Combinatorial Auctions

Final Project Report for Combinatorial Optimization, Fall 2020

Joshua Peter Ebenezer

December 10th, 2020

Contents

1 Introduction **2**

2 Formulation **2**

3 Exact methods **3**

4 Relaxations **3**

4.1 Total Unimodularity 3

4.2 Balanced matrices 4

4.3 Perfect matrices 5

4.4 Graph Theoretic methods 5

4.5 Supermodular bidding functions 5

5 Approximate methods **6**

5.1 Submodular bidding functions with the value oracle model 6

5.1.1 Preliminaries 6

5.1.2 The Submodular Welfare Problem 8

5.1.3 Continuous Greedy Process 10

5.1.4 Continuous Greedy Process for submodular maximization 11

5.2 Submodular Welfare Problems for the Demand Oracle model 12

6 Conclusion **15**

Abstract

A combinatorial auction is a type of auction in which bidders can assign values to combinations of items, instead of only assigning values to individual items. These are useful in situations where bidders place more or less value to a combination of items than to the sum of the individual values of those items. In this project report, some of the main results in combinatorial auctions are studied in an expository fashion, with particular attention given to a subclass of combinatorial auctions called the submodular welfare problem.

1 Introduction

Many auctions involve the sale of a group of items that may be more valuable than the sum of the individual values of those items. Examples include the auctioning of spectrum for telecommunication, flight routes, network routing, delivery routes, railroad routes, etc. The problem of identifying which sets of bids to accept is called the combinatorial auction problem (CAP) [1], and will be studied here. The typical objective is to maximize the social welfare, i.e. the pareto-optimal solution that assigns sets to the bidders such that the total benefit to all the bidders is maximized.

I first present the formulation of the problem, and some specific cases in which polynomial time solutions can be achieved.

2 Formulation

Let N be the set of bidders and M be the set of m distinct objects. For every subset S of M let $c^j(S)$ be the bid that agent $j \in N$ is willing to place for S. One can assume that $c^j(S) \geq 0$. Let $y(S, j) = 1$ if the set of objects S is allocated to $j \in N$ and 0 otherwise. The problem can be written as

$$\begin{aligned} \max \quad & \Sigma_{j \in N} \Sigma_{S \in M} c^j(S) y(S, j) \qquad (1) \\ \text{s.t.} \quad & \Sigma_{S \ni i} \Sigma_{j \in N} y(S, j) \leq 1 \quad \forall i \in M \\ & \Sigma_{S \in M} y(S, j) \leq 1 \quad \forall j \in N \\ & y(S, j) = 0, 1 \quad \forall S \subseteq M, j \in N \end{aligned}$$

The objective is to maximize the total reward. The first constraint ensures that the same item is not assigned to different bidders (i.e., assigned sets don't overlap). The second constraint ensure that each bidder never receives more than one set. This problem is NP-complete, and this can be shown by reductions from two well-known NP-complete problems. Firstly, if the bidding functions are superadditive, this formulation is equivalent to the set packing problem (SPP).

SET PACKING PROBLEM: Given a set M of m elements and a collection V of subsets with non-negative weights, find the largest weight collection of subsets that are pairwise disjoint.

Define $x_j = 1$ if the jth set in V with weight c_j is selected and $x_j = 0$ otherwise, and define $A_{ij} = 1$ if the jth set in V contains element $i \in M$. SPP can be formulated as:

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq 1 \\ & x_j = 0, 1 \quad \forall j \in N \end{aligned} \tag{2}$$

If we treat the subsets as the collection of objects being bid for and the weights as the values of the bids, SPP is the same as CAP. SPP is NP-hard, and therefore the CAP is also NP-hard. CAP can also be cast as a reduction of the NP-complete Independent Set problem.

3 Exact methods

Exact solutions can be found by branch and bound algorithms or cutting plane algorithms. Upper bounds are found by solving Lagrangean relaxations of CAP, by relaxing the 0-1 constraint and by moving other constraints to the optimization objective with a Lagrangean penalty term. The relaxed problem would be:

$$\begin{aligned} Z(\lambda) = \max \quad & c^T x + \lambda^T (1 - Ax) \\ \text{s.t.} \quad & 0 \leq x \leq 1 \end{aligned} \tag{3}$$

For a given value of λ, one can find $Z(\lambda)$ easily. Notice that

$$\begin{aligned} Z(\lambda) &= \max \quad c^T x + \lambda^T (1 - Ax) \ \text{ s.t. } \ 0 \leq x \leq 1 \\ &= \max \quad (c - A^T \lambda)^T x + \lambda^T 1 \ \text{ s.t. } \ 0 \leq x \leq 1 \end{aligned} \tag{4}$$

This will be maximized if we set $x_j = 1$ when $c_j - (A^T \lambda)_j > 0$ and 0 otherwise. Any value of $Z(\lambda)$ is an upper bound to CAP. Define

$$Z_{LP} = \min_{\lambda} Z(\lambda) \tag{5}$$

The value of Z_{LP} can be found by the subgradient algorithm iteratively, and clearly provides the best upper bound to the solution of CAP from the Lagrangean formulation.

