Combinatorial Auction: A Survey (Part I)

Sven de Vries Rakesh V. Vohra

IJOC, 15(3): 284-309, 2003

Presented by James Lee

on May 10, 2006 for course Comp 670O, Spring 2006, HKUST

Outline

Introduction

What is a *combinatorial auction*?

Example: Auction off a dining set - one table and four chairs:

- Auction off the entire dining set?
- Five seperate auctions for each piece?

Because of complimentary or substitution effects, bidders have preferences not just for particular items but for sets of items, sometimes called *bundles*.

Definition

A combinatorial auction is an auction where bidders are allowed to bid on *combinations* of items, or bundles, instead of individual items.

Introduction
Applications

Examples

These are some applications of combinatorial auctions:

- Radio spectrum right (Jackson, 1976)
- Airport time slot allocation (Rassenti et al., 1982)
- Financial securities (Srinivasen et al., 1998)
- FCC's "Nationwide Narrowband Auction" of spectrum right in 1994, where auctions were run in parallel (Cramton, 2002)
- Sears, to select carriers (Ledyard et al., 2000), SAITECH-INC, Logistics.com, etc.
- Trading system by OptiMark Technologies allows bidder to submit price-quantity-stock triples

Introduction
Design Issues

Objective of auctioneer: To maximize revenue or *economic efficiency*?

- Restrict the collection of bundles on which bids are allowed?
- Single round - How should the bundles be allocated? What should the payment rules be?
- Multiple rounds (called iterative) - What information should be revealed to bidders from one round to the next?

Other considerations: Speed, practicality, bidder preferences, discouraging collusion and encouraging competition.

Introduction
Design Issues

Objective of auctioneer: To maximize revenue or *economic efficiency*?

- Restrict the collection of bundles on which bids are allowed?
- Single round - How should the bundles be allocated? What should the payment rules be?
- Multiple rounds (called iterative) - What information should be revealed to bidders from one round to the next?

Other considerations: Speed, practicality, bidder preferences, discouraging collusion and encouraging competition.

Main problems that auction designers must resolve:

- Bid expression
- Allocation of bundles among bidders
- Incentive issues

Bid Expression

- There are $2^m - 1$ nonempty subsets of m items.
- It is impractical for bidders to submit bid for every bundle.
- Therefore, bidders will only be allowed to submit bid for a limited number of bundles.

Even though the number of bundle is limited, the list could still be quite large. How do bidders communicate this large list to the auctioneer?

- Use an “oracle”, which is a program that computes the bid for a given bundle
- Auctioneer may specify a *bidding language*, which bidders must use to encode their preferences
- Restrict the collection of bundles on which bidders might bid

The Combinatorial Auction Problem
Formulation as Integer Program

Notations:

- N = set of n bidders
- M = set of m *distinct* objects
- $b_j(S)$ = the price that agent $j \in N$ will pay for $S \subseteq M$
- $y_j(S) = 1$ if $S \subseteq M$ is allocated to agent $j \in N$

$$\begin{aligned}\text{Maximize} \quad & \sum_{j \in N} \sum_{S \subseteq M} b_j(S)\, y_j(S) \\ \text{subject to} \quad & \sum_{S \ni i} \sum_{j \in N} y_j(S) \leq 1 \; \forall i \in M, \\ & \sum_{S \subseteq M} y_j(S) \leq 1 \; \forall j \in N\end{aligned}$$

Call this formulation CAP1.

The Combinatorial Auction Problem
Formulation as Integer Program

Definition

A function f is superadditive if $S \cap S' = \emptyset \Rightarrow f(S \cup S') \geq f(S) + f(S')$.

If the bid functions b_j are superadditive, an alternative formulation is:

- $b(S) = \max_{j \in N} b_j(S)$
- $x_s = 1$ if the highest bid on S is accepted

$$\text{Maximize } \sum_{S \subseteq M} b(S)\, x_S$$

$$\text{subject to } \sum_{S \ni i} x_s \leq 1 \; \forall i \in M$$

Call this formulation CAP2.

