Single-Minded Case in Combinatorial Auctions

Akhoury Shauryam

Algorithmic Game Theory

Combinatorial Auctions: Introduction

- **Definition:** Auctions where multiple items are auctioned simultaneously, and bidders express preferences over bundles of items.

Combinatorial Auctions: Introduction

- **Definition:** Auctions where multiple items are auctioned simultaneously, and bidders express preferences over bundles of items.
- **Motivation:** Allows efficient allocation of resources with interdependencies.

Combinatorial Auctions: Introduction

- **Definition:** Auctions where multiple items are auctioned simultaneously, and bidders express preferences over bundles of items.
- **Motivation:** Allows efficient allocation of resources with interdependencies.
- **Applications:**

Combinatorial Auctions: Introduction

- **Definition:** Auctions where multiple items are auctioned simultaneously, and bidders express preferences over bundles of items.
- **Motivation:** Allows efficient allocation of resources with interdependencies.
- **Applications:**
  - Spectrum Auctions

Combinatorial Auctions: Introduction

- **Definition:** Auctions where multiple items are auctioned simultaneously, and bidders express preferences over bundles of items.
- **Motivation:** Allows efficient allocation of resources with interdependencies.
- **Applications:**
  - Spectrum Auctions
  - Logistics and Transportation

Combinatorial Auctions: Introduction

- **Definition:** Auctions where multiple items are auctioned simultaneously, and bidders express preferences over bundles of items.
- **Motivation:** Allows efficient allocation of resources with interdependencies.
- **Applications:**
  - Spectrum Auctions
  - Logistics and Transportation
  - Communication Networks

Combinatorial Auctions: Introduction

- **Definition:** Auctions where multiple items are auctioned simultaneously, and bidders express preferences over bundles of items.
- **Motivation:** Allows efficient allocation of resources with interdependencies.
- **Applications:**
  - Spectrum Auctions
  - Logistics and Transportation
  - Communication Networks
- **Challenges:**

Combinatorial Auctions: Introduction

- **Definition:** Auctions where multiple items are auctioned simultaneously, and bidders express preferences over bundles of items.
- **Motivation:** Allows efficient allocation of resources with interdependencies.
- **Applications:**
  - Spectrum Auctions
  - Logistics and Transportation
  - Communication Networks
- **Challenges:**
  - Computational Complexity: Winner determination is NP-hard.

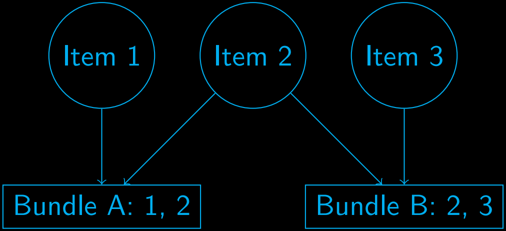
Combinatorial Auctions: Introduction

- **Definition:** Auctions where multiple items are auctioned simultaneously, and bidders express preferences over bundles of items.
- **Motivation:** Allows efficient allocation of resources with interdependencies.
- **Applications:**
  - Spectrum Auctions
  - Logistics and Transportation
  - Communication Networks
- **Challenges:**
  - Computational Complexity: Winner determination is NP-hard.
  - Communication Complexity: Preferences over bundles are large.

Combinatorial Auctions: Introduction

- **Definition:** Auctions where multiple items are auctioned simultaneously, and bidders express preferences over bundles of items.
- **Motivation:** Allows efficient allocation of resources with interdependencies.
- **Applications:**
  - Spectrum Auctions
  - Logistics and Transportation
  - Communication Networks
- **Challenges:**
  - Computational Complexity: Winner determination is NP-hard.
  - Communication Complexity: Preferences over bundles are large.
  - Strategic Behavior: Mechanisms must incentivize truthfulness.

Diagram



Combinatorial Auctions: Mathematical Definition

Formal Setup:

Combinatorial Auctions: Mathematical Definition

Formal Setup:

- m: Number of items, $\{1, 2, \ldots, m\}$.

Combinatorial Auctions: Mathematical Definition

Formal Setup:

- m: Number of items, $\{1, 2, \ldots, m\}$.
- n: Number of bidders, $\{1, 2, \ldots, n\}$.

Combinatorial Auctions: Mathematical Definition

Formal Setup:

- m: Number of items, $\{1, 2, \ldots, m\}$.
- n: Number of bidders, $\{1, 2, \ldots, n\}$.
- Each bidder i has a valuation function $v_i : 2^{\{1,2,\ldots,m\}} \to \mathbb{R}_{\geq 0}$, mapping each subset of items to its value.

Combinatorial Auctions: Mathematical Definition

Formal Setup:

- m: Number of items, $\{1, 2, \ldots, m\}$.
- n: Number of bidders, $\{1, 2, \ldots, n\}$.
- Each bidder i has a valuation function $v_i : 2^{\{1,2,\ldots,m\}} \to \mathbb{R}_{\geq 0}$, mapping each subset of items to its value.