4 Relaxations

4.1 Total Unimodularity

A matrix is said to be totally unimodular (TU) if the determinant of every square submatrix is 0 or ± 1. If the matrix A in equation 2 is TU, then the solution to CAP is integral.
Proof: According to Cramer's rule, the extreme points are given by

$$x_j = \frac{\det A_j}{\det A} \quad j = 1, 2 \ldots N \tag{6}$$

where A_j is the matrix formed by replacing the jth column of A with 1. The denominator is ± 1 if the problem is feasible and the numerator is a sum of 1s and -1s since the submatrices used to calculate the determinant along the jth row all have values in 0 or ± 1. Therefore this fraction is integral.

A special case of totally unimodular matrices can occur when the items being bid are contiguous in some sense. For example, if the items are plots of land, contiguous plots would be more favorable to bidders, and hence subsets with adjacent items (plots of land) would be more likely to appear in the set of subsets with non-negative weights (i.e., sets which are of interest to bidders). This would mean that columns in A would have adjacent ones. Such a matrix has the **consecutive-ones property**, and is totally unimodular.

Another example can be found from bipartite graphs. Adjacency matrices of bipartite graphs are totally unimodular. A constraint matrix corresponding to bipartite graphs would imply that the sets with non-negative weights (i.e. important sets) are those with bidding items from two disjoints sets. These two sets could be complementary in some way. An example could be air routes between the East Coast and West Coast. Routes between locations on the East Coast or the West Coast could be of less importance, while routes that connect the coasts would be more important.

4.2 Balanced matrices

A 0, 1 matrix is balanced if it does not contain a square submatrix of odd order with exactly two ones per row and per column. If A is balanced, the solution to 2 is integral.
Proof: Assume that this is not true, and that A is the minimal size matrix for which this is not true, i.e. that for every proper submatrix of A, (2) has an integral solution. Assume that x^* is a fractional extreme point. Since $x^* < \mathbf{1}$, at least two elements of each row of A have to be 1. Let A' be the matrix obtained from removing the first row from A. By our assumption that A is minimal, the extreme points corresponding to A' are integral. Let $P(A')$ be the polytope corresponding to A'. Since $x^* \in P(A')$ (i.e. it belongs to a face of dimension 1), it must be a linear combination of some extreme points x_S and x_T.

$$x^* = \lambda x^S + (1 - \lambda)x^Q \tag{7}$$

Let $S = \{j : x_j^S = 1\}$ and $T = \{j : x_j^Q = 1\}$. Since $0 < x_j^* < 1 \forall j$, $S \cap T = \phi$ and $S \cup T = E$, where E is the set of labels for the columns of A. Since $Ax^S = 1$ and $Ax^Q = 1$, each row of A other than the first row has exactly two 1s. A similar argument can be applied to each row to show that all rows have exactly two 1s. Since A is balanced, it is the incident-edge matrix of a bipartite graph. Since it is square and has two ones in each row, the graph has an even circuit, which contradicts the non-singularity of A. QED.

This case can arise for CAP in some instances. For example, consider that some locations are being bid for, and that these locations are connected by a transportation network with no cycles. Denote the tree that represents these locations as T, and define a distance measure d between vertices (locations) on the tree. For each vertex v in T, define $D(v, r)$ as the set of all vertices (locations) which are less than r units away from v in terms of the distance measure. Bidders are constrained to bid for sets that are of the form $D(v, r)$ for some vertex v and some number r. This could be a constraint or something they choose themselves to minimize transportation costs. The constraint matrix of the corresponding problem will have one column for each set of the form $D(v, r)$ and one row for each vertex in the tree. Since there are no cycles in the graph, $D(v, r_1) \subseteq D(v, r_2)$ for $r_1 \leq r_2$, and therefore the constraint matrix cannot have any odd submatrix with exactly two ones in its rows and columns. Such a matrix is balanced.

4.3 Perfect matrices

If the constraint matrix A is the vertex-clique adjacency matrix of a perfect graph, (2) has an integer solution. A perfect graph is is a graph in which the chromatic number of every induced subgraph equals the order of the largest clique of that subgraph. The smallest number of colors needed to color a graph G is called its chromatic number. A clique is a subset of vertices of an undirected graph such that every two distinct vertices in the clique are adjacent.

A case in which perfect matrices can appear as constraint matrices for CAP is as follows. Consider the case of balanced matrices, where there is a tree of locations with no cycles. If bidders are constrained to bid for any connected subset of vertices, the resulting constraint matrix would be perfect.