The Combinatorial Auction Problem
Multi-unit Combinatorial Auction

- CAP1 assume the objects are distinct.
- When there are multiple copies of the same object, bidders may want more than one copy of the same unit.
- In this case, CAP1 can be extended to a *multi-unit combinatorial auction*. (Leyton-Brown et al., 2000a; Gonen and Lehmann, 2000)

Notations:

- m_i = number of units of object i available
- $q = (q_1, q_2, \dots, q_m)$, q_i = number of units of object i demanded
- $\Omega_j \subseteq \mathbb{Z}^M \cap \prod_{i \in M}[0, m_i]$ is the restriction of bundles on agent j
- $y_j(q) = 1$ if the bundle $q \in \Omega_j$ is allocated to agent j
- P_j^A, P_j^B = polyhedra of feasible solutions under extra constraints.

The Combinatorial Auction Problem
Multi-unit Combinatorial Auction

$$\text{Maximize} \sum_{j\in N}\sum_{q\in\Omega_j} b_j(q)\, y_j(q)$$

$$\text{subject to} \sum_{j\in N}\sum_{q\in\Omega_j} y_j(q)\, q_i \le m_i \;\forall i \in M \qquad (\text{GCAP}_1)$$

$$y_j(q) \in P_j^A \;\forall j \in N \qquad (\text{GCAP}_2)$$

$$y \in P^A \qquad (\text{GCAP}_3)$$

$$y_j(q) \in P_j^B \;\forall j \in N \qquad (\text{GCAP}_4)$$

Set-Packing Problem

Definition

SPP: Given a set M and a collection V of subsets with nonnegative weights, find the largest-weight collection of pairwise disjoint subsets.

Formulation as an integer program:

- $x_j = 1$ if the jth set in V with weight c_j is selected
- $a_{ij} = 1$ if the jth set in V contains element $i \in M$

$$\text{Maximize} \sum_{j \in V} c_j x_j \qquad \text{subject to} \sum_{j \in V} a_{ij} x_j \leq 1 \ \forall i \in M$$

$$\text{or maximize } c \cdot x \qquad \text{subject to } Ax \leq 1$$

CAP is an instance of SPP. (Rothkopf et al., 1998; Sandholm, 1999)

Set-Packing Problem
Related Problems

There are two related problems:

- *Set-partitioning problem* (SPA): Maximize $c \cdot x$ subject to $Ax = 1$
- *Set-covering problem* (SCP): Minimize $c \cdot x$ subject to $Ax \geq 1$

Example

Auctions used in the transport industry are of the set-covering type:

- Objects are origin-destination pairs, called lanes.
- Bidders submit bids on bundles of lanes.
- The auctioneer wishes to choose a collection of bids of lowest cost such that all lanes are served.

SPA and SCP are similar to SPP, however, they have different computational and structural properties. (Balas and Padberg, 1976)

Complexity of SPP

Solving SPP by enumerating all possible 0-1 solutions:

- $2^{|V|}$ solutions, where $|V|$ is the number of variables
- Impractical for all but small values of $|V|$
- SPP is $\mathcal{NP}$-hard (More precisely, the recognition version of SPP is $\mathcal{NP}$-complete)

Complexity of SPP

Solving SPP by enumerating all possible 0-1 solutions:

- $2^{|V|}$ solutions, where $|V|$ is the number of variables
- Impractical for all but small values of $|V|$
- SPP is $\mathcal{NP}$-hard (More precisely, the recognition version of SPP is $\mathcal{NP}$-complete)

For the CAP,

- *Number of bids* is exponential in number of items m
- Any algorithm that is polynomial in the input size (here, the number of bids) would be exponential in m.
- Effective solution procedures for the CAP relies on:
  - Small number of distinct bids structured in computationally useful ways.
  - The underlying SPP can be solved reasonably quickly.

Solvable Instances of SPP

Definition

A polyhedron is integral if all its extreme points are integral.

- If the polyhedron $P(A) = \{\, x \mid Ax \leq 1,\ x \geq 0 \,\}$, is integral, the SPP can be solved as a linear program.
- In general, linear programs can be solved in polynomial time.
- In most cases, because of the special structure of these problems, algorithm more efficient than linear-programming ones exist.
- However, the connection CAP to linear programming is important because it allows one to interpret dual variables as prices for the objects being auctioned.
- Some sufficient conditions has been identified for a polyhedron to be integral.

Solvable Instances of SPP
Total Unimodularity

Definition

A matrix is totally unimodular (TU) if the determinant of every square submatrix is 0, 1 or -1.

