Algorithmic game theory

Martin Balko

10th lecture

December 6th 2024

Mechanism design basics

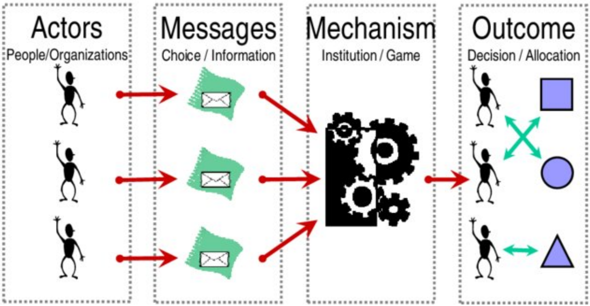
Mechanism design

Mechanism design

- Designing games toward desired objectives.

Mechanism design

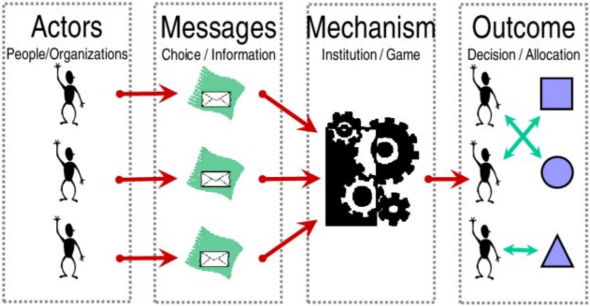
- Designing games toward desired objectives.
- We try to design rules of the game so that strategic behavior by participants leads to a desirable outcome.



Source: Innovations in Defense Acquisition: Asymmetric Information and Incentive Contract Design

Mechanism design

- Designing games toward desired objectives.
- We try to design rules of the game so that strategic behavior by participants leads to a desirable outcome.

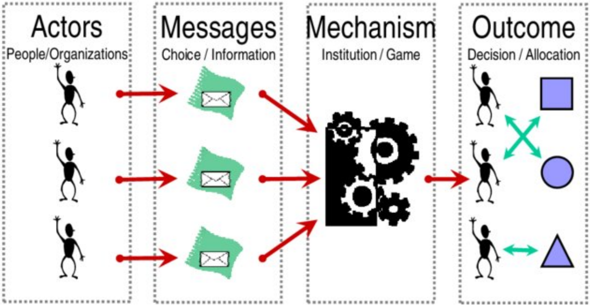


Source: Innovations in Defense Acquisition: Asymmetric Information and Incentive Contract Design

- We start with single item auctions.

Mechanism design

- Designing games toward desired objectives.
- We try to design rules of the game so that strategic behavior by participants leads to a desirable outcome.



Source: Innovations in Defense Acquisition: Asymmetric Information and Incentive Contract Design

- We start with single item auctions.
- We then extend these desired properties to a more general setting of single-parameter environments using so-called Myerson's lemma.

Single item auctions

-

Source: https://www.widewalls.ch

Single item auctions

Single item auctions

- There is a seller selling a single good (a painting, for example) to some number n of bidders who are potentially interested in buying the item.

Single item auctions

- There is a seller selling a single good (a painting, for example) to some number n of bidders who are potentially interested in buying the item.
- Each bidder i has a valuation v_i that he is willing to pay for the item.

Single item auctions

- There is a seller selling a single good (a painting, for example) to some number n of bidders who are potentially interested in buying the item.
- Each bidder i has a valuation v_i that he is willing to pay for the item. The other bidders nor the seller know v_i.

Single item auctions

- There is a seller selling a single good (a painting, for example) to some number n of bidders who are potentially interested in buying the item.
- Each bidder i has a valuation v_i that he is willing to pay for the item. The other bidders nor the seller know v_i.
- Each bidder i privately communicates a bid b_i to the seller.

Single item auctions

- There is a seller selling a single good (a painting, for example) to some number n of bidders who are potentially interested in buying the item.
- Each bidder i has a valuation v_i that he is willing to pay for the item. The other bidders nor the seller know v_i.
- Each bidder i privately communicates a bid b_i to the seller. The seller then decides who receives the item (if any) and the selling price p.