Properties of Valuation Functions:

Combinatorial Auctions: Mathematical Definition

Formal Setup:

- m: Number of items, $\{1, 2, \ldots, m\}$.
- n: Number of bidders, $\{1, 2, \ldots, n\}$.
- Each bidder i has a valuation function $v_i : 2^{\{1,2,\ldots,m\}} \to \mathbb{R}_{\geq 0}$, mapping each subset of items to its value.

Properties of Valuation Functions:

- **Monotonicity:** $v_i(S) \leq v_i(T)$ for $S \subseteq T$.

Combinatorial Auctions: Mathematical Definition

Formal Setup:

- m: Number of items, $\{1, 2, \ldots, m\}$.
- n: Number of bidders, $\{1, 2, \ldots, n\}$.
- Each bidder i has a valuation function $v_i : 2^{\{1,2,\ldots,m\}} \to \mathbb{R}_{\geq 0}$, mapping each subset of items to its value.

Properties of Valuation Functions:

- **Monotonicity:** $v_i(S) \leq v_i(T)$ for $S \subseteq T$.
- **Normalization:** $v_i(\emptyset) = 0$.

Combinatorial Auctions: Mathematical Definition

Formal Setup:

- m: Number of items, $\{1, 2, \ldots, m\}$.
- n: Number of bidders, $\{1, 2, \ldots, n\}$.
- Each bidder i has a valuation function $v_i : 2^{\{1,2,\ldots,m\}} \to \mathbb{R}_{\geq 0}$, mapping each subset of items to its value.

Properties of Valuation Functions:

- **Monotonicity:** $v_i(S) \leq v_i(T)$ for $S \subseteq T$.
- **Normalization:** $v_i(\emptyset) = 0$.

Goal: Find an allocation $S_1, S_2, \ldots, S_n$ of items such that:

Combinatorial Auctions: Mathematical Definition

Formal Setup:

- m: Number of items, $\{1, 2, \ldots, m\}$.
- n: Number of bidders, $\{1, 2, \ldots, n\}$.
- Each bidder i has a valuation function $v_i : 2^{\{1,2,\ldots,m\}} \to \mathbb{R}_{\geq 0}$, mapping each subset of items to its value.

Properties of Valuation Functions:

- **Monotonicity:** $v_i(S) \leq v_i(T)$ for $S \subseteq T$.
- **Normalization:** $v_i(\emptyset) = 0$.

Goal: Find an allocation $S_1, S_2, \ldots, S_n$ of items such that:

$$\max_{S_1, S_2, \ldots, S_n} \sum_{i=1}^{n} v_i(S_i),$$

Combinatorial Auctions: Mathematical Definition

Formal Setup:

- m: Number of items, $\{1, 2, \ldots, m\}$.
- n: Number of bidders, $\{1, 2, \ldots, n\}$.
- Each bidder i has a valuation function $v_i : 2^{\{1,2,\ldots,m\}} \to \mathbb{R}_{\geq 0}$, mapping each subset of items to its value.

Properties of Valuation Functions:

- **Monotonicity:** $v_i(S) \leq v_i(T)$ for $S \subseteq T$.
- **Normalization:** $v_i(\emptyset) = 0$.

Goal: Find an allocation $S_1, S_2, \ldots, S_n$ of items such that:

$$\max_{S_1, S_2, \ldots, S_n} \sum_{i=1}^{n} v_i(S_i),$$

where $S_i \cap S_j = \emptyset$ for $i \neq j$.

Single-Minded Case in Combinatorial Auctions

- **Definition:** Each bidder i is only interested in a specific bundle S_i^* of items.

Single-Minded Case in Combinatorial Auctions

- **Definition:** Each bidder i is only interested in a specific bundle S_i^* of items.
- **Valuation Function:**

$$u_i(S) =$$

Single-Minded Case in Combinatorial Auctions

- **Definition:** Each bidder i is only interested in a specific bundle S_i^* of items.
- **Valuation Function:**

$$u_i(S) = \left\{ \right.$$

Single-Minded Case in Combinatorial Auctions

- **Definition:** Each bidder i is only interested in a specific bundle S_i^* of items.
- **Valuation Function:**

$$u_i(S) = \begin{cases} v_i^* & \text{if } S \supseteq S_i^*, \\ & \end{cases}$$

Single-Minded Case in Combinatorial Auctions

- **Definition:** Each bidder i is only interested in a specific bundle S_i^* of items.
- **Valuation Function:**

$$u_i(S) = \begin{cases} v_i^* & \text{if } S \supseteq S_i^*, \\ 0 & \text{otherwise.} \end{cases}$$

Single-Minded Case in Combinatorial Auctions

- **Definition:** Each bidder i is only interested in a specific bundle S_i^* of items.
- **Valuation Function:**

$$u_i(S) = \begin{cases} v_i^* & \text{if } S \supseteq S_i^*, \\ 0 & \text{otherwise.} \end{cases}$$

- **Key Properties:**

Single-Minded Case in Combinatorial Auctions

- **Definition:** Each bidder i is only interested in a specific bundle S_i^* of items.
- **Valuation Function:**

$$u_i(S) = \begin{cases} v_i^* & \text{if } S \supseteq S_i^*, \\ 0 & \text{otherwise.} \end{cases}$$

- **Key Properties:**
  - Monotonicity: $u_i(S) \leq u_i(T)$ for $S \subseteq T$.