If $A = \{a_{ij}\}_{i \in M, j \in V}$ is TU, all extreme points of the polyhedron $P(A)$ are integral. (Nemhauser and Wolsey, 1988)

Example

- A 0-1 matrix has the *consecutive-ones property* if the nonzero entries in each column occur consecutively. They are TU matrices.
- Suppose the objects are parcels of land along a shore line.
- The most interesting combinations would be contiguous.
- Number of distinct bids limited by a polynomial in no. of the objects.
- The constraint matrix A of CAP2 has consecutive-ones property.

Solvable Instances of SPP
Balanced Matrices

Definition

A 0-1 matrix is balanced if it has no square submatrix of odd order with exactly two 1s in each row and column.

If B is balanced, the linear program $\max\{\, c \cdot x \mid Bx \leq 1, x \geq 0 \,\}$ has an integral optimal solution whenever c is integral. (Schrijver, 1986)

Example

- Consider a tree T with a distance function d, in which vertices represent parcels of land connected by a road network with no cycles.
- Let $N(v, r)$ be the set of vertices in T that are within distance r of v.
- Bidders can bid for subset of vertices constrained in the form $N(v, r)$.
- The constraint matrix has one column for each $N(v, r)$ and one row for each vertex of T. It is balanced.

Solvable Instances of SPP
Perfect Matrices

Definition

A graph is perfect if its chromatic number equals its clique number.

More generally, if the constraint matrix A can be identified with the vertex-clique adjacency matrix of a perfect graph, then SPP can be solved in polynomial time. (Grötschel et al., 1988, Chapter 9)

Example

- Consider a tree T as in previous example.
- Bidders can bid for *any* connected subset of parcels.
- The constraint matrix will have one column for each connected subset of T and one row for each vertex of T. This matrix is perfect.

Solvable Instances of SPP
Graph-Theoretic Methods

When the constraint matrix A admits a graph-theoretic interpretation in terms of an easy problem, the underlying SPP can be solved in polynomial time even if $P(A)$ is not integral.

- The best-known instance is each column of A has at most two 1s.
- In this case, the SPP becomes an instance of *maximum-weight matching problem* in a graph, which can be solved in polynomial time.
- Row $\Leftrightarrow$ vertex, column $\Leftrightarrow$ edge
- $P(A)$ contains fractional extreme points. (e.g. K_3)
- However, if the number of 1s in each *row* is restricted instead of in each column, the instance correspond to the *stable-set problem* in graphs, which is difficult to solve.

Solvable Instances of SPP
Graph-Theoretic Methods

Definition

A 0-1 matrix has the circular-ones property if the nonzero entries in each column (row) are consecutive, where the first and last columns (rows) are treated consecutively.

- The constraint matrix can be identified with the vertex-clique adjacency matrix of a circular arc graph.
- The SPP becomes the *maximum-weight independent set problem* for a circular arc graph, which can be solved in polynomial time. (Golumbic and Hammer, 1988)

Example

Objects that are parcels of land lying on the shore of an island.

Solvable Instances of SPP
Using Preferences

Restriction in the preference orderings of the bidders:

- Two types of bidders
- N_r, set of type-r bidders, have bid functions $b_j = g_r$ $(r = 1,\, 2)$
- g_r are *nondecreasing* $(S \subseteq T \Rightarrow g_r(S) \leq g_r(T))$, integer-valued, and *supermodular* $(g_r(S) + g_r(T) \leq g_r(S \cup T) + g_r(S \cap T))$ functions.

The dual to the linear programming relaxation of CAP1 is:

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

$$\text{subject to } \sum_{i \in S} p_i + q_j \geq g_r(S) \;\forall S \subseteq M, j \in N_r \;(r = 1, 2)$$

$$p_i, q_j \geq 0 \;\forall i \in M, j \in N$$

Solvable Instances of SPP
Using Preferences

- This problem is an instance of the *polymatroid-intersection problem*.
- It is polynomially solvable. (Grötschel et al., 1988, Theorem 10.1.13)
- It is *totally dual integral*, which means the primal problem, the LP relaxation of CAP1, has an integer optimal solution. (Bikhchandani and Mamer, 1997)
- However, when there are more than two types of bidders, the dual problem admits fractional extreme points.
- Finding an integer optimal solution for the intersection of three or more polymatroids is $\mathcal{NP}$-hard (Papadimitriou and Steiglitz, 1982, Section 12.6.3)