Single item auctions

- There is a seller selling a single good (a painting, for example) to some number n of bidders who are potentially interested in buying the item.
- Each bidder i has a valuation v_i that he is willing to pay for the item. The other bidders nor the seller know v_i.
- Each bidder i privately communicates a bid b_i to the seller. The seller then decides who receives the item (if any) and the selling price p.
- If a bidder loses the auction, then his utility u_i is 0.

Single item auctions

- There is a seller selling a single good (a painting, for example) to some number n of bidders who are potentially interested in buying the item.
- Each bidder i has a valuation v_i that he is willing to pay for the item. The other bidders nor the seller know v_i.
- Each bidder i privately communicates a bid b_i to the seller. The seller then decides who receives the item (if any) and the selling price p.
- If a bidder loses the auction, then his utility u_i is 0. If the bidder wins the auction at price p, then his utility is $u_i = v_i - p$.

Single item auctions

- There is a seller selling a single good (a painting, for example) to some number n of bidders who are potentially interested in buying the item.
- Each bidder i has a valuation v_i that he is willing to pay for the item. The other bidders nor the seller know v_i.
- Each bidder i privately communicates a bid b_i to the seller. The seller then decides who receives the item (if any) and the selling price p.
- If a bidder loses the auction, then his utility u_i is 0. If the bidder wins the auction at price p, then his utility is $u_i = v_i - p$.

- Our goal is to design a mechanism how to decide the allocation of the item to a bidder in a way that cannot be strategically manipulated.

Single item auctions

- There is a seller selling a single good (a painting, for example) to some number n of bidders who are potentially interested in buying the item.
- Each bidder i has a valuation v_i that he is willing to pay for the item. The other bidders nor the seller know v_i.
- Each bidder i privately communicates a bid b_i to the seller. The seller then decides who receives the item (if any) and the selling price p.
- If a bidder loses the auction, then his utility u_i is 0. If the bidder wins the auction at price p, then his utility is $u_i = v_i - p$.

- Our goal is to design a mechanism how to decide the allocation of the item to a bidder in a way that cannot be strategically manipulated.
- To do so, we need to appropriately implement the rules for the seller how to decide the winner and the selling price.

How not to design a single item auction

How not to design a single item auction

- Not every choice of the rules leads to a desirable auction.

How not to design a single item auction

- Not every choice of the rules leads to a desirable auction.
- Consider selling the item for free to bidder i with the highest bid b_i.

How not to design a single item auction

- Not every choice of the rules leads to a desirable auction.

- Consider selling the item for free to bidder i with the highest bid b_i.
  - This is not a very good choice, as then the bidders will benefit from exaggerating their valuations v_i by reporting b_i that is much larger than v_i.

How not to design a single item auction

- Not every choice of the rules leads to a desirable auction.

- Consider selling the item for free to bidder i with the highest bid b_i.
  - This is not a very good choice, as then the bidders will benefit from exaggerating their valuations v_i by reporting b_i that is much larger than v_i.
  - So this into a game of “who can name the highest number”.

How not to design a single item auction

- Not every choice of the rules leads to a desirable auction.

- Consider selling the item for free to bidder i with the highest bid b_i.
  - This is not a very good choice, as then the bidders will benefit from exaggerating their valuations v_i by reporting b_i that is much larger than v_i.
  - So this into a game of "who can name the highest number".

- Consider selling the item to bidder i with the highest bid b_i for the selling price b_i.

How not to design a single item auction

- Not every choice of the rules leads to a desirable auction.

- Consider selling the item for free to bidder i with the highest bid b_i.
  - This is not a very good choice, as then the bidders will benefit from exaggerating their valuations v_i by reporting b_i that is much larger than v_i.
  - So this into a game of "who can name the highest number".

- Consider selling the item to bidder i with the highest bid b_i for the selling price b_i.
  - This looks much more reasonable and such auctions are common in practice.