Single-Minded Case in Combinatorial Auctions

- **Definition:** Each bidder i is only interested in a specific bundle S_i^* of items.
- **Valuation Function:**

$$u_i(S) = \begin{cases} v_i^* & \text{if } S \supseteq S_i^*, \\ 0 & \text{otherwise.} \end{cases}$$

- **Key Properties:**
  - Monotonicity: $u_i(S) \leq u_i(T)$ for $S \subseteq T$.
  - Normalization: $u_i(\emptyset) = 0$.

Problem Statement

Problem Statement

Objective: Allocate items to maximize social welfare.

$$\max_{W} \sum_{i \in W} v_i^*$$

Problem Statement

Objective: Allocate items to maximize social welfare.

$$\max_{W} \sum_{i \in W} v_i^*$$

Constraints:

Problem Statement

Objective: Allocate items to maximize social welfare.

$$\max_W \sum_{i \in W} v_i^*$$

Constraints:

- Each bidder receives either their desired bundle $(S_i = S_i^*)$ or nothing $(S_i = \emptyset)$.

Problem Statement

Objective: Allocate items to maximize social welfare.

$$\max_{W} \sum_{i \in W} v_i^*$$

Constraints:

- Each bidder receives either their desired bundle $(S_i = S_i^*)$ or nothing $(S_i = \emptyset)$.
- Bundles are disjoint: $S_i^* \cap S_j^* = \emptyset$ for $i \neq j$.

NP-Hardness Reduction: Independent Set Problem

Independent Set Problem:

NP-Hardness Reduction: Independent Set Problem

Independent Set Problem:

- **Input:** A graph $G = (V, E)$ and a number k.

NP-Hardness Reduction: Independent Set Problem

Independent Set Problem:

- **Input:** A graph $G = (V, E)$ and a number k.
- **Goal:** Find a set $U \subseteq V$ such that:

NP-Hardness Reduction: Independent Set Problem

Independent Set Problem:

- **Input:** A graph $G = (V, E)$ and a number k.
- **Goal:** Find a set $U \subseteq V$ such that:
  - No two vertices in U are connected by an edge.

NP-Hardness Reduction: Independent Set Problem

Independent Set Problem:

- **Input:** A graph $G = (V, E)$ and a number k.
- **Goal:** Find a set $U \subseteq V$ such that:
  - No two vertices in U are connected by an edge.
  - $|U| \geq k$.

NP-Hardness Reduction: Independent Set Problem

Independent Set Problem:

- **Input:** A graph $G = (V, E)$ and a number k.
- **Goal:** Find a set $U \subseteq V$ such that:
  - No two vertices in U are connected by an edge.
  - $|U| \geq k$.

Reduction to Single-Minded Allocation:

NP-Hardness Reduction: Independent Set Problem

Independent Set Problem:

- **Input:** A graph $G = (V, E)$ and a number k.
- **Goal:** Find a set $U \subseteq V$ such that:
  - No two vertices in U are connected by an edge.
  - $|U| \geq k$.

Reduction to Single-Minded Allocation:

- **Items:** E (edges of the graph).

NP-Hardness Reduction: Independent Set Problem

Independent Set Problem:

- **Input:** A graph $G = (V, E)$ and a number k.
- **Goal:** Find a set $U \subseteq V$ such that:
  - No two vertices in U are connected by an edge.
  - $|U| \geq k$.

Reduction to Single-Minded Allocation:

- **Items:** E (edges of the graph).
- **Bidders:** V (vertices of the graph).

NP-Hardness Reduction: Independent Set Problem

Independent Set Problem:

- **Input:** A graph $G = (V, E)$ and a number k.
- **Goal:** Find a set $U \subseteq V$ such that:
  - No two vertices in U are connected by an edge.
  - $|U| \geq k$.

Reduction to Single-Minded Allocation:

- **Items:** E (edges of the graph).
- **Bidders:** V (vertices of the graph).
- **Desired Bundle:** $S_v^* = \{e \in E \mid v \text{ is an endpoint of } e\}$.

NP-Hardness Reduction: Independent Set Problem

Independent Set Problem:

- **Input:** A graph $G = (V, E)$ and a number k.
- **Goal:** Find a set $U \subseteq V$ such that:
  - No two vertices in U are connected by an edge.
  - $|U| \geq k$.