4.4 Graph Theoretic methods

There are situations where the extreme points are not necessarily all integral but the problem can be solved in polynomial time. If A has at most two 1s in each column (i.e. not more than two elements in each set of interest) then A can be interpreted as the incidence matrix of a graph, with the rows representing vertices and the columns representing edges. CAP then reduces to the maximum weight matching problem , which can be solved in polynomial time.

4.5 Supermodular bidding functions

The previous methods are relaxations of the problem based on the composition of the sets. One could also place restrictions on the bidding functions c^j. A common restriction that is placed is non-decreasing supermodularity. The cost function c is non-decreasing if $c^j(S) \leq c^j(T) \;\; \forall S \subseteq T$. c^j is supermodular if $c^j(S) + c^j(T) \leq c^j(S \cap T) + c^j(S \cup T)$.

Assume we have two supermodular functions g^1 and g^2, such that one class N^1 of bidders has bidding functions $c^j(.) = g^1(.)$ for $j \in N^1$, and another class N^2 of bidders has bidding functions $c^j(.) = g^2(.)$ for $j \in N^2$. Then the dual of the linear programming relaxation of (1) can be written as

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

This problem is an instance of the polymatroid-intersection problem and is polynomially solvable. It has the property of being totally dual integral, which implies that its linear programming dual (which is the linear programming relaxation of (1)) has an integer solution. This method cannot be extended to 3 or more classes of supermodular bidding function, as the problem of finding an integer optimal solution for 3 or more polymatroids is known to be NP-hard. Supermodular functions arise in cases where there is an increasing return as the sets grow larger.

5 Approximate methods

It is known that approximating the SPP to within a factor of $m^{1/2-\epsilon}$ is NP-hard. However, just as for exact solutions, good approximates can be achieved for special cases of the problem. A special case of the bidding function which yields good approximations is monotone submodularity. Define

$$c_S(j) = c(S \cup \{j\}) - c(S) \tag{8}$$

as the marginal value of item j with respect to set S. c is monotone if $c_S(j) \geq 0 \quad \forall j$ and c is submodular if $c_S(j) \geq c_T(j)$ whenever $S \subset T$. This corresponds to the property of diminishing returns in economics, i.e. as the set grows larger the value added by new items becomes smaller. CAP with a submodular bidding function is also known as the submodular welfare problem. There area two kinds of submodular welfare problems, depending on the oracle:

1. **Value Oracle model**: An oracle that can answer the question "What is the value of $f(S)$"?
2. **Demand Oracle model**: An oracle that can answer the question: "Given an assignment of prices to items $p : [m] \to R$, what is $\max_S(f(S) - \sum_{j\in S} p_j$"?

The demand oracle is more powerful than the value oracle model.

5.1 Submodular bidding functions with the value oracle model

Vondrák showed that one can achieve a $1 - \frac{1}{e}$ approximation of the optimal value for the submodular welfare problem using a randomized continuous greedy algorithm for the value oracle model [4]. The proof is presented here.

5.1.1 Preliminaries

Smooth submodular functions A discrete function $f : 2^X \implies R$ is submodular if $f(S \cap T) + f(S \cup T) \leq f(S) + f(T)$ for any $S, T \subseteq X$. This definition can be extended to continuous functions as:

$$F(x \vee y) + F(x \wedge y) \leq F(x) + F(y) \tag{9}$$

where $(x \vee y)_i = \max\{x_i, y_i\}$ and $(x \wedge y)_i = \min\{x_i, y_i\}$. A continuous function can be said to be monotone if $F(x) \leq F(y)$ when $x \leq y$. The concept of smooth monotone submodularity

is now introduced.
DEFINITION: A function $F : [0,1]^X \to R$ is smooth monotone submodular if

- F has second partial derivatives everywhere.
- $\frac{\delta F}{\delta y_j} \geq 0 \;\; \forall j \in X$ everywhere (monotonicity)
- $\frac{\delta^2 F}{\delta y_j \delta y_j} \leq 0 \;\; \forall i, j \in X$ everywhere (submodularity).

A smooth monotone submodular function can be obtained from a monotone submodular function $f : 2^X \to R_+$: For $y \in [0,1]^X$, let $\hat{y}$ denote a random vector in $\{0,1\}^X$ where each coordinate is independently rounded to 1 with probability y_j or 0 otherwise. Then define

$$F(y) = \mathbf{E}[f(\hat{y})] = \sum_{R \subseteq X} f(R) \prod_{i \in R} y_i \prod_{j \notin R} (1 - y_j) \tag{10}$$

This function is linear in each variable y_i, and is hence a multilinear polynomial. The first derivative is

$$\frac{\delta F}{\delta y_j} = \mathbf{E}[f(\hat{y})|\hat{y}_j = 1] - \mathbf{E}[f(\hat{y})|\hat{y}_j = 0] \tag{11}$$