How not to design a single item auction

- Not every choice of the rules leads to a desirable auction.

- Consider selling the item for free to bidder i with the highest bid b_i.
    - This is not a very good choice, as then the bidders will benefit from exaggerating their valuations v_i by reporting b_i that is much larger than v_i.
    - So this into a game of "who can name the highest number".

- Consider selling the item to bidder i with the highest bid b_i for the selling price b_i.
    - This looks much more reasonable and such auctions are common in practice. However, there are still some drawbacks.

How not to design a single item auction

- Not every choice of the rules leads to a desirable auction.

- Consider selling the item for free to bidder i with the highest bid b_i.
  - This is not a very good choice, as then the bidders will benefit from exaggerating their valuations v_i by reporting b_i that is much larger than v_i.
  - So this into a game of "who can name the highest number".

- Consider selling the item to bidder i with the highest bid b_i for the selling price b_i.
  - This looks much more reasonable and such auctions are common in practice. However, there are still some drawbacks.
  - It is difficult for the bidders to figure out how to bid.

How not to design a single item auction

- Not every choice of the rules leads to a desirable auction.

- Consider selling the item for free to bidder i with the highest bid b_i.
  - This is not a very good choice, as then the bidders will benefit from exaggerating their valuations v_i by reporting b_i that is much larger than v_i.
  - So this into a game of "who can name the highest number".

- Consider selling the item to bidder i with the highest bid b_i for the selling price b_i.
  - This looks much more reasonable and such auctions are common in practice. However, there are still some drawbacks.
  - It is difficult for the bidders to figure out how to bid. If bidder i wins and pays $b_i = v_i$, then his utility is $v_i - b_i = 0$, the same as if he loses the bid.

How not to design a single item auction

- Not every choice of the rules leads to a desirable auction.

- Consider selling the item for free to bidder i with the highest bid b_i.
  - This is not a very good choice, as then the bidders will benefit from exaggerating their valuations v_i by reporting b_i that is much larger than v_i.
  - So this into a game of "who can name the highest number".

- Consider selling the item to bidder i with the highest bid b_i for the selling price b_i.
  - This looks much more reasonable and such auctions are common in practice. However, there are still some drawbacks.
  - It is difficult for the bidders to figure out how to bid. If bidder i wins and pays $b_i = v_i$, then his utility is $v_i - b_i = 0$, the same as if he loses the bid. So he should be declaring lower bid b_i than v_i, but what is the value b_i he should bid?

So what do we want?

So what do we want?

- We now formalize the conditions that our auction should satisfy.

So what do we want?

- We now formalize the conditions that our auction should satisfy.
- A dominant strategy for bidder i is a strategy that maximizes the utility of bidder i, no matter what the other bidders do.

So what do we want?

- We now formalize the conditions that our auction should satisfy.
- A dominant strategy for bidder i is a strategy that maximizes the utility of bidder i, no matter what the other bidders do.
- The social surplus is $\sum_{i=1}^{n} v_i x_i$, where $x_i = 1$ if bidder i wins and $x_i = 0$ otherwise subject to $\sum_{i=1}^{n} x_i \leq 1$ (the seller sells only a single item).

So what do we want?

- We now formalize the conditions that our auction should satisfy.
- A dominant strategy for bidder i is a strategy that maximizes the utility of bidder i, no matter what the other bidders do.
- The social surplus is $\sum_{i=1}^{n} v_i x_i$, where $x_i = 1$ if bidder i wins and $x_i = 0$ otherwise subject to $\sum_{i=1}^{n} x_i \leq 1$ (the seller sells only a single item).
- We want our auction to be awesome

So what do we want?

- We now formalize the conditions that our auction should satisfy.
- A dominant strategy for bidder i is a strategy that maximizes the utility of bidder i, no matter what the other bidders do.
- The social surplus is $\sum_{i=1}^{n} v_i x_i$, where $x_i = 1$ if bidder i wins and $x_i = 0$ otherwise subject to $\sum_{i=1}^{n} x_i \leq 1$ (the seller sells only a single item).
- We want our auction to be awesome, that is, it should satisfy:

So what do we want?

