SC 631 course project presentation : A survey of Combinatorial Auctions

Sahasrajit Sarmasarkar(Roll no 160070100)
Devanshu Singh Gharawar(Roll no 16D070010)

Auctions: an Intro

- What is an auction? An auction is a game where a seller called auctioneer wishes to distribute objects to potential buyers called bidders based on some information communicated by the bidders to the auctioneer.

Auctions: an Intro

- What is an auction? An auction is a game where a seller called auctioneer wishes to distribute objects to potential buyers called bidders based on some information communicated by the bidders to the auctioneer.
- Where has it been used? Used in a variety of scenarios like FCC spectrum auctions, auctions for airport time slots, railroad segments, delivery routes and network routing.

Auctions: an Intro

- What is an auction? An auction is a game where a seller called auctioneer wishes to distribute objects to potential buyers called bidders based on some information communicated by the bidders to the auctioneer.
- Where has it been used? Used in a variety of scenarios like FCC spectrum auctions, auctions for airport time slots, railroad segments, delivery routes and network routing.

What is this information communicated and how should it be conveyed? The most obvious thing would be to ask for bids for every object from each bidder and allot each object to the highest bidder? Can we do something better? ..Think..

Potential issues/ alternate auction strategies

- What if some bidders value each object individually lowly but have utility for pairs of objects? For example, a dining table has minimal utility in absence of chairs. In this case, allowing bids on "bundles" or subsets of items may be better.

Potential issues/ alternate auction strategies

- What if some bidders value each object individually lowly but have utility for pairs of objects? For example, a dining table has minimal utility in absence of chairs. In this case, allowing bids on "bundles" or subsets of items may be better.
- Should the auction be held in multiple rounds i.e allowing bidders to change their bids to stay in the game for subsequent rounds? This may be useful in the perspective of the auctioneer as in each subsequent round the bid is only supposed to increase to stay in the game.

Potential issues/ alternate auction strategies

- What if some bidders value each object individually lowly but have utility for pairs of objects? For example, a dining table has minimal utility in absence of chairs. In this case, allowing bids on "bundles" or subsets of items may be better.
- Should the auction be held in multiple rounds i.e allowing bidders to change their bids to stay in the game for subsequent rounds? This may be useful in the perspective of the auctioneer as in each subsequent round the bid is only supposed to increase to stay in the game.
- If at all the auction takes place in multiple rounds, what information should pass from one round to the next? This may have significance in order to avoid collusion between the players.

Potential issues/ alternate auction strategies

- What if some bidders value each object individually lowly but have utility for pairs of objects? For example, a dining table has minimal utility in absence of chairs. In this case, allowing bids on "bundles" or subsets of items may be better.
- Should the auction be held in multiple rounds i.e allowing bidders to change their bids to stay in the game for subsequent rounds? This may be useful in the perspective of the auctioneer as in each subsequent round the bid is only supposed to increase to stay in the game.
- If at all the auction takes place in multiple rounds, what information should pass from one round to the next? This may have significance in order to avoid collusion between the players.

However, which policy is optimal may depend on the objective function the auctioneer is willing to optimise. Is he caring about "self" or "overall" good?

Objective functions for the auctioneer

- Economically efficient- Under this objective function, the auctioneer chooses the strategy to maximise the overall utility of all the bidders.

Objective functions for the auctioneer

- Economically efficient- Under this objective function, the auctioneer chooses the strategy to maximise the overall utility of all the bidders.
- Maximise revenue- Under this objective function, the auctioneer chooses a strategy to maximise his overall revenue.

Objective functions for the auctioneer

- Economically efficient- Under this objective function, the auctioneer chooses the strategy to maximise the overall utility of all the bidders.
- Maximise revenue- Under this objective function, the auctioneer chooses a strategy to maximise his overall revenue.

Clearly, a different objective function may change the strategy from one to another. There also may be other functions to be taken care of like computation cost for computing which bidder gets what, communication cost between the bidders and auctioneer.