Note that each coordinate y_j is independent of all other coordinates. We also know that f is monotonic, which implies that $f(\hat{y})|\hat{y}_j = 1 \geq f(\hat{y})|\hat{y}_j = 0$. By these two properties, we can assert that

$$\frac{\delta F}{\delta y_j} = \mathbf{E}[f(\hat{y})|\hat{y}_j = 1] - \mathbf{E}[f(\hat{y})|\hat{y}_j = 0] \geq 0 \tag{12}$$

For $i \neq j$, the second derivative is

$$\frac{\delta^2 F}{\delta y_i \delta y_j} = \mathbf{E}[f(\hat{y})|\hat{y}_j = 1, \hat{y}_i = 1] + \mathbf{E}[f(\hat{y})|\hat{y}_j = 0, \hat{y}_i = 0] - \mathbf{E}[f(\hat{y})|\hat{y}_j = 0, \hat{y}_i = 1] - \mathbf{E}[f(\hat{y})|\hat{y}_j = 1, \hat{y}_i = 0] \tag{13}$$

Note that $(f(\hat{y})|\hat{y}_j = 1, \hat{y}_i = 1)$ corresponds to the maximum value of y (for each y_i) when y_i and y_j are the variables, and $(f(\hat{y})|\hat{y}_j = 0, \hat{y}_i = 0)$ corresponds to the minimum value of y when y_i and y_j are the variables. From 9 and the submodularity of f, we then determine that the sum of the first two terms are smaller than the sum of the last two terms. Therefore

$$\frac{\delta^2 F}{\delta y_i \delta y_j} \leq 0 \tag{14}$$

Since f is multilinear, when $i = j$

$$\frac{\delta^2 F}{\delta y_i^2} = 0. \tag{15}$$

These are the properties of a smooth monotone submodular function, and therefore this process can transform a monotone submodular discrete function into a smooth monotone submodular continuous function.

Matroids and matroid polytopes: A finite matroid M is a pair $(E, \mathcal{I})$, where E is a finite set (called the ground set) and $\mathcal{I}$ is a family of subsets of E (called the independent sets) with the following properties:

1. The empty set is independent, i.e., $\phi \in \mathcal{I}$.
2. Every subset of an independent set is independent, i.e., for each $A' \subseteq A \subseteq E$, if $A \in \mathcal{I}$ then $A' \in \mathcal{I}$. This is sometimes called the hereditary property, or the downward-closed property.
3. If A and B are two independent sets (i.e., each set is independent) and A has more elements than B, then there exists $x \in A \backslash B$ such that $B \cup \{x\}$ is in $\mathcal{I}$. This is sometimes called the augmentation property or the independent set exchange property.

For a matroid M, the matroid polytope of M is defined as

$$P(M) = \text{conv}\{\mathbf{1}_I : I \in \mathcal{I}\} \tag{16}$$

Another definition of matroid polytopes is

$$P(\mathcal{M}) = \{x \geq 0 : \forall S \in X; \sum_{j \in S} x_j \leq r_M(S)\} \tag{17}$$

where $r_M(S) = \max\{|I| : I \subseteq S \& I \in \mathcal{I}\}$ is the rank function of matroid M. Note that for any x, such that $0 \leq x \leq y, y \in P \implies x \in P$. This is called the down-monotone property.
Pipage rounding: Pipage rounding is a rounding technique, and it is used as a black box to prove the result. The following lemma is stated without proof.
LEMMA 0: *There is a polynomial-time randomized algorithm, which, given a membership oracle for matroid $M = (X, \mathcal{I})$, a value oracle for a monotone submodular function $f : 2^X \to \mathbb{R}_+$, and $y \in P(M)$ returns an independent set $S \in \mathcal{I}$ of value $f(S) \geq (1 - o(1))F(y) = (1 - o(1))\mathbf{E}[f(\hat{y})]$ with high probability.*

The $o(1)$ term can be made polynomially small in $n = |X|$. This lemma tells us that a function of a fraction $y \in P(M)$ (where $P(M)$ is the matroid polytope of M) can be converted to an integral one without significant loss in the objective function. Therefore we can optimize the continuous problem $\max\{F(y) : y \in P(M)\}$ instead of $\max\{f(S) : S \in I\}$, and once we find the optimum solution x^*, we can convert it into the optimum set S^* which will solve the CAP.

5.1.2 The Submodular Welfare Problem

As before, let the set of n players be N, the set of m items M, and for each $i \in N$, let $c^i : w^N \to \mathbb{R}_+$ be the respective utility function which maps each set of objects to the bidding values placed by the ith bidder. Define a new domain set $X = N \times M$. X contains all possible allocations to each bidder. Note than any set $S \subseteq X$ can be written uniquely as $S = \bigcup_{i \in N} (\{i\} \times S_i)$, where S_i is the set of items allocated to bidder i. Then define a function $f : 2^X \to \mathbb{R}_+$:

$$f(S) = \sum_{i \in N} c^i(S_i) \tag{18}$$

Note that this similar to the objective function in the original formulation in 1, with the difference that we make $|N|$ copies of each item, one for each player. In order to impose the constraint that in reality we can only use one copy of each item so that we are able to allocate them to different bidders, define the paritition matroid $\mathcal{M} = (X, \mathcal{I})$:

$$\mathcal{I} = \{S \subseteq X | \forall j; |S \cap (N \times \{j\})| \leq 1\} \tag{19}$$