Reduction to Single-Minded Allocation:

- **Items:** E (edges of the graph).
- **Bidders:** V (vertices of the graph).
- **Desired Bundle:** $S_v^* = \{e \in E \mid v \text{ is an endpoint of } e\}$.
- **Value:** $v_v^* = 1$.

Reduction to Independent Set: Conflict Graph

Example Setup:

Reduction to Independent Set: Conflict Graph

Example Setup:

- Bidder v_1 wants $\{A, B\}$.

Reduction to Independent Set: Conflict Graph

Example Setup:

- Bidder v_1 wants $\{A, B\}$.
- Bidder v_2 wants $\{B, C\}$.

Reduction to Independent Set: Conflict Graph

Example Setup:

- Bidder v_1 wants $\{A, B\}$.
- Bidder v_2 wants $\{B, C\}$.
- Bidder v_3 wants $\{C, D\}$.

Reduction to Independent Set: Conflict Graph

Example Setup:

- Bidder v_1 wants $\{A, B\}$.
- Bidder v_2 wants $\{B, C\}$.
- Bidder v_3 wants $\{C, D\}$.
- Bidder v_4 wants $\{A, E\}$.

Reduction to Independent Set: Conflict Graph

Example Setup:

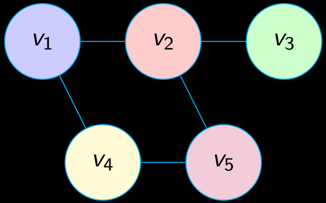
- Bidder v_1 wants $\{A, B\}$.
- Bidder v_2 wants $\{B, C\}$.
- Bidder v_3 wants $\{C, D\}$.
- Bidder v_4 wants $\{A, E\}$.
- Bidder v_5 wants $\{B, E\}$.

Reduction to Independent Set: Conflict Graph

Example Setup:

- Bidder v_1 wants $\{A, B\}$.
- Bidder v_2 wants $\{B, C\}$.
- Bidder v_3 wants $\{C, D\}$.
- Bidder v_4 wants $\{A, E\}$.
- Bidder v_5 wants $\{B, E\}$.

Conflict Graph:



Proof: Independent Set $\rightarrow$ Allocation

Mapping an Independent Set to an Allocation:

Proof: Independent Set $\rightarrow$ Allocation

Mapping an Independent Set to an Allocation:

- Let $U \subseteq V$ be an independent set in G.

Proof: Independent Set $\to$ Allocation

Mapping an Independent Set to an Allocation:

- Let $U \subseteq V$ be an independent set in G.
- For each $v \in U$, allocate the bundle S_v^*.

Proof: Independent Set $\rightarrow$ Allocation

Mapping an Independent Set to an Allocation:

- Let $U \subseteq V$ be an independent set in G.
- For each $v \in U$, allocate the bundle S_v^*.
- Since U is independent:

Proof: Independent Set $\rightarrow$ Allocation

Mapping an Independent Set to an Allocation:

- Let $U \subseteq V$ be an independent set in G.
- For each $v \in U$, allocate the bundle S_v^*.
- Since U is independent:
  - No two vertices in U share an edge.

Proof: Independent Set $\rightarrow$ Allocation

Mapping an Independent Set to an Allocation:

- Let $U \subseteq V$ be an independent set in G.
- For each $v \in U$, allocate the bundle S_v^*.
- Since U is independent:
  - No two vertices in U share an edge.
  - Therefore, $S_u^* \cap S_v^* = \emptyset$ for all $u, v \in U, u \neq v$.

Proof: Independent Set $\rightarrow$ Allocation

Mapping an Independent Set to an Allocation:

- Let $U \subseteq V$ be an independent set in G.
- For each $v \in U$, allocate the bundle S_v^*.
- Since U is independent:
  - No two vertices in U share an edge.
  - Therefore, $S_u^* \cap S_v^* = \emptyset$ for all $u, v \in U, u \neq v$.
- Resulting allocation $W = U$ is feasible in the single-minded allocation problem.
- Social welfare of this allocation:

$$\sum_{v \in W} v_v^* = |U|.$$

Proof: Allocation $\to$ Independent Set

Mapping an Allocation to an Independent Set:

Proof: Allocation $\to$ Independent Set

Mapping an Allocation to an Independent Set:

- Let W be a feasible allocation in the single-minded allocation problem.

Proof: Allocation $\rightarrow$ Independent Set

Mapping an Allocation to an Independent Set:

- Let W be a feasible allocation in the single-minded allocation problem.
- For each bidder $v \in W$, include v in the independent set U.