Objective functions for the auctioneer

- Economically efficient- Under this objective function, the auctioneer chooses the strategy to maximise the overall utility of all the bidders.
- Maximise revenue- Under this objective function, the auctioneer chooses a strategy to maximise his overall revenue.

Clearly, a different objective function may change the strategy from one to another. There also may be other functions to be taken care of like computation cost for computing which bidder gets what, communication cost between the bidders and auctioneer.
For example, if each bidder bids on a subset of objects, it may suffer from the following.

- High communication cost- This is because each bidder needs to transmit bids on all possible subsets.
- High computation cost- This may occur here because the algorithm may have large run time as the problem may be NP-hard (similar to set packing problem).

Framework and setting of combinatorial auction problem(CAP1)

We currently formulate the single round combinatorial auction problem as an optimisation problem and we solve the problem to maximise the revenue of the auctioneer.

Framework and setting of combinatorial auction problem(CAP1)

We currently formulate the single round combinatorial auction problem as an optimisation problem and we solve the problem to maximise the revenue of the auctioneer. Notations:

- Let N be the set of bidders and M the set of m distinct objects. Let $b_j(S)$ denote the bid by a bidder $j \in N$ for every subset $S \subseteq M$.
- Clearly bids with $b_j(S) \leq 0$ would never get selected hence we assume $b_j(S) \geq 0$ without loss of generality.
- Let $y(S, j)$ be 1 if subset S is allotted to player j, 0 otherwise.

Framework and setting of combinatorial auction problem(CAP1)

We currently formulate the single round combinatorial auction problem as an optimisation problem and we solve the problem to maximise the revenue of the auctioneer. Notations:

- Let N be the set of bidders and M the set of m distinct objects. Let $b_j(S)$ denote the bid by a bidder $j \in N$ for every subset $S \subseteq M$.
- Clearly bids with $b_j(S) \leq 0$ would never get selected hence we assume $b_j(S) \geq 0$ without loss of generality.
- Let $y(S, j)$ be 1 if subset S is allotted to player j, 0 otherwise.

The optimisation problem can be formulated as follows:

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

$$\sum_{S \ni i} \sum_{j \in N} y(S, j) \leq \forall i \in M \; ; \; y(S, j) = 0, 1 \forall S \subseteq M, j \in N$$

Modification of CAP formulation under superadditivity(CAP2)

- Let us assume that each utility function for every bidder is superadditive i.e. $b_i(S_1) + b_i(S_2) \leq b_i(S_1 \cup S_2)\ \forall i \in N$; $\forall S_1, S_2 \in M; S_1 \cap S_2 = \phi$

Modification of CAP formulation under superadditivity(CAP2)

- Let us assume that each utility function for every bidder is superadditive i.e. $b_i(S_1) + b_i(S_2) \leq b_i(S_1 \cup S_2)\ \forall i \in N$; $\forall S_1, S_2 \in M; S_1 \cap S_2 = \phi$
- Under this setting, a more simpler formulation is possible. Let us denote $b(S) = \max_j b_j(S)$ and $x_S = 1$ if the highest bid on set S is accepted 0 otherwise.

Modification of CAP formulation under superadditivity(CAP2)

- Let us assume that each utility function for every bidder is superadditive i.e. $b_i(S_1) + b_i(S_2) \leq b_i(S_1 \cup S_2)\ \forall i \in N$; $\forall S_1, S_2 \in M; S_1 \cap S_2 = \phi$
- Under this setting, a more simpler formulation is possible. Let us denote $b(S) = \max_j b_j(S)$ and $x_S = 1$ if the highest bid on set S is accepted 0 otherwise.

$$\max \sum_{S \subseteq M} b(S) x_S \text{ s.t}$$

$$\sum_{S \ni i} x_S \leq 1 \forall i \in M \ ; \ x_S = 0, 1 \forall S \subseteq M$$

Modification of CAP formulation under superadditivity(CAP2)

- Let us assume that each utility function for every bidder is superadditive i.e. $b_i(S_1) + b_i(S_2) \leq b_i(S_1 \cup S_2)\ \forall i \in N$; $\forall S_1, S_2 \in M; S_1 \cap S_2 = \phi$
- Under this setting, a more simpler formulation is possible. Let us denote $b(S) = \max_j b_j(S)$ and $x_S = 1$ if the highest bid on set S is accepted 0 otherwise.

$$\max \sum_{S \subseteq M} b(S) x_S \text{ s.t}$$

$$\sum_{S \ni i} x_S \leq 1 \forall i \in M \ ; \ x_S = 0, 1 \forall S \subseteq M$$

The above formulation follows from the result that under this super additive setting if some subset is alloted to some bidder it must be the bidder with the largest bid for this subset.