Note that this family satisfies the properties of a family of independent subsets.
Proof:

1. $\phi \in \mathcal{I}$, because $\forall j; |\phi \cap (N \times \{j\})| = 0 < 1$
2. If $A' \subseteq A \subseteq E$, and $A \in \mathcal{I}$, then $|A' \cap (N \times \{j\})| \leq |A \cap (N \times \{j\})| \leq 1$, and therefore $A' \in \mathcal{I}$.
3. Let $(N \times \{j\}) = E_j$. If A and B are two independent sets (i.e., each set is independent) and A has more elements than B, $\exists E_j$ s.t. $|E_j \cap A| > |E_j \cap B|$. Then for any $x \in E_j \cap (A \backslash B)$, $|(B \cup x) \cap E_j| \leq |A \cap E_j| \leq 1$.

There $\mathcal{I}$ satisfies all the properties of a family of independent sets, and $\mathcal{M}$ is a matroid. QED.

The Submodular Welfare Problem is thus $\max\{f(S) : S \in I\}$. Due to pipage rounding, instead of solving this discrete problem we can solve the continuous optimization problem $\max\{F(y) : y \in N(\mathcal{M})\}$. Before describing the algorithm we will state and prove a lemma.
Lemma 1: Let $OPT = \max_{S \in \mathcal{I}} f(S)$. For any $y \in [0,1]^X$ and a random set R corresponding to $\hat{y}$, with elements sampled independently according to y_j. Then

$$OPT \leq F(y) + \max_{I \in \mathcal{I}} \sum_{j \in I} \mathbf{E}[f_{R(t)}(j)] \tag{20}$$

Proof: Let $OPT = f(O)$. Define

$$f_R(S) = f(R \cup S) - f(R) \tag{21}$$

and

$$f_R(j) = f(R \cup \{j\}) - f(R) \tag{22}$$

By submodularity $f_R(j) \geq 0$. Therefore $f(O) \leq f(R) + \sum_{j \in O} f_R(j)$ for any set R. Taking the expectation over any random set R,

$$\begin{aligned} OPT &\leq \mathbf{E}[f(R) + \sum_{j \in O} f_R(j)] \\ &= F(y) + \mathbf{E}[\sum_{j \in O} f_R(j)] \\ &\leq F(y) + \max_{I \in \mathcal{I}} \mathbf{E}[f_{R(t)}(j)] \end{aligned}$$

QED.

5.1.3 Continuous Greedy Process

Given matroid $\mathcal{M} = (X, \mathcal{I})$, the continuous greedy algorithm as followed:

1. Let $\delta = \frac{1}{(n)^2}$ where $n = |X|$. Start at $t = 0$ and $y(0) = 0$
2. Let $R(t)$ be a random set containing each item j independently with probability $y_j(t)$. For all j, estimate
$$\omega_{ij}(t) = \mathbf{E}[f_{R(t)}(j)] \tag{23}$$
by taking the average of n^5 independent samples.
3. Let $I(t)$ be a maximum weight independent set in $\mathcal{M}$ according to the weights $\omega_j(t)$. This can be found with a greedy algorithm. Let $y(t+\delta) = y(t) + \delta \mathbf{1}_{I(t)}$
4. Increment $t := t + \delta$, if $t < 1$ go back to Step 2. Otherwise return $y(1)$.

After this, we apply pipage rounding to $y(1)$ to obtain the optimal solution. The optimality of this algorithm is proved as follows:
Lemma 2: *The fractional solution y found by the continuous greedy algorithm satisfies with high probability*
$$F(y) = \mathbf{E}[f(\hat{y})] \geq (1 - \frac{1}{e} - o(1))OPT \tag{24}$$