- We now formalize the conditions that our auction should satisfy.
- A dominant strategy for bidder i is a strategy that maximizes the utility of bidder i, no matter what the other bidders do.
- The social surplus is $\sum_{i=1}^{n} v_i x_i$, where $x_i = 1$ if bidder i wins and $x_i = 0$ otherwise subject to $\sum_{i=1}^{n} x_i \leq 1$ (the seller sells only a single item).

- We want our auction to be awesome, that is, it should satisfy:
  - Strong incentive guarantees: The auction is dominant-strategy incentive-compatible (DSIC), that is, it satisfies the following two properties. Every bidder has a dominant strategy: bid truthfully, that is, set his bid b_i to his valuation v_i.

So what do we want?

- We now formalize the conditions that our auction should satisfy.
- A dominant strategy for bidder i is a strategy that maximizes the utility of bidder i, no matter what the other bidders do.
- The social surplus is $\sum_{i=1}^{n} v_i x_i$, where $x_i = 1$ if bidder i wins and $x_i = 0$ otherwise subject to $\sum_{i=1}^{n} x_i \leq 1$ (the seller sells only a single item).

- We want our auction to be awesome, that is, it should satisfy:
  - Strong incentive guarantees: The auction is dominant-strategy incentive-compatible (DSIC), that is, it satisfies the following two properties. Every bidder has a dominant strategy: bid truthfully, that is, set his bid b_i to his valuation v_i. Moreover, the utility of every truth-telling bidder is guaranteed to be non-negative.

So what do we want?

- We now formalize the conditions that our auction should satisfy.
- A dominant strategy for bidder i is a strategy that maximizes the utility of bidder i, no matter what the other bidders do.
- The social surplus is $\sum_{i=1}^{n} v_i x_i$, where $x_i = 1$ if bidder i wins and $x_i = 0$ otherwise subject to $\sum_{i=1}^{n} x_i \leq 1$ (the seller sells only a single item).

- We want our auction to be awesome, that is, it should satisfy:
  - Strong incentive guarantees: The auction is dominant-strategy incentive-compatible (DSIC), that is, it satisfies the following two properties. Every bidder has a dominant strategy: bid truthfully, that is, set his bid b_i to his valuation v_i. Moreover, the utility of every truth-telling bidder is guaranteed to be non-negative.
  - Strong performance guarantees: If all bidders bid truthfully then the auction maximizes the social surplus.

So what do we want?

- We now formalize the conditions that our auction should satisfy.
- A dominant strategy for bidder i is a strategy that maximizes the utility of bidder i, no matter what the other bidders do.
- The social surplus is $\sum_{i=1}^{n} v_i x_i$, where $x_i = 1$ if bidder i wins and $x_i = 0$ otherwise subject to $\sum_{i=1}^{n} x_i \leq 1$ (the seller sells only a single item).

- We want our auction to be awesome, that is, it should satisfy:
  - Strong incentive guarantees: The auction is dominant-strategy incentive-compatible (DSIC), that is, it satisfies the following two properties. Every bidder has a dominant strategy: bid truthfully, that is, set his bid b_i to his valuation v_i. Moreover, the utility of every truth-telling bidder is guaranteed to be non-negative.
  - Strong performance guarantees: If all bidders bid truthfully then the auction maximizes the social surplus.
  - Computational efficiency: The auction can be implemented in polynomial time.

Why do we want this?

Why do we want this?

- Let us justify why do we insist on these three conditions.

Why do we want this?

- Let us justify why do we insist on these three conditions.
    - Strong incentive guarantees:

Why do we want this?

- Let us justify why do we insist on these three conditions.
    - Strong incentive guarantees: DSIC property makes it easy to choose a bid for each bidder (bid $b_i = v_i$).

Why do we want this?