Proof: Allocation $\rightarrow$ Independent Set

Mapping an Allocation to an Independent Set:

- Let W be a feasible allocation in the single-minded allocation problem.
- For each bidder $v \in W$, include v in the independent set U.
- Since $S_u^* \cap S_v^* = \emptyset$ for all $u, v \in W$, $u \neq v$:

Proof: Allocation $\rightarrow$ Independent Set

Mapping an Allocation to an Independent Set:

- Let W be a feasible allocation in the single-minded allocation problem.
- For each bidder $v \in W$, include v in the independent set U.
- Since $S_u^* \cap S_v^* = \emptyset$ for all $u, v \in W, u \neq v$:
  - No two vertices in U share an edge.

Proof: Allocation $\rightarrow$ Independent Set

Mapping an Allocation to an Independent Set:

- Let W be a feasible allocation in the single-minded allocation problem.
- For each bidder $v \in W$, include v in the independent set U.
- Since $S_u^* \cap S_v^* = \emptyset$ for all $u, v \in W, u \neq v$:
  - No two vertices in U share an edge.
  - Therefore, U is an independent set in G.

Proof: Allocation $\rightarrow$ Independent Set

Mapping an Allocation to an Independent Set:

- Let W be a feasible allocation in the single-minded allocation problem.
- For each bidder $v \in W$, include v in the independent set U.
- Since $S_u^* \cap S_v^* = \emptyset$ for all $u, v \in W, u \neq v$:
  - No two vertices in U share an edge.
  - Therefore, U is an independent set in G.
- Size of the independent set:

$$|U| = \sum_{v \in W} 1.$$

Why Use a Non-VCG Mechanism?

- **Computational Complexity:**

Why Use a Non-VCG Mechanism?

- **Computational Complexity:**
  - Finding the exact social welfare-maximizing allocation (required for VCG payments) is NP-hard.

Why Use a Non-VCG Mechanism?

- **Computational Complexity:**
  - Finding the exact social welfare-maximizing allocation (required for VCG payments) is NP-hard.
  - Approximation is necessary for large-scale auctions, but approximations break VCG compatibility.

Why Use a Non-VCG Mechanism?

- **Computational Complexity:**
  - Finding the exact social welfare-maximizing allocation (required for VCG payments) is NP-hard.
  - Approximation is necessary for large-scale auctions, but approximations break VCG compatibility.
- **Strategic Considerations:**

Why Use a Non-VCG Mechanism?

- **Computational Complexity:**
  - Finding the exact social welfare-maximizing allocation (required for VCG payments) is NP-hard.
  - Approximation is necessary for large-scale auctions, but approximations break VCG compatibility.
- **Strategic Considerations:**
  - VCG mechanisms ensure truthful bidding when the allocation is exact.

Why Use a Non-VCG Mechanism?

- **Computational Complexity:**
  - Finding the exact social welfare-maximizing allocation (required for VCG payments) is NP-hard.
  - Approximation is necessary for large-scale auctions, but approximations break VCG compatibility.
- **Strategic Considerations:**
  - VCG mechanisms ensure truthful bidding when the allocation is exact.
  - Using approximations, however, can lead to strategic misreporting of values.

Why Use a Non-VCG Mechanism?

- **Computational Complexity:**
  - Finding the exact social welfare-maximizing allocation (required for VCG payments) is NP-hard.
  - Approximation is necessary for large-scale auctions, but approximations break VCG compatibility.
- **Strategic Considerations:**
  - VCG mechanisms ensure truthful bidding when the allocation is exact.
  - Using approximations, however, can lead to strategic misreporting of values.
- **Greedy Mechanism:**

Why Use a Non-VCG Mechanism?

- **Computational Complexity:**
  - Finding the exact social welfare-maximizing allocation (required for VCG payments) is NP-hard.
  - Approximation is necessary for large-scale auctions, but approximations break VCG compatibility.
- **Strategic Considerations:**
  - VCG mechanisms ensure truthful bidding when the allocation is exact.
  - Using approximations, however, can lead to strategic misreporting of values.
- **Greedy Mechanism:**
  - Achieves a $\sqrt{m}$-approximation of social welfare.

Why Use a Non-VCG Mechanism?

- **Computational Complexity:**
  - Finding the exact social welfare-maximizing allocation (required for VCG payments) is NP-hard.
  - Approximation is necessary for large-scale auctions, but approximations break VCG compatibility.
- **Strategic Considerations:**
  - VCG mechanisms ensure truthful bidding when the allocation is exact.
  - Using approximations, however, can lead to strategic misreporting of values.
- **Greedy Mechanism:**
  - Achieves a $\sqrt{m}$-approximation of social welfare.
  - Maintains incentive compatibility with simpler payment rules (critical value).

Why Use a Non-VCG Mechanism?

- **Computational Complexity:**
  - Finding the exact social welfare-maximizing allocation (required for VCG payments) is NP-hard.
  - Approximation is necessary for large-scale auctions, but approximations break VCG compatibility.
- **Strategic Considerations:**
  - VCG mechanisms ensure truthful bidding when the allocation is exact.
  - Using approximations, however, can lead to strategic misreporting of values.
- **Greedy Mechanism:**
  - Achieves a $\sqrt{m}$-approximation of social welfare.
  - Maintains incentive compatibility with simpler payment rules (critical value).
  - Provides computational efficiency for large instances.