Proof:
We start with $F(y(0)) = 0$ and analyze how much $F(y(t))$ increases for each step of the algorithm. Consider a random set $D(t)$, independent of $R(t)$, that contains each item j independently with probability $\Delta_j(t) = y_j(t\delta) - y_j(t)$. Therefore $\Delta(t) = y(t+\delta) - y(t) = \delta \mathbf{1}_{I(t)}$. This means that $D(t)$ is a random subset of $I(t)$ with each element appears independently with probability δ.
From the definition in (10), $F(y(t+\delta)) = E[R(t+\delta)]$. $R(t+\delta)$ contains items independently with probability $y(t+\delta) = y(t)+\delta\mathbf{1}_{I(t)} = y(t)+\Delta(t)$. An item does NOT belong to $R(t) \cup D(t)$ if it is in neither $R(t)$ or $D(t)$. Since $R(t)$ and $D(t)$ are independent, the probability that a item does not belong in either is $(1 - y_j(t))(1 - \Delta_j(t))$. Therefore $R(t) \cup D(t)$ contains items independently with probabilities $1-(1-y_j(t))(1-\Delta_j(t)) = y_j(t)+\Delta_j(t)-y_j(t)\Delta_j(t)$. This is smaller than the probabilities associated with $R(t+\delta)$. Therefore, by monotonicity, $F(y(t+\delta)) = E[R(t+\delta)] \geq E[R(t) \cup D(t)]$.
We thus have
$$\begin{aligned}
F(y(t+\delta)) - F(y) &= E[f(R(t+\delta))] - E[f(R(t))] \\
&\geq E[f(R(t) \cup D(t)) - f(R(t))] \\
&= \mathbf{E}[f_R(D(t))] \\
&\geq \sum_j Pr[(D(t) = \{j\}]\mathbf{E}[f_{R(t)}(j))] \\
&= \sum_{j \in I(t)} \delta(1-\delta)^{|I(t)|-1}\mathbf{E}[f_{R(t)}(j)] \\
&\geq \delta(1 - n\delta) \sum_{j \in I(t)} \mathbf{E}[f_{R(t)}(j)]
\end{aligned}$$

$I(t)$ is an independent set that maximizes $\sum_{j\in I}\omega_j(t)$, which is an estimate of $\sum_{j\in I}\mathbf{E}[f_{R(t)}(j)]$. Since we used n^5 samples, by the Chernoff bound,

$$P(\omega_j(t)-\mathbf{E}[f_{R(t)}(j)]\geq \frac{OPT}{n^2})\leq \exp(\frac{-2\frac{OPT^2}{n^4}}{n^5(b-a)^2})=\exp(\frac{-2OPT^2 n}{(b-a)^2}) \tag{25}$$

Hence the probability that the error in the estimate $\omega_j(t)$ of $\mathbf{E}[f_{R(t)}(j)]$ is more than OPT/n^2 is exponentially small in n since $OPT\geq \max_{R,j} f_R(j)$. I is the maximum sum of these estimates. Hence, w.h.p., the error in computing I is at most n times the error in computing each estimate, which gives OPT/n . Thus

$$\begin{aligned}
F(y(t+\delta))-F(y) &\geq \delta(1-n\delta)(\max_{I\in\mathcal{I}}\sum_{j\in I}\mathbf{E}[f_{R(t)}(j)]-OPT/n)\\
&\geq \delta(1-n\delta)(OPT-F(y(t))-OPT/n) \qquad \text{(from Lemma 1)}\\
&= \delta(1-1/n)(OPT-F(y(t))-OPT/n) \qquad (\text{since } \delta=\frac{1}{n^2})\\
&= \delta(\tilde{OPT}-F(y(t)) \qquad (\text{where } \tilde{OPT}=(1-2/n)OPT)
\end{aligned}$$

We subtract $\tilde{OPT}$ from both sides and rearrange this to get

$$(1-\delta)(\tilde{OPT}-F(y(t)))\geq \tilde{OPT}-F(y(t+\delta)) \tag{26}$$

Since $F(y(0))=0$, by induction we get

$$\tilde{OPT}-F(y(k\delta))\leq (1-\delta)^k\tilde{OPT} \tag{27}$$

For $k=\frac{1}{\delta}$

$$\tilde{OPT}-F(y(k\delta))\leq (1-\delta)^{\frac{1}{\delta}}\tilde{OPT}\leq \frac{1}{e}\tilde{OPT} \tag{28}$$

$$F(y(1))\geq (1-2/n)(1-\frac{1}{e})OPT\geq (1-\frac{1}{e}-o(1))OPT \tag{29}$$

Finally, using pipage rounding (Lemma 0) we can convert the y we obtained here into an integer solution corresponding to an indepedent set.

5.1.4 Continuous Greedy Process for submodular maximization

Denote items by j and bidders by i. The algorithm is as follows:

1. Let $\delta=\frac{1}{(mn)^2}$. Start at $t=0$ and $y_{ij}=0\ \forall i,j$

2. Let $R_i(t)$ be a random set containing each item j independently with probability $y_j(t)$. For all i,j, estimate the expected marginal profit of player i from item j:

$$\omega_{ij}(t)=E[\omega_i(R_i(t)+j)-\omega_i(R_i(t))] \tag{30}$$

   by taking the average of $(mn)^5$ independent samples.

3. For each j, let $i_j(t) = \arg\max_i \omega_{ij}(t)$ be the preferred player for item j. Set $y_{ij}(t+\delta) = y_{ij}(t) + \delta$ for the preferred player $i = i_j(t)$ and $y_{ij}(t+\delta) = y_{ij}$ otherwise.

4. Increment $t := t + \delta$, if $t < 1$ go back to Step 2.

5. Allocate each item independently with probability $y_{ij}(t)$ to each player i.

By lemma 2, the continuous greedy algorithm gives a $(1 - 1/e - o(1))$ approximation in expectation for the submodular welfare problem in the value oracle model.