- Let us justify why do we insist on these three conditions.
  - Strong incentive guarantees: DSIC property makes it easy to choose a bid for each bidder (bid $b_i = v_i$). It is also easy for the seller to reason about the auction's outcome, he can only assume that bidders will bid truthfully.

Why do we want this?

- Let us justify why do we insist on these three conditions.
  - Strong incentive guarantees: DSIC property makes it easy to choose a bid for each bidder (bid $b_i = v_i$). It is also easy for the seller to reason about the auction's outcome, he can only assume that bidders will bid truthfully.
  - Strong performance guarantees:

Why do we want this?

- Let us justify why do we insist on these three conditions.
  - Strong incentive guarantees: DSIC property makes it easy to choose a bid for each bidder (bid $b_i = v_i$). It is also easy for the seller to reason about the auction's outcome, he can only assume that bidders will bid truthfully.
  - Strong performance guarantees: DSIC by itself is not enough

Why do we want this?

- Let us justify why do we insist on these three conditions.
  - Strong incentive guarantees: DSIC property makes it easy to choose a bid for each bidder (bid $b_i = v_i$). It is also easy for the seller to reason about the auction's outcome, he can only assume that bidders will bid truthfully.
  - Strong performance guarantees: DSIC by itself is not enough (giving the item for free to a random bidder or giving the item to nobody is DSIC).

Why do we want this?

- Let us justify why do we insist on these three conditions.
  - Strong incentive guarantees: DSIC property makes it easy to choose a bid for each bidder (bid $b_i = v_i$). It is also easy for the seller to reason about the auction's outcome, he can only assume that bidders will bid truthfully.
  - Strong performance guarantees: DSIC by itself is not enough (giving the item for free to a random bidder or giving the item to nobody is DSIC). This property successfully identifies the bidder with the highest valuation even though if this is private information.

Why do we want this?

- Let us justify why do we insist on these three conditions.
  - Strong incentive guarantees: DSIC property makes it easy to choose a bid for each bidder (bid $b_i = v_i$). It is also easy for the seller to reason about the auction's outcome, he can only assume that bidders will bid truthfully.
  - Strong performance guarantees: DSIC by itself is not enough (giving the item for free to a random bidder or giving the item to nobody is DSIC). This property successfully identifies the bidder with the highest valuation even though if this is private information. That is, we solve the surplus-maximization optimization problem.

Why do we want this?

- Let us justify why do we insist on these three conditions.
  - Strong incentive guarantees: DSIC property makes it easy to choose a bid for each bidder (bid $b_i = v_i$). It is also easy for the seller to reason about the auction's outcome, he can only assume that bidders will bid truthfully.
  - Strong performance guarantees: DSIC by itself is not enough (giving the item for free to a random bidder or giving the item to nobody is DSIC). This property successfully identifies the bidder with the highest valuation even though if this is private information. That is, we solve the surplus-maximization optimization problem.
  - Computational efficiency: should be obviously desirable.

Why do we want this?

- Let us justify why do we insist on these three conditions.
    - Strong incentive guarantees: DSIC property makes it easy to choose a bid for each bidder (bid $b_i = v_i$). It is also easy for the seller to reason about the auction's outcome, he can only assume that bidders will bid truthfully.
    - Strong performance guarantees: DSIC by itself is not enough (giving the item for free to a random bidder or giving the item to nobody is DSIC). This property successfully identifies the bidder with the highest valuation even though if this is private information. That is, we solve the surplus-maximization optimization problem.
    - Computational efficiency: should be obviously desirable.
- So this is the auction that we want.

Why do we want this?

- Let us justify why do we insist on these three conditions.
  - Strong incentive guarantees: DSIC property makes it easy to choose a bid for each bidder (bid $b_i = v_i$). It is also easy for the seller to reason about the auction's outcome, he can only assume that bidders will bid truthfully.
  - Strong performance guarantees: DSIC by itself is not enough (giving the item for free to a random bidder or giving the item to nobody is DSIC). This property successfully identifies the bidder with the highest valuation even though if this is private information. That is, we solve the surplus-maximization optimization problem.
  - Computational efficiency: should be obviously desirable.
- So this is the auction that we want. Is it attainable though?

