, then $C_1 \cap C_2 = \emptyset$, since $\mathcal{P}$ forms a tree structure.

Note that we can always add A to $\mathcal{P}$ without violating the tree structure property. In other words, if $\mathcal{P}$ forms a tree structure, $\mathcal{P} \cup \{A\} \cup \{\{x\} : x \in A\}$ also forms a tree structure. Also note that every C has at most one ingoing arc (otherwise there would be two sets covering C and then their intersection would be at least $C \neq \emptyset$). If $A \in \mathcal{P}$, then every $C \neq A$ will have exactly one ingoing arc.

We call $C \in \mathcal{P}$ a **leaf**, if C has no outgoing arcs.

INPUT: $T(\mathcal{P} \cup A)$ and $b(C)$ for all $C \in \mathcal{P}$.

1. Set $\mathcal{W}_{\text{OPT}}(C) = \{C\}$ for every leaf C.
2. For every $C \in \mathcal{P}$, calculate $\bar{d}(C) := d(A, C)$.
3. Find $C_{\max}$ such that $\bar{d}(C_{\max}) \geq \bar{d}(C')$ for all $C' \in \mathcal{P}$.
4. (a) Let C_0 be tail of the unique ingoing arc of $C_{\max}$ (i.e., C_0 covers $C_{\max}$).
   (b) Let $\mathcal{S} := \{C \in \mathcal{P} : (C_0, C) \text{ is an arc in } (T(\mathcal{P}))\}$. (Note that by the choice of $C_{\max}$ every $C \in \mathcal{S}$ is a leaf)
   (c) Calculate $\text{rev}(\mathcal{S}) := \Sigma_{C\in\mathcal{S}} b(C)$.
   (d) If $b(C_0) > \text{rev}(\mathcal{S})$, then set $\mathcal{W}_{\text{OPT}}(C_0) := \{C_0\}$.
   (e) If $b(C_0) \leq \text{rev}(\mathcal{S})$, then set $b(C_0) := \text{rev}(\mathcal{S})$ and $\mathcal{W}_{\text{OPT}}(C_0) := \cup_{C\in\mathcal{S}} \mathcal{W}_{\text{OPT}}(C)$.
5. If $C_0 = A$, then STOP ($\mathcal{W}_{\text{OPT}}(A)$ is an optimal outcome).
6. Set $\mathcal{P} := \mathcal{P}\backslash\mathcal{S}$ and go to Step 3. (C_0 becomes a leaf in $T(\mathcal{P})$)

Steps 1, 2, and 3 require $O(|\mathcal{P}|)$ time. Steps 4, 5, and 6 require $O(n)$ time since any C can have at most $|C| \leq n$ outgoing arcs. Steps 3 to 6 can be repeated at most $|\mathcal{P}|$ times since every time $\mathcal{P}$ is updated in Step 6, at least $C_{\max}$ is deleted from $\mathcal{P}$. Hence, the algorithm requires $O(|\mathcal{P}|(|\mathcal{P}| + n))$ time.

Proof of Theorem 4. Let us first show by induction on $|\mathcal{P} \cup \{A\}|$ that $\mathcal{W}_{\text{OPT}}(A)$ from Algorithm 2 is an optimal outcome. Obviously, the algorithm produces an optimal outcome for $\mathcal{P} \cup \{A\} = \{A\}$.

Let $\mathcal{S} := \{C \in \mathcal{P} : (A, C) \in A(T(\mathcal{P}))\}$. Note that $\mathcal{P}_C := \{C' \in \mathcal{P} : C' \subseteq C\}$ forms a tree structure and the algorithm produces an optimal outcome for every $\mathcal{P}_C$ where $C \in \mathcal{S}$. Also note that $\mathcal{W}_{\text{OPT}}(C)$ is this optimal outcome and that $\text{rev}(\mathcal{W}_{\text{OPT}}(C)) = b(C)$ in the original algorithm. Since every $C' \in \mathcal{P}$ $(C' \neq A)$ intersects exactly one $C \in \mathcal{S}$, any optimal outcome for $\mathcal{P}$ is either $\mathcal{W}_{\text{OPT}} = \{A\}$ or a disjoint union of optimal outcomes in $\mathcal{P}_C$. Therefore, $\mathcal{W}_{\text{OPT}}(A)$ is an optimal outcome.