Connection to SPP (Set Packing Problem) problem

Problem formulation: Given a ground set M of objects and a collection V of subsets of non-negative weights, find the largest weight collection of subsets that are pairwise independent.

Connection to SPP (Set Packing Problem) problem

Problem formulation: Given a ground set M of objects and a collection V of subsets of non-negative weights, find the largest weight collection of subsets that are pairwise independent.
Notations: We denote $x_j = 1$ if the j^{th} set in V with weight c_j is selected and $x_j = 0$ otherwise. Define a_{ij} to be 1 if the j^{th} set in V contains element $i \in M$

Connection to SPP (Set Packing Problem) problem

Problem formulation: Given a ground set M of objects and a collection V of subsets of non-negative weights, find the largest weight collection of subsets that are pairwise independent.

Notations: We denote $x_j = 1$ if the j^{th} set in V with weight c_j is selected and $x_j = 0$ otherwise. Define a_{ij} to be 1 if the j^{th} set in V contains element $i \in M$

$$\max \sum_{j \in V} c_j . x_j \text{ s.t}$$

$$\sum_{j \in V} a_{ij} x_j \leq 1 \forall i \in M \text{ ; } x_j = 0, 1 \forall j \in V$$

Connection to SPP (Set Packing Problem) problem

Problem formulation: Given a ground set M of objects and a collection V of subsets of non-negative weights, find the largest weight collection of subsets that are pairwise independent.
Notations: We denote $x_j = 1$ if the j^{th} set in V with weight c_j is selected and $x_j = 0$ otherwise. Define a_{ij} to be 1 if the j^{th} set in V contains element $i \in M$

$$\max \sum_{j \in V} c_j . x_j \text{ s.t}$$

$$\sum_{j \in V} a_{ij} x_j \leq 1 \forall i \in M \text{ ; } x_j = 0, 1 \forall j \in V$$

- Note that the formulation under CAP 2 directly matches with the set packing problem.

Connection to SPP (Set Packing Problem) problem

Problem formulation: Given a ground set M of objects and a collection V of subsets of non-negative weights, find the largest weight collection of subsets that are pairwise independent.

Notations: We denote $x_j = 1$ if the j^{th} set in V with weight c_j is selected and $x_j = 0$ otherwise. Define a_{ij} to be 1 if the j^{th} set in V contains element $i \in M$

$$\max \sum_{j \in V} c_j . x_j \text{ s.t}$$

$$\sum_{j \in V} a_{ij} x_j \leq 1 \forall i \in M \; ; \; x_j = 0, 1 \forall j \in V$$

- Note that the formulation under CAP 2 directly matches with the set packing problem.
- If the values the users bid are actually the true utilities for the bidders then the allocation under this problem would be economically efficient.

Complexity of SPP problem

How hard is the SPP problem? In general, it can be shown that the SPP problem is NP-hard. One trivial solution might be to check at all $2^{|V|}$ nodes but it might be computationally very expensive for even moderate $|V|$.

Complexity of SPP problem

How hard is the SPP problem? In general, it can be shown that the SPP problem is NP-hard. One trivial solution might be to check at all $2^{|V|}$ nodes but it might be computationally very expensive for even moderate $|V|$.

Under the general scenario, the input size itself might be exponential in $|M|$, thus even if the computation is exponential in $|M|$, it would be polynomial in the input size.

Complexity of SPP problem

How hard is the SPP problem? In general, it can be shown that the SPP problem is NP-hard. One trivial solution might be to check at all $2^{|V|}$ nodes but it might be computationally very expensive for even moderate $|V|$.