5.2 Submodular Welfare Problems for the Demand Oracle model

Dobzinksi and Schapira proved that one can achieve a $\frac{e}{e-1}$ approximation for submodular bidders with a demand oracle. We first define XOS valuations [3]. A valuation is called additive if $\forall S \subseteq M$, $c(S) = \sum_{j \in S} c(\{j\})$. An additive valuation is completely defined by the values assigned to the items. We can thus represent additive valuations in the following clause form:

$$(x_1 : c_1 \vee x_2 : c_2 \vee \cdots \vee x_m : c_m) \tag{31}$$

XOS valuation: A valuation is c said to be XOS if there is a set of additive valuations $\{a_1, a_2 ... a_t\}$ such that $c(S) = \max_k\{a_k(S)\}$ for all $S \subseteq M$. XOS valuations can be denoted by

$$(x_1 : a_1(\{1\}) \vee x_2 : a_1(\{2\}) \vee \cdots \vee x_m : a_1(\{m\}) \oplus \cdots \oplus (x_1 : a_t(\{1\}) \vee x_2 : a_t(\{2\}) \vee \cdots \vee x_m : a_t(\{m\}) \tag{32}$$

Note that the class of XOS valuations strictly contains the class of submodular valuations. The clause a_i which gives the maximum value $c(S) = \max_k\{a_k(S)\}$ is called the maximizing clause. An XOS oracle is an oracle that when given a bundle S can return the maximizing clause for S for a given valuation c. In the context of auctions, each bidder has a set of valuations.

The algorithm first solves the linear relaxation of the problem and then performs randomized rounding to allocate sets. The input to the algorithm are the n valuations v_i, for each of which we are given a demand oracle and an XOS oracle. The output allocation $T_1, T_2 ... T_n$ is a $\frac{1}{1-(1-\frac{1}{n})^n} < \frac{e}{e-1}$ approximation to the optimal allocation. The algorithm is as follows:

1. First solve the linear relaxation of the problem:

$$\begin{aligned}
\max \sum_{i,S} x_{i,S} c_i(S) \\
\text{s.t.} \sum_{i,S|j \in S} x_{i,S} \leq 1 \ \forall j \in M \\
\sum_{S} x_{i,S} \leq 1 \ \forall i \\
S : x_{i,S} \geq 0 \ \forall i
\end{aligned}$$

The first constraint ensures each item appears only once in each allocation for a bidder. The second constraint ensures that for each bidder only one set of items is allocated. The third constraint ensures that the allocations are non-negative.
The linear program has exponentially many variables, but can be solved in polynomial time. To see this, we construct the dual of this program. Define the dual variable $y \in R^{N+1}$. Let $x_{i,S}$ be a flattened row vector of dimension $(N.2^{|M|}) \times 1$. Define

$$A : R^{N\times(N.2^{|M|})} | a_{ij} = \begin{cases} 1 & \text{if } \; i \in S_j \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

By this definition, we are implicitly allocating a copy of each set to each player, and if $a_{ij} = 1$ for $j = kN$, a_{il} will also be equal to 1 for $j \le l \le j + N$. We can now write the dual as

$$\begin{aligned} \min \;\; & 1^T_{(N+1)\times 1} y \\ \text{s.t.} \;\; & y \ge 0 \\ & [A^T 1_{(N.2^{|M|})\times 1}] y \ge c \end{aligned}$$

This problem can be solved using the ellipsoid method. The ellipsoid algorithm requires a separation oracle, and it has been shown that a separation oracle can be constructed from the demand oracle.

2. Use randomized rounding to find an initial allocation $S_1, S_2 \ldots S_n$.
   For each bidder i, independently choose a set S_i with probability $x_{i,S}$, and choose the empty set with probability $1 - \sum_S x_{i,S}$

3. Let $(x_1 : p_1^i \vee \ldots x_m : p_m^i)$ be the maximizing clause for S_i in c_i. When the valuations are submodular, a demand oracle can simulate an XOS oracle [2] and therefore the maximizing clause can be found for each S_i.

4. Allocate the jth item to bidder i for which $p_j^i \ge p_j^{i'} \;\; \forall i' \in N$.

Proof of Optimality: The allocation is clearly feasible. Let the maximizing clause for S in c_i be $(x_1 : p_1^i \vee \ldots x_m : p_m^i)$ for every bidder i and every set of items S. Then

$$OPT = \sum_{i,S} x_{i,S} v_i(S) = \sum_{i,S} x_{i,S} \sum_j p_j^{(i,S)} = \sum_j (\sum_{i,S} x_{i,S} p_j^{(i,S)}) \tag{34}$$