It remains to show that the algorithm requires at most $O(n^2)$ time. Since the algorithm requires $O(|\mathcal{P}|(|\mathcal{P}| + n))$ time, it suffices to show that for any tree structure $\mathcal{P}$, $|\mathcal{P}| \leq 2n - 1$. This can be easily proved by induction. Clearly, the statement is true for $n = 1$. Let $D \in \mathcal{P}$ be maximal in $\mathcal{P}\backslash\{A\}$ (that is, there is no combination in $\mathcal{P}\backslash\{A\}$ containing D). Then, from the definition of tree structures,

$$|\mathcal{P}\backslash\{A\}| = |\{C \in \mathcal{P} : C \subseteq D\}| + |\{C \in \mathcal{P} : C \subseteq A\backslash D\}.$$

So, by induction, we conclude that $|\mathcal{P}| \leq 1 + |\mathcal{P}\backslash\{A\}| \leq 1 + (2|D| - 1) + (2(n - |D|) - 1) = 2n - 1$. □

Proof of Theorem 5. By Observation 3, we can eliminate any $C \in S$ such that $b(C) \leq \Sigma_{x\in C} b(\{x\})$. This can be done in $O(|S|n)$ time.

Note that there can be at most k sets from S in any outcome $\mathcal{W}$. This is because every two sets in $\mathcal{W}$ must be disjoint and there can be at most $k - 1$ disjoint sets $C \in \mathcal{W}$, $\mu(C) > \mu(A)/k$. Therefore, there are all together

$$\sum_{i=0}^{k-1} \binom{|S|}{i} = O(|S|^{k-1})$$

candidates for $\mathcal{W}_{\text{OPT}}$. For each such $\mathcal{W}$, we need to calculate $\text{rev}(\mathcal{W})$ (this can be done in at most n steps) and then find $\mathcal{W}_{\text{OPT}}$ among them. All together, we need $O(n|S|^{k-1})$ time. □

Proof of Theorem 6. We first show (a). Let G' be a graph whose set of vertices is A and set of edges is the set of all combinations of size two, i.e., $\{x, y\}$ is an edge in the graph G' if and only if $\{x, y\} \in \mathcal{P}$. Let G be a graph obtained from G' by adding a vertex $\bar{x}$ and an edge $e_x := (x, \bar{x})$ with $b(e_x) := b(\{x\})$ for every $\{x\} \in \mathcal{P}$. Obviously, G

has at most $2n$ vertices. Note that the natural one-to-one correspondence between the edges of G and the combinations $C \in \mathcal{P}$ preserves disjointness, and therefore, $\mathcal{W}$ is an outcome if and only if corresponding edges are matching in G. But the maximum weight matching in G can be found in $O((2n)^3) = O(n^3)$ time by Edmonds' algorithm (see Lawler 1976).

In order to show (b), we will show that the problem of finding $\mathcal{W}_{\text{OPT}}$ is NP-complete in a particular special case of such an auction. Suppose that $b(C) = 0$ for all C such that $|C| \le 2$ and $b(C) = 0$ or 1 for all other $C \in \mathcal{P}$. Then $\mathcal{W}_{\text{OPT}}$ is an optimal outcome if and only if $\{C \in \mathcal{W}_{\text{OPT}} : b(C) = 1\}$ is an maximal 3-set packing on a hypergraph $\mathcal{P}$. The 3-set packing problem is known to be NP-complete (Garey and Johnson 1979). Hence, even in the very simple case when all $b(C)$ are either 0 or 1, this problem is NP-complete. □

PROOF OF THEOREM 7. As it was shown in Theorem 5, there are only $O(|S|^k)$ different outcomes $\mathcal{W}$ containing disjoint sets from S (Maybe some additional sets from S can be eliminated using Observation 3 and $O(|A \backslash C|^3)$ time is needed to check if C should be eliminated. All together, $O(|S|n^3)$ will suffice).

Let $\mathcal{W}_{\text{OPT}}$ be an optimal outcome for auction of $A^* = A \backslash (\cup_{C \in S \cap \mathcal{W}} C)$ where $\mathcal{P}^* = \{C \subseteq A^* : C \in \mathcal{P} \text{ and } |C| \le 2\}$. $\mathcal{W}^*_{\text{OPT}}$ can be determined in $O(|A^*|^3)$ time by Theorem 6. Then $\mathcal{W} \cup \mathcal{W}^*_{\text{OPT}}$ is a candidate for $\mathcal{W}_{\text{OPT}}$ with $\text{rev}(\mathcal{W} \cup \mathcal{W}^*_{\text{OPT}}) = \text{rev}(\mathcal{W}^*_{\text{OPT}}) + \Sigma_{C \in S \cap \mathcal{W}} b(C)$.

Therefore, we can find $\mathcal{W}_{\text{OPT}}$ in $O(n^3|S|^k)$ time. □

Algorithm 3: Finding $\mathcal{W}_{\text{OPT}}$ for Intervals

INPUT: $b([i, j])$ for all i, j.