Under the general scenario, the input size itself might be exponential in $|M|$, thus even if the computation is exponential in $|M|$, it would be polynomial in the input size.

Thus for a reasonable algorithm, we need to ensure that the number of distinct bids is not too large and the corresponding SPP problem can be solved in polynomial time.

Solvable instances of SPP problem

Integral polyhedron: One most common scenario where there exists a polynomial algorithm is when the extreme points of the polyhedron $P(A) = \{x : j \in V; a_{ij}x_j \leq 1 \forall i \in M; x_j \geq 0 \forall j \in V\}$ are all integral, i.e. 0-1. In these cases, we can simply solve it as a linear programming in polynomial time.
Let us now consider the various scenario under which this maybe satisfied.

Solvable instances of SPP problem

Integral polyhedron: One most common scenario where there exists a polynomial algorithm is when the extreme points of the polyhedron $P(A) = \{x : j \in V; a_{ij}x_j \leq 1 \forall i \in M; x_j \geq 0 \forall j \in V\}$ are all integral, i.e. 0-1. In these cases, we can simply solve it as a linear programming in polynomial time.

Let us now consider the various scenario under which this maybe satisfied.

- Total unimoudularity(TU): A matrix A is said to be TU if the determinant of every square submatrix is $0, 1$ or -1. If the matrix $A = \{a_{ij}\}_{\{i \in M, j \in V\}}$ is TU then all extreme points of the polyhedron $P(A)$ are integral.

Solvable instances of SPP problem

Integral polyhedron: One most common scenario where there exists a polynomial algorithm is when the extreme points of the polyhedron $P(A) = \{x : j \in V; a_{ij}x_j \leq 1 \forall i \in M; x_j \geq 0 \forall j \in V\}$ are all integral, i.e. 0-1. In these cases, we can simply solve it as a linear programming in polynomial time.

Let us now consider the various scenario under which this maybe satisfied.

- Total unimoudularity(TU): A matrix A is said to be TU if the determinant of every square submatrix is $0, 1$ or -1. If the matrix $A = \{a_{ij}\}_{\{i \in M, j \in V\}}$ is TU then all extreme points of the polyhedron $P(A)$ are integral.
  A special case of TU matrices are those with the consecutive-ones property. A 0-1 matrix has the consecutive-ones property if the non-zero entries in each column occur consecutively.

Intuition under TU modularity

Under this scenario, let us model the objects as parcels of land along a shore line. The shore line would impose a linear order on the parcels.We can argue in this scenario that the most interesting bids for players in thhis scenario is contigious.

- The number of bids would clearly be limited to the number of intervals a polynomial size in number of objects.

Intuition under TU modularity

Under this scenario, let us model the objects as parcels of land along a shore line. The shore line would impose a linear order on the parcels.We can argue in this scenario that the most interesting bids for players in thhis scenario is contigious.

This leads to few important consequences.

- The number of bids would clearly be limited to the number of intervals a polynomial size in number of objects. Also since this matrix A of corresponding CAP2 has the consecutive ones property, the polyhedron $P(A)$ would be integral implying the optimisation problem can be solved in polynomial time in M.

Intuition under TU modularity

Under this scenario, let us model the objects as parcels of land along a shore line. The shore line would impose a linear order on the parcels.We can argue in this scenario that the most interesting bids for players in thhis scenario is contigious.

This leads to few important consequences.

- The number of bids would clearly be limited to the number of intervals a polynomial size in number of objects. Also since this matrix A of corresponding CAP2 has the consecutive ones property, the polyhedron $P(A)$ would be integral implying the optimisation problem can be solved in polynomial time in M.
- Thus if the valuation of the bidders in consecutive intervals satisfy the superadditive property, we can say that CAP2 formulation models it correctly and solvable in polynomial time.

Intuition under TU modularity

Under this scenario, let us model the objects as parcels of land along a shore line. The shore line would impose a linear order on the parcels.We can argue in this scenario that the most interesting bids for players in thhis scenario is contigious.