Lemma 11.9: Incentive Compatibility Conditions

Lemma Statement: A mechanism for single-minded bidders, where losers pay 0, is incentive compatible if and only if:

Lemma 11.9: Incentive Compatibility Conditions

Lemma Statement: A mechanism for single-minded bidders, where losers pay 0, is incentive compatible if and only if:

1. **Monotonicity:** A bidder who wins with bid (S^*, v^*) keeps winning for any $v' > v^*$ and $S' \subseteq S^*$.

Lemma 11.9: Incentive Compatibility Conditions

Lemma Statement: A mechanism for single-minded bidders, where losers pay 0, is incentive compatible if and only if:

1. **Monotonicity:** A bidder who wins with bid (S^*, v^*) keeps winning for any $v' > v^*$ and $S' \subseteq S^*$.
2. **Critical Payment:** A bidder who wins pays the minimum value v' such that (S^*, v') still wins.

Proof Outline:

Lemma 11.9: Incentive Compatibility Conditions

Lemma Statement: A mechanism for single-minded bidders, where losers pay 0, is incentive compatible if and only if:

1. **Monotonicity:** A bidder who wins with bid (S^*, v^*) keeps winning for any $v' > v^*$ and $S' \subseteq S^*$.
2. **Critical Payment:** A bidder who wins pays the minimum value v' such that (S^*, v') still wins.

Proof Outline:

- Truthful reporting is dominant because:

Lemma 11.9: Incentive Compatibility Conditions

Lemma Statement: A mechanism for single-minded bidders, where losers pay 0, is incentive compatible if and only if:

1. **Monotonicity:** A bidder who wins with bid (S^*, v^*) keeps winning for any $v' > v^*$ and $S' \subseteq S^*$.
2. **Critical Payment:** A bidder who wins pays the minimum value v' such that (S^*, v') still wins.

Proof Outline:

- Truthful reporting is dominant because:
  - Overreporting v^* or shrinking S^* does not change the payment.

Lemma 11.9: Incentive Compatibility Conditions

Lemma Statement: A mechanism for single-minded bidders, where losers pay 0, is incentive compatible if and only if:

1. **Monotonicity:** A bidder who wins with bid (S^*, v^*) keeps winning for any $v' > v^*$ and $S' \subseteq S^*$.
2. **Critical Payment:** A bidder who wins pays the minimum value v' such that (S^*, v') still wins.

Proof Outline:

- Truthful reporting is dominant because:
  - Overreporting v^* or shrinking S^* does not change the payment.
  - Underreporting v^* or enlarging S^* risks losing the auction.

Proof of Lemma 11.9

Monotonicity Ensures Incentive Compatibility:

Proof of Lemma 11.9

Monotonicity Ensures Incentive Compatibility:

- If a bidder wins with (S^*, v^*), increasing v^* or shrinking S^* does not change the outcome.

Proof of Lemma 11.9

Monotonicity Ensures Incentive Compatibility:

- If a bidder wins with (S^*, v^*), increasing v^* or shrinking S^* does not change the outcome.
- Truthful bidding maximizes chances of winning without losing utility.

Proof of Lemma 11.9

Monotonicity Ensures Incentive Compatibility:

- If a bidder wins with (S^*, v^*), increasing v^* or shrinking S^* does not change the outcome.
- Truthful bidding maximizes chances of winning without losing utility.

Critical Payment Ensures No Strategic Gain:

Proof of Lemma 11.9

Monotonicity Ensures Incentive Compatibility:

- If a bidder wins with (S^*, v^*), increasing v^* or shrinking S^* does not change the outcome.
- Truthful bidding maximizes chances of winning without losing utility.

Critical Payment Ensures No Strategic Gain:

- Winning bidders pay the infimum v' needed to win, ensuring:

$$u = v^* - p \quad \text{(utility is maximized when bidding truthfully).}$$

Proof of Lemma 11.9

Monotonicity Ensures Incentive Compatibility:

- If a bidder wins with (S^*, v^*), increasing v^* or shrinking S^* does not change the outcome.
- Truthful bidding maximizes chances of winning without losing utility.

Critical Payment Ensures No Strategic Gain:

- Winning bidders pay the infimum v' needed to win, ensuring:

$$u = v^* - p \quad \text{(utility is maximized when bidding truthfully).}$$

- Misreporting risks:

Proof of Lemma 11.9

Monotonicity Ensures Incentive Compatibility:

- If a bidder wins with (S^*, v^*), increasing v^* or shrinking S^* does not change the outcome.
- Truthful bidding maximizes chances of winning without losing utility.