1. Set $\mathcal{W}(1) := \{[1, 1]\}$ and $w(1) := b([1, 1])$. Set $r := 2$.
2. Set $w(r) := b([1, r])$ and $\mathcal{W}(r) := \{[1, r]\}$.
3. For $l = 2$ to r, do:
   If $w(l - 1) + b([l, r]) > w(r)$, then
   (a) Set $w(r) := w(l - 1) + b([l, r])$
   (b) Set $\mathcal{W}(r) := \mathcal{W}(l - 1) \cup \{[l, r]\}$
4. If $r < n$, then set $r := r + 1$ and go to Step 2.
5. STOP ($\mathcal{W}(n)$ is an optimal outcome and $\text{rev}(\mathcal{W}_{\text{OPT}}) = w(n)$).

PROOF OF THEOREM 8. Obviously, the presented algorithm needs $O(n^2)$ time. We will show by induction on n that $\mathcal{W}(n)$ is an optimal outcome. Clearly, $\mathcal{W}(1) = \{[1,1]\}$ is an optimal outcome when $n = 1$. Suppose that $\mathcal{W}(m)$ is an optimal outcome for the auction of first m assets whenever $m < n$. Let $\mathcal{W}_{\text{OPT}}$ be an optimal outcome for the auction of A. Let $\mathcal{W}^* := \mathcal{W}_{\text{OPT}} \backslash \{[m + 1, n]\}$ where $[m + 1, n]$ is the unique set from $\mathcal{W}_{\text{OPT}}$ containing asset n. Note that $\mathcal{W}^*$ is an outcome for the auction of $[1, m]$. From Proposition 1 and the induction hypothesis, $\text{rev}(\mathcal{W}^*) = w(m)$ and $\text{rev}(\mathcal{W}_{\text{OPT}}) = w(m) + b([m + 1, n]) \le w(n)$. The last inequality follows from the Step 3 of the algorithm when $l = m + 1$ and $r = n$. By optimality of $\mathcal{W}_{\text{OPT}}$, we conclude that the last inequality can be replaced by equality and $\mathcal{W}(n)$ is an optimal schedule. Hence, (a) follows.

Any outcome $\mathcal{W}$ contains at most one set $[i, j]$ where $i > j$ because 1 and n are in every such set and sets in $\mathcal{W}$ are disjoint. If an outcome $\mathcal{W}$ contains such $[i, j]$ then, for any other $[l, r] \in \mathcal{W}$, $j < l \le r < i$ (because $\mathcal{W}$ is a collection of disjoint sets). If we rename all the assets $k < i$ into $k' := k + n$, then A becomes $\{i, i + 1, \ldots, n, (n + 1), \ldots, n + i - 1\}$, $[i, j]$ becomes $[i, n + j]$, and any other $[l, r] \in \mathcal{W}$ becomes $[n + l, n + r]$. Hence, the algorithm will determine an optimal outcome for the auction of $X_i := \{i, i + 1, \ldots, n, (n + 1), \ldots, n + i - 1\}$ where $\mathcal{P} := \{[l, r] : i \le l \le r \le n + i - 1\}$. Since every outcome $\mathcal{W}$ is an outcome for the auction of X_i (Find $[i, j] \in \mathcal{W}$ that contains 1), it suffices to compare optimal outcomes for auctions of X_i. All this can be done in $O(n^3)$ time. □

PROOF OF THEOREM 9. We will show that the problem is NP-complete even if $\mathcal{P}$ consists of all two by two rectangles only (i.e., $\mathcal{P} := \{[a, a + 1] \times [c, c + 1]: 1 \le a < m \text{ and } 1 \le a < n\}$).[22]

As was shown in Berman et al. (1990), the *optimal* 2×2 *salvage problem* is NP-complete. The input for this problem is an $n \times n$ grid and some set S of unit squares with integer coordinates. A 2×2 rectangle with integer vertices is called functional if all four of its unit squares belong to S. The problem is to determine a maximal number of functional nonoverlapping (disjoint) 2×2 rectangles.

Any set S of unit squares with integer coordinates defines $b(C)$ for every $C \in \mathcal{P}$ in a natural way: set $b(C) := 1$ if C is functional and set $b(C) := 0$ if C is not functional. Then finding $\mathcal{W}_{\text{OPT}}$ is equivalent to solving the optimal 2×2 salvage problem. □

[22] This is equivalent to the assumption that $b(C) = 0$ for all other rectangles C. Hence, we will show that problem is NP-complete even in this restricted case.

References

Ahuja, R. K., T. L. Magnanti, J. B. Orlin 1993. *Network Flows*. Prentice Hall, Englewood Cliffs, NJ.