This leads to few important consequences.

- The number of bids would clearly be limited to the number of intervals a polynomial size in number of objects. Also since this matrix A of corresponding CAP2 has the consecutive ones property, the polyhedron $P(A)$ would be integral implying the optimisation problem can be solved in polynomial time in M.
- Thus if the valuation of the bidders in consecutive intervals satisfy the superadditive property, we can say that CAP2 formulation models it correctly and solvable in polynomial time.
- However if the supperadditive property does not hold we may have to use CAP1 with additional constraints which may violate the consecutive ones property.

Balanced matrices

A 0-1 matrix B is balanced if it has no square submatrix of odd order with exactly two 1s in each row and column. If the matrix B is balanced then the linear program max $\{\sum c_j x_j \sum_j b_{ij} x_j \leq 1 \forall i; x_j \geq 0 \forall j\}$ has an integral optimal solution whenever the c_j are integral

Balanced matrices

A 0-1 matrix B is balanced if it has no square submatrix of odd order with exactly two 1s in each row and column. If the matrix B is balanced then the linear program max $\{\sum c_j x_j \sum_j b_{ij} x_j \leq 1 \forall i; x_j \geq 0 \forall j\}$ has an integral optimal solution whenever the c_j are integral An example relevant to this setting

Balanced matrices

A 0-1 matrix B is balanced if it has no square submatrix of odd order with exactly two 1s in each row and column. If the matrix B is balanced then the linear program max $\{\sum c_j x_j \sum_j b_{ij} x_j \leq 1 \forall i; x_j \geq 0 \forall j\}$ has an integral optimal solution whenever the c_j are integral An example relevant to this setting

- For each vertex $v \in T$ let $N(v, r)$ denote the set of all vertices in T that are within distance r of v. We can the vertices as parcels of land connected by a road network with no cycles. Bidders can bid for subsets of parcels but the subsets are con- strained to be of the form $N(v, r)$ for some vertex v and some number r.

Balanced matrices

A 0-1 matrix B is balanced if it has no square submatrix of odd order with exactly two 1s in each row and column. If the matrix B is balanced then the linear program max $\{\sum c_j x_j \sum_j b_{ij} x_j \leq 1 \forall i; x_j \geq 0 \forall j\}$ has an integral optimal solution whenever the c_j are integral   An example relevant to this setting

- For each vertex $v \in T$ let $N(v, r)$ denote the set of all vertices in T that are within distance r of v. We can the vertices as parcels of land connected by a road network with no cycles. Bidders can bid for subsets of parcels but the subsets are con- strained to be of the form $N(v, r)$ for some vertex v and some number r.
- Now the constraint matrix of the corresponding SPP will have one column for each set of the form $N(v, r)$ and one row for each vertex of T . This constraint matrix is balanced

Restricting preferences by allowing just two types of bidders

In this setting,we restrict the preference of the bidders themselves instead of restricting the sets of objects over which preferences can be expressed.

Restricting preferences by allowing just two types of bidders

In this setting,we restrict the preference of the bidders themselves instead of restricting the sets of objects over which preferences can be expressed. One common restriction placed on the preference on the bidder functions is that it is non-decreasing (i.e. $b_j(S) \leq b_j(T)\ \forall S \subseteq T$)and and supermodular (i.e. $b_j(S) + b_j(T) \leq b_j(S \cap T) + b_j(S \cup T)$)

Restricting preferences by allowing just two types of bidders

In this setting,we restrict the preference of the bidders themselves instead of restricting the sets of objects over which preferences can be expressed. One common restriction placed on the preference on the bidder functions is that it is non-decreasing (i.e. $b_j(S) \leq b_j(T)\ \forall S \subseteq T$)and and supermodular (i.e. $b_j(S) + b_j(T) \leq b_j(S \cap T) + b_j(S \cup T)$)
Let us now suppose that the bidder functions come in two types i.e. $b_j(.) = g^1(.)$ or $g^2(.)$ where $g^1(.)$ and $g^2(.)$ are non-decreasing super modular functions. Let N^r denote the set of type r bidders.