Critical Payment Ensures No Strategic Gain:

- Winning bidders pay the infimum v' needed to win, ensuring:

$$u = v^* - p \quad \text{(utility is maximized when bidding truthfully).}$$

- Misreporting risks:
  - Overreporting v^*: No effect on payment.

Proof of Lemma 11.9

Monotonicity Ensures Incentive Compatibility:

- If a bidder wins with (S^*, v^*), increasing v^* or shrinking S^* does not change the outcome.
- Truthful bidding maximizes chances of winning without losing utility.

Critical Payment Ensures No Strategic Gain:

- Winning bidders pay the infimum v' needed to win, ensuring:

$$u = v^* - p \quad \text{(utility is maximized when bidding truthfully).}$$

- Misreporting risks:
  - Overreporting v^*: No effect on payment.
  - Underreporting v^*: Risk of losing allocation.

Continued

Sufficiency: Monotonicity and critical payment ensure truthful reporting is a dominant strategy.

Continued

Sufficiency: Monotonicity and critical payment ensure truthful reporting is a dominant strategy.
Necessity: Without these conditions, bidders can improve utility by misreporting.

Greedy Mechanism for Allocation

Steps of the Mechanism:

Greedy Mechanism for Allocation

Steps of the Mechanism:

1. Rank bidders by $\frac{v_i^*}{\sqrt{|S_i^*|}}$, prioritizing smaller bundles with higher values.

Greedy Mechanism for Allocation

Steps of the Mechanism:

1. Rank bidders by $\frac{v_i^*}{\sqrt{|S_i^*|}}$, prioritizing smaller bundles with higher values.
2. Allocate items greedily:

Greedy Mechanism for Allocation

Steps of the Mechanism:

1. Rank bidders by $\frac{v_i^*}{\sqrt{|S_i^*|}}$, prioritizing smaller bundles with higher values.
2. Allocate items greedily:
   - A bidder is selected only if their desired bundle S_i^* does not conflict with already allocated items.

Greedy Mechanism for Allocation

Steps of the Mechanism:

1. Rank bidders by $\frac{v_i^*}{\sqrt{|S_i^*|}}$, prioritizing smaller bundles with higher values.
2. Allocate items greedily:
   - A bidder is selected only if their desired bundle S_i^* does not conflict with already allocated items.
3. Winners pay the critical value, i.e., the minimum value required to retain their allocation.

Greedy Mechanism for Allocation

Steps of the Mechanism:

1. Rank bidders by $\frac{v_i^*}{\sqrt{|S_i^*|}}$, prioritizing smaller bundles with higher values.
2. Allocate items greedily:
   - A bidder is selected only if their desired bundle S_i^* does not conflict with already allocated items.
3. Winners pay the critical value, i.e., the minimum value required to retain their allocation.

Output: An allocation W that approximates the optimal social welfare.

Approximation Guarantee: Setup and Bounding Contributions

Approximation Guarantee: Setup and Bounding Contributions

Goal: Prove.

$$\frac{\text{Optimal Welfare}}{\text{Greedy Welfare}} \leq \sqrt{m}.$$

Approximation Guarantee: Setup and Bounding Contributions

Goal: Prove.

$$\frac{\text{Optimal Welfare}}{\text{Greedy Welfare}} \leq \sqrt{m}.$$

Setup:

Approximation Guarantee: Setup and Bounding Contributions

Goal: Prove.

$$\frac{\text{Optimal Welfare}}{\text{Greedy Welfare}} \leq \sqrt{m}.$$

Setup:

- Let OPT denote the optimal allocation, with welfare:

$$\text{OPT Welfare} = \sum_{j \in OPT} v_j^*.$$

Approximation Guarantee: Setup and Bounding Contributions

Goal: Prove.

$$\frac{\text{Optimal Welfare}}{\text{Greedy Welfare}} \leq \sqrt{m}.$$

Setup:

- Let OPT denote the optimal allocation, with welfare:

$$\text{OPT Welfare} = \sum_{j \in OPT} v_j^*.$$

- Let W denote the allocation from the greedy mechanism, with welfare:

$$\text{Greedy Welfare} = \sum_{i \in W} v_i^*.$$

Approximation Guarantee: Setup and Bounding Contributions

Goal: Prove.

$$\frac{\text{Optimal Welfare}}{\text{Greedy Welfare}} \leq \sqrt{m}.$$

Setup:

- Let OPT denote the optimal allocation, with welfare:

$$\text{OPT Welfare} = \sum_{j \in OPT} v_j^*.$$

- Let W denote the allocation from the greedy mechanism, with welfare:

$$\text{Greedy Welfare} = \sum_{i \in W} v_i^*.$$

- For each $i \in W$, define its conflict-set OPT_i as:

$$OPT_i = \{j \in OPT \mid S_j^* \cap S_i^* \neq \emptyset\}.$$

Continued

Bounding the Contribution of OPT:

Continued

Bounding the Contribution of *OPT*:

- The total value of *OPT* can be rewritten as:

$$\sum_{j \in OPT} v_j^* \leq \sum_{i \in W} \sum_{j \in OPT_i} v_j^*.$$

Continued

Bounding the Contribution of OPT:

- The total value of OPT can be rewritten as:

$$\sum_{j \in OPT} v_j^* \leq \sum_{i \in W} \sum_{j \in OPT_i} v_j^*.$$

- Each $j \in OPT_i$ was not selected by the greedy algorithm, so:

$$v_j^* \leq v_i^* \cdot \frac{\sqrt{|S_j^*|}}{\sqrt{|S_i^*|}}.$$

Approximation Guarantee: Applying Cauchy-Schwarz and Final Bound

Applying Cauchy-Schwarz Inequality:

Approximation Guarantee: Applying Cauchy-Schwarz and Final Bound

Applying Cauchy-Schwarz Inequality:

- Summing over $j \in OPT_i$, we bound the sum of square roots using Cauchy-Schwarz:

$$\sum_{j \in OPT_i} \sqrt{|S_j^*|} \leq \sqrt{|OPT_i|} \cdot \sqrt{\sum_{j \in OPT_i} |S_j^*|}.$$

Approximation Guarantee: Applying Cauchy-Schwarz and Final Bound

Applying Cauchy-Schwarz Inequality:

- Summing over $j \in OPT_i$, we bound the sum of square roots using Cauchy-Schwarz:

$$\sum_{j \in OPT_i} \sqrt{|S_j^*|} \leq \sqrt{|OPT_i|} \cdot \sqrt{\sum_{j \in OPT_i} |S_j^*|}.$$

- Since $|OPT_i| \leq |S_i^*|$ and $\sum_{j \in OPT_i} |S_j^*| \leq m$:

$$\sum_{j \in OPT_i} \sqrt{|S_j^*|} \leq \sqrt{|S_i^*|} \cdot \sqrt{m}.$$

Approximation Guarantee: Applying Cauchy-Schwarz and Final Bound

Applying Cauchy-Schwarz Inequality:

- Summing over $j \in OPT_i$, we bound the sum of square roots using Cauchy-Schwarz:

$$\sum_{j \in OPT_i} \sqrt{|S_j^*|} \leq \sqrt{|OPT_i|} \cdot \sqrt{\sum_{j \in OPT_i} |S_j^*|}.$$

- Since $|OPT_i| \leq |S_i^*|$ and $\sum_{j \in OPT_i} |S_j^*| \leq m$:

$$\sum_{j \in OPT_i} \sqrt{|S_j^*|} \leq \sqrt{|S_i^*|} \cdot \sqrt{m}.$$

Bounding the Contribution of OPT_i:

- Substituting this bound into the earlier inequality:

$$\sum_{j \in OPT_i} v_j^* \leq v_i^* \cdot \frac{\sqrt{|S_i^*|} \cdot \sqrt{m}}{\sqrt{|S_i^*|}} = v_i^* \cdot \sqrt{m}.$$

Continued

Final Bound for OPT:

Continued

Final Bound for OPT:

- Summing over all $i \in W$, the total contribution of OPT:

$$\sum_{j \in OPT} v_j^* \leq \sum_{i \in W} \left(v_i^* \cdot \sqrt{m} \right).$$

Continued

Final Bound for OPT:

- Summing over all $i \in W$, the total contribution of OPT:

$$\sum_{j \in OPT} v_j^* \leq \sum_{i \in W} \left(v_i^* \cdot \sqrt{m} \right).$$

- Factoring out $\sqrt{m}$:

$$\sum_{j \in OPT} v_j^* \leq \sqrt{m} \cdot \sum_{i \in W} v_i^*.$$

Continued

Final Bound for OPT:

- Summing over all $i \in W$, the total contribution of OPT:

$$\sum_{j \in OPT} v_j^* \leq \sum_{i \in W} \left(v_i^* \cdot \sqrt{m} \right).$$

- Factoring out $\sqrt{m}$:

$$\sum_{j \in OPT} v_j^* \leq \sqrt{m} \cdot \sum_{i \in W} v_i^*.$$

- Therefore:

$$\frac{\text{Optimal Welfare}}{\text{Greedy Welfare}} \leq \sqrt{m}.$$

Conclusion

- Single-minded case simplifies bidder preferences but remains computationally challenging.

Conclusion

- Single-minded case simplifies bidder preferences but remains computationally challenging.
- Greedy mechanism offers a $\sqrt{m}$-approximation with efficient computation.

Conclusion

- Single-minded case simplifies bidder preferences but remains computationally challenging.
- Greedy mechanism offers a $\sqrt{m}$-approximation with efficient computation.
- Future work could include exploring tighter approximation bounds and scalable mechanisms.

Thank You

Thank You!

Thank You

Thank You!

Questions are welcome!