Banks, J., J. O. Ledyard, D. Porter 1989. Allocating uncertain and unresponsive resources: an experimental approach. *RAND J. Econom.* **20** 1–23.

Bartholdi, J. J. III, J. B. Orlin, D. H. Ratliff 1980. Cyclic scheduling via integer programs with circular ones. *Oper. Res.* **28** 1074–1085.

Berman, F., D. Johnson, T. Leighton, P. W. Shor, L. Snyder 1990. Generalized planar matching. *J. Algorithms*. **11** 153–184.

Bondy, J. A., U. S. R. Murty 1976. *Graph Theory With Applications*. American Elsevier, New York.

Branco, F., 1995. Multi-object auctions: on the use of combinational bids. Discussion Paper 1216, Center for Economic Policy Research, London.

Bykowsky, M. M., R. J. Cull, J. O. Ledyard 1995. Mutually destructive bidding: the FCC auction design problem. Social science working paper 916, California Institute of Technology, Pasadena.

Campbell, C. M. 1996. Efficient auctions of multiple objects. Working paper, Dept. of Economics, Ohio State University.

Cormen, T. C., C. E. Leiserson, R. L. Rivest 1990. *Introduction to Algorithms*. MIT Press, Cambridge, MA.

Edmonds, J. 1965. Maximum matching and a polyhedron with 0,1 vertices. *J. Res. Nat. Bur. Standards*. **69B** 125–130.

Federal Communications Commission. 1993. In the matter of implementation of section 309(j) of Communication Act, competitive bidding: notice of proposed rule making. PP Docket No. 93-253, Released October 12.

Garey, M. R., D. S. Johnson 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco .

Harstad, R. M., M. H. Rothkopf 1995. Combinatorial auctions with synergies. Working paper, RUTCOR, Rutgers University, New Brunswick, NJ.

Karp, R. M. 1972. Reducibility among combinatorial problems. R. E. Miller, J. W. Thatcher, eds., *Complexity of Computer Computations*, Plenum Press, New York. 85–103.

Krishna, V., R. W. Rosenthal 1996. Simultaneous auctions with synergies. *Games Econom. Behav.* **17** 1–31.

Lawler, E. L. 1976. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, New York.

Ledyard, J. O., D. Porter, J. Rangel 1994. Using computerized exchange systems to solve an allocation problem in project management. *J. Organ. Comput.* **4** No. 3.

McAfee, R. P. 1993. Auction design for personal communications services. PacTel Exhibit in PP Docket No. 93-253.

McCabe, K. S., S. Rassenti, V. L. Smith 1991. Smart computer assisted markets. *Science* **254**, 534–538.

McCurdy, R. C. 1981. Private communication. November 3.

McMillan, J. 1994. Selling spectrum rights. *J. Econom. Perspect.* **8** 145–162.

Mercer, A., D. Tielin 1996. Competitive tendering strategies in the bus industry. *J. Oper. Res. Soc.* **47** 1452–1460.

Milgrom, P. R., R. B. Wilson 1993. Affidavit of Paul R. Milgrom and Robert B. Wilson before the Federal Communications Commission, Docket No. 93-253, November 8.

Murty, K. G. 1992. *Network Programming*. Prentice Hall, Englewood Cliffs, NJ.

Papadimitriou, C. H. 1994. *Computational Complexity*, Addison-Wesley, Reading, MA.

Pulleyblank, W. R. 1995. Matchings and extensions. R. L. Graham, M. Grötschel, L. Lovász, eds. *Handbook of Combinatorics*. Vol. 1. MIT Press, Cambridge, MA. 179–232.

Rassenti, S. J., V. L. Smith, R. L. Bulfin 1982. A combinatorial auction mechanism for airport time slot allocation. *Bell J. Econom.* **13** 402–417.

Rothkopf, M. H. 1977. Bidding in simultaneous auctions with a constraint on exposure. *Oper. Res.* **25** 620–629.

Rothkopf, M. H. 1983. Bidding theory: the phenomena to be modeled. R. Engelbrecht-Wiggans, M. Shubik, R. Stark, eds. *Auctions, Bidding and Contracting: Uses and Theory*. New York University Press, New York. 105–120.

Rosenthal, R. W., R. Wang. 1966. Simultaneous auctions with synergies and common values. *Games Econom. Behav.* **17** 32–55.

Schackmann, A. 1989. Private communication.

Schrijver, A. 1986. *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester.

Smith, S. A., M. H. Rothkopf. 1985. Simultaneous bidding with a fixed charge if any bid succeeds. *Oper. Res.* **33** 28–37.

Accepted by Edward H. Kaplan; received September 11, 1995. *This paper has been with the authors* 8 *months for* 2 *revisions.*