Vickrey's auction

Vickrey's auction

- An awesome auction exists!

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\ldots,n\} \setminus \{i\}} b_j$.

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\ldots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof:

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof: We need to verify the three conditions.

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\ldots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof: We need to verify the three conditions.
  - Strong incentive guarantees:

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\ldots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof: We need to verify the three conditions.
  - Strong incentive guarantees: We show that utility of i is maximized for $b_i = v_i$.

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof: We need to verify the three conditions.
  - Strong incentive guarantees: We show that utility of i is maximized for $b_i = v_i$. Let $B = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$.

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\ldots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof: We need to verify the three conditions.
  - Strong incentive guarantees: We show that utility of i is maximized for $b_i = v_i$. Let $B = \max_{j \in \{1,\ldots,n\} \setminus \{i\}} b_j$. If $b_i < B$, then i loses and gets utility 0.

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\ldots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof: We need to verify the three conditions.
  - Strong incentive guarantees: We show that utility of i is maximized for $b_i = v_i$. Let $B = \max_{j \in \{1,\ldots,n\} \setminus \{i\}} b_j$. If $b_i < B$, then i loses and gets utility 0. If $b_i \geq B$, then i wins and gets utility $v_i - B$.

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\ldots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof: We need to verify the three conditions.
  - Strong incentive guarantees: We show that utility of i is maximized for $b_i = v_i$. Let $B = \max_{j \in \{1,\ldots,n\} \setminus \{i\}} b_j$. If $b_i < B$, then i loses and gets utility 0. If $b_i \geq B$, then i wins and gets utility $v_i - B$. If $v_i < B$, then i can get at most $\max\{0, v_i - B\} = 0$.

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\ldots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof: We need to verify the three conditions.
  - Strong incentive guarantees: We show that utility of i is maximized for $b_i = v_i$. Let $B = \max_{j \in \{1,\ldots,n\} \setminus \{i\}} b_j$. If $b_i < B$, then i loses and gets utility 0. If $b_i \geq B$, then i wins and gets utility $v_i - B$. If $v_i < B$, then i can get at most $\max\{0, v_i - B\} = 0$. If $v_i \geq B$, then i can get at most $\max\{0, v_i - B\} = v_i - B \geq 0$.

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof: We need to verify the three conditions.
  - Strong incentive guarantees: We show that utility of i is maximized for $b_i = v_i$. Let $B = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$. If $b_i < B$, then i loses and gets utility 0. If $b_i \geq B$, then i wins and gets utility $v_i - B$. If $v_i < B$, then i can get at most $\max\{0, v_i - B\} = 0$. If $v_i \geq B$, then i can get at most $\max\{0, v_i - B\} = v_i - B \geq 0$. He gets these by bidding truthfully.

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof: We need to verify the three conditions.
  - Strong incentive guarantees: We show that utility of i is maximized for $b_i = v_i$. Let $B = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$. If $b_i < B$, then i loses and gets utility 0. If $b_i \geq B$, then i wins and gets utility $v_i - B$. If $v_i < B$, then i can get at most $\max\{0, v_i - B\} = 0$. If $v_i \geq B$, then i can get at most $\max\{0, v_i - B\} = v_i - B \geq 0$. He gets these by bidding truthfully.
  - Strong performance guarantees:

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof: We need to verify the three conditions.
  - Strong incentive guarantees: We show that utility of i is maximized for $b_i = v_i$. Let $B = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$. If $b_i < B$, then i loses and gets utility 0. If $b_i \geq B$, then i wins and gets utility $v_i - B$. If $v_i < B$, then i can get at most $\max\{0, v_i - B\} = 0$. If $v_i \geq B$, then i can get at most $\max\{0, v_i - B\} = v_i - B \geq 0$. He gets these by bidding truthfully.
  - Strong performance guarantees: If i is the winner, then $v_i \geq v_j$ for every j, as all bidders bid truthfully.