Restricting preferences by allowing just two types of bidders

In this setting,we restrict the preference of the bidders themselves instead of restricting the sets of objects over which preferences can be expressed. One common restriction placed on the preference on the bidder functions is that it is non-decreasing (i.e. $b_j(S) \leq b_j(T)\ \forall S \subseteq T$)and and supermodular (i.e. $b_j(S) + b_j(T) \leq b_j(S \cap T) + b_j(S \cup T)$)
Let us now suppose that the bidder functions come in two types i.e. $b_j(.) = g^1(.)$ or $g^2(.)$ where $g^1(.)$ and $g^2(.)$ are non-decreasing super modular functions. Let N^r denote the set of type r bidders.
Let us consider the dual of linear programming relaxation under CAP1:

$$\min \sum_{i \in M} p_i + \sum_{j \in N} q_j \text{ s.t}$$

$$\sum_{i \in S} p_i + q_j \geq g^1(S) \forall S \subseteq M; j \in N^1$$

$$\sum_{i \in S} p_i + q_j \geq g^2(S) \forall S \subseteq M; j \in N^2$$

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

Some properties of the above dual formation

It has the property of being totally dual integral, which means that its linear-programming dual, the linear relaxation of the original primal problem, has an integer optimal solution. This formulation has been used in variety of settings

Some properties of the above dual formation

It has the property of being totally dual integral, which means that its linear-programming dual, the linear relaxation of the original primal problem, has an integer optimal solution. This formulation has been used in variety of settings
However, this idea cannot be extended to more than to types of bidders as it has been proved that the dual problem in this case admits fractional integer points as solutions.

Gross substitute property

Another way to restrict preferences could be using gross-substitute property. To describe it let the value that bidder j assigns to the set $S \subseteq M$ of objects be $v_j(S)$.

Gross substitute property

Another way to restrict preferences could be using gross-substitute property. To describe it let the value that bidder j assigns to the set $S \subseteq M$ of objects be $v_j(S)$.
Given a vector of prices p, let the collection of subsets that maximise utility of j be denoted by $D_j(p)$ and defined as:
$D_j(p) = \{S \subseteq M : v_j(S) - \sum_{i \in S} p_i \geq v_j(T) - \sum_{i \in T} p_i \forall T \subseteq M\}$

Gross substitute property

Another way to restrict preferences could be using gross-substitute property. To describe it let the value that bidder j assigns to the set $S \subseteq M$ of objects be $v_j(S)$.
Given a vector of prices p, let the collection of subsets that maximise utility of j be denoted by $D_j(p)$ and defined as:
$D_j(p) = \{S \subseteq M : v_j(S) - \sum_{i \in S} p_i \geq v_j(T) - \sum_{i \in T} p_i \forall T \subseteq M\}$
The gross substitute property requires that for all price vectors p, p' such that $p' \geq p$ and all $A \in D_j(p)$, there exists $B \in D_j(p)$ such that $\{i \in A : p_i = p'_i\} \subseteq B$

Gross substitute property

Another way to restrict preferences could be using gross-substitute property. To describe it let the value that bidder j assigns to the set $S \subseteq M$ of objects be $v_j(S)$.

Given a vector of prices p, let the collection of subsets that maximise utility of j be denoted by $D_j(p)$ and defined as:

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

The gross substitute property requires that for all price vectors p, p' such that $p' \geq p$ and all $A \in D_j(p)$, there exists $B \in D_j(p)$ such that $\{i \in A : p_i = p'_i\} \subseteq B$

An important property when each of the $b^j(.)$ has the gross-substitute property the linear-programming relaxation of CAP1 and CAP2 have an optimal integer solution.

References

Thank You


THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
GOZAIMASHITA
EFCHARISTO
JUSPAXAR
DANKSCHEEN
TASHAKKUR ATU
YAQHANYELAY
SUKSAMA
EKHMET
MEHRBANI
GRAZIE
PALDIES
KOMAPSUMNIDA
MAAKE
BOLZİN
MERCI
BİYAN
SHUKRIA
TINGKI