Let Q_j be the random variable $\max_{i\in N}\{p_j^i\}$ that we obtain after randomized rounding. Let ALG be the random variable $\sum_i c_i(T_i)$ after the algorithm is finished. $ALG \ge \sum_j Q_j$ because we are dealing with an XOS problem.
Lemma 4 : For every item j,

$$E[Q_j] \ge (1 - (1 - \frac{1}{n})^n)(\sum_{i,S} x_{i,S} p_j^{(i,S)}) \tag{35}$$

Proof: Consider an item j. Denote X_i^j as the probability that bidder i gets item j in the initial allocation. We have

$$X_i^j = \sum_{S|j \in S} x_{i,S} \tag{36}$$

because the probability that S is assigned to bidder i is $x_{i,S}$. Denote V_i^j as the expected value of j for bidder i, conditioned on bidder i being allocated item j in the initial allocation.

$$V_i^j = \frac{\sum_{S|j \in S} x_{i,S} p_j^{(i,S)}}{X_i^j} \tag{37}$$

Order the bidders in decreasing order of V_i^j, and without loss of generality assume that the ordering is $1, 2 \dots n$. Assign item j to the bidder who has the highest expected value of j (i.e., the highest bidder in the order of V_i^j for a particular j.) Denote the expected value of j in this allocation as $E[T_j]$. Note that $E[Q_j] \geq E[T_j]$, because while for Q_j we are taking the expectation over the max of the valuations, for T_j we are taking the max over the expectation of the valuations. So to lower bound $E[Q_j]$, we will find a lower bound for $E[T_j]$. Writing out the expression for $E[T_j]$,

$$E[T_j] = X_1^j V_1^j + (1 - X_1^j) X_2^j V_2^j + \dots + (1 - X_1^j)(1 - X_2^j) \dots (1 - X_{n-1}^j) X_n^j V_n^j \tag{38}$$

The first constraint of the LP dictates that $\sum_i X_i^j \leq 1$. Therefore for $1 \leq k \leq n$, we have

$$\begin{aligned} 1 - (1 - X_1^j)(1 - X_2^j) \dots (1 - X_k^j) &\geq (1 - \frac{\sum_{i=1}^k X_i^j}{k})^k \\ &\geq (1 - \frac{1}{k})^k \sum_{i=1}^k X_i^j \\ &\geq (1 - \frac{1}{n})^n \sum_{i=1}^k X_i^j \end{aligned}$$

The last two inequalities are results from calculus. Define $V_{n+1}^j = 0$ and multiply both sides of the above inequality with the quantity $(V_k^j - V_{k+1}^j)$ (which we know is positive because of the ranking) and sum both sides over all k. On the left hand side, we get

$$\begin{aligned} &\sum_k 1 - (1 - X_1^j) \dots (1 - X_k^j)(V_k^j - V_{k+1}^j) \\ &= X_1^j V_1^j + \sum_{k=2}^n X_k^j [(1 - X_1^j) \dots (1 - X_{k-1}^j)] V_k^j \\ &= E[T_j] \end{aligned}$$

On the right hand side,

$$\sum_k (1 - \frac{1}{n})^n \sum_{i=1}^k X_i^j (V_k^j - V_{k+1}^j) = \sum_k (1 - \frac{1}{n})^n V_k^j X_k^j$$

Therefore

$$\begin{aligned} E[T_j] &\geq (1-\frac{1}{n})^n \sum_i V_i^j X_i^j \\ &= (1-\frac{1}{n})^n \sum_{S|j\in S} x_{i,S} p_j^{(i,S)} \end{aligned}$$

By linearity of expectation

$$\begin{aligned} E[ALG] &\geq \sum_j E[Q_j] \\ &\geq \sum_j (1-(1-\frac{1}{n})^n) \sum_{S|j\in S} x_{i,S} p_j^{(i,S)} \\ &= (1-(1-\frac{1}{n})^n) OPT \end{aligned}$$

QED.

6 Conclusion

In this project, a number of techniques to solve combinatorial auction problems have been reviewed. We started with exact methods, then described special cases of the constraints that allow exact polynomial solutions, also including where possible cases in real-life applications where such special cases arise, and then described some approximate methods. For the approximate algorithms, we studied the special case of submodular welfare, and showed two algorithms that achieve near optimal performance for two different oracle models.

References

[1] Sven De Vries and Rakesh V Vohra. "Combinatorial auctions: A survey". In: *INFORMS Journal on computing* 15.3 (2003), pp. 284–309.

[2] Shahar Dobzinski, Noam Nisan, and Michael Schapira. "Approximation algorithms for combinatorial auctions with complement-free bidders". In: *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing.* 2005, pp. 610–618.

[3] Shahar Dobzinski and Michael Schapira. "An improved approximation algorithm for combinatorial auctions with submodular bidders". In: *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm.* 2006, pp. 1064–1073.

[4] Jan Vondrák. "Optimal approximation for the submodular welfare problem in the value oracle model". In: *Proceedings of the fortieth annual ACM symposium on Theory of computing.* 2008, pp. 67–74.