Solvable Instances of SPP

Gross Substitutes Property

Notations:

- $v_j(S)$ = valuation of $S \subseteq M$ by bidder j
- $p = (p_1, \dots, p_k)$, price vector of objects
- $D_j(p)$ = collection of subsets of objects that maximize bidder j's utility, defined by:

$$D_j(p) = \left\{ S \subseteq M \;\middle|\; v_j(S) - \sum_{i \in S} p_i \geq v_j(T) - \sum_{i \in T} p_i \ \forall T \subseteq M \right\}.$$

The *gross-substitutes condition* requires that: For all p, p' with $p' \geq p$, and all $A \in D_j(p)$, there exists $B \in D_j(p)$ such that $\{ i \in A \mid p_i = p'_i \} \subseteq B$.

Solvable Instances of SPP
Gross Substitutes Property

Example

Auctions where bidders are interested in multiple units of the same item and have diminishing marginal utilities.

- LP relaxations of CAP1 and CAP2 have optimal integer solutions.
- There exist a primal-dual algorithm for the linear relaxation of CAP1.
- Gross-substitutes is related to $M^{\natural}$-concavity. (Murota and Tamura, 2000)
- From results of $M^{\natural}$-concave functions, CAP1 can be solved in time polynomial in the number of objects under the assumption of gross substitutes by using a proper oracle.

Exact Methods
Relaxation and Column Generation

- An *exact method* gives a feasible and optimal solution to the SPP and the CAP.
- Three varieties: branch and bound, cutting planes, branch and cut
- Upper bound on the optimal solution value is obtained by solving a *relaxation* of the optimizaton problem.
- For SPP: Lagrangean relaxation and linear-programming relaxation

Exact Methods
Relaxation and Column Generation

- An *exact method* gives a feasible and optimal solution to the SPP and the CAP.
- Three varieties: branch and bound, cutting planes, branch and cut
- Upper bound on the optimal solution value is obtained by solving a *relaxation* of the optimizaton problem.
- For SPP: Lagrangean relaxation and linear-programming relaxation
- In addition, since number of columns (bids) is huge, the columns are generated when need. This technique is called *column generation*. (Gilmore and Gomory, 1961; Barnhart et al., 1998)
- Optimal solution to an instance of SPA with 1053127 variables and 145 constraints (correspond to a problem with 145 items and 1053127 bids) is solved under 25 minutes. (Hoffman and Padberg, 1993)

Exact Methods

Software used to solve CAP Exactly

- Logistic.com's Optibid$^{\text{TM}}$, based on integer-program solver with a series of proprietary formulations and starting heuristic algorithms
- SAITECH-INC's SBID, also based on integer programming
- Rothkopf et al. (1998) use dynamic programming
- Sandholm (1999) and Fujishima et al. (2000) used refinements by pruning and additional bounding heuristics
- Commerically available linear integer program solver (CPLEX) can solve instances of up to 19 items, with 2^{19} variables. This test case is similar to those considered by Sandholm and Fujishima et al.

Approximate Methods

- Every heuristic approach for solving general integer-programming problems has been applied to the SPP.
- However, there were no comparative testing across such methods.
- Anything one can think of for approximating the SPP has probably been thought of.
- It is possible to embed approximation algorithms within exact algorithms to get good approximation to lower bound, and then iteratively tightens the upper bound.

Theorem

No polynomial-time algorithm for the SPP that can deliver a worst case ratio larger than $n^{\epsilon-1}$ for any $\epsilon > 0$ unless $\mathcal{ZPP} = \mathcal{NP}$. (Håstad, 1999)

Approximate Methods
Analysis and Testing

- However, polynomial algorithms that have a worst-case ratio of $O(n/(\log n)^2)$ are known. (Boppana and Halldórsson, 1992)
- Bounds that are functions of the data of the underlying CAP1 are also known. (Akcoglu et al., 2002)
- Worst-case analysis does not represent the "typical" accruacy of an approximation algorithm.
- Probabilistic analysis, on the other hand, compare the probability distribution over the value of the optimal and approximate solution.
- Some approximation algorithms are too complicated to be analyzed and have to resort to empirical testing. However, it is not easy to select good test problems that proximal to the optimal solution.

Summary

- Introducing the CAP
- Formulations of CAP
- SPP and complexity
- Solvable instances
- Exact and approximate methods