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof: We need to verify the three conditions.
  - Strong incentive guarantees: We show that utility of i is maximized for $b_i = v_i$. Let $B = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$. If $b_i < B$, then i loses and gets utility 0. If $b_i \geq B$, then i wins and gets utility $v_i - B$. If $v_i < B$, then i can get at most $\max\{0, v_i - B\} = 0$. If $v_i \geq B$, then i can get at most $\max\{0, v_i - B\} = v_i - B \geq 0$. He gets these by bidding truthfully.
  - Strong performance guarantees: If i is the winner, then $v_i \geq v_j$ for every j, as all bidders bid truthfully. Since $x_i = 1$ and $x_j = 0$ for $j \neq i$, the social surplus is then equal to v_i and is maximized.

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof: We need to verify the three conditions.
  - Strong incentive guarantees: We show that utility of i is maximized for $b_i = v_i$. Let $B = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$. If $b_i < B$, then i loses and gets utility 0. If $b_i \geq B$, then i wins and gets utility $v_i - B$. If $v_i < B$, then i can get at most $\max\{0, v_i - B\} = 0$. If $v_i \geq B$, then i can get at most $\max\{0, v_i - B\} = v_i - B \geq 0$. He gets these by bidding truthfully.
  - Strong performance guarantees: If i is the winner, then $v_i \geq v_j$ for every j, as all bidders bid truthfully. Since $x_i = 1$ and $x_j = 0$ for $j \neq i$, the social surplus is then equal to v_i and is maximized.
  - Computational efficiency:

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof: We need to verify the three conditions.
  - Strong incentive guarantees: We show that utility of i is maximized for $b_i = v_i$. Let $B = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$. If $b_i < B$, then i loses and gets utility 0. If $b_i \geq B$, then i wins and gets utility $v_i - B$. If $v_i < B$, then i can get at most $\max\{0, v_i - B\} = 0$. If $v_i \geq B$, then i can get at most $\max\{0, v_i - B\} = v_i - B \geq 0$. He gets these by bidding truthfully.
  - Strong performance guarantees: If i is the winner, then $v_i \geq v_j$ for every j, as all bidders bid truthfully. Since $x_i = 1$ and $x_j = 0$ for $j \neq i$, the social surplus is then equal to v_i and is maximized.
  - Computational efficiency: The auction runs in linear time.

Vickrey's auction

- An awesome auction exists! And it is surprisingly simple.
- Vickrey's second price auction: the winner is the bidder i with the highest bid b_i and pays the second highest bid $p = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$.

Theorem 3.3 (Vickrey, 1961)

Vickrey's auction is awesome.

- Proof: We need to verify the three conditions.
  - Strong incentive guarantees: We show that utility of i is maximized for $b_i = v_i$. Let $B = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$. If $b_i < B$, then i loses and gets utility 0. If $b_i \geq B$, then i wins and gets utility $v_i - B$. If $v_i < B$, then i can get at most $\max\{0, v_i - B\} = 0$. If $v_i \geq B$, then i can get at most $\max\{0, v_i - B\} = v_i - B \geq 0$. He gets these by bidding truthfully.
  - Strong performance guarantees: If i is the winner, then $v_i \geq v_j$ for every j, as all bidders bid truthfully. Since $x_i = 1$ and $x_j = 0$ for $j \neq i$, the social surplus is then equal to v_i and is maximized.
  - Computational efficiency: The auction runs in linear time. □

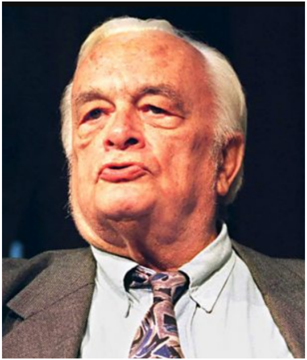
Vickrey's auction: remarks

Vickrey's auction: remarks

- First described by William Vickrey in 1961 though it had been used by stamp collectors since 1893.

Vickrey's auction: remarks

- First described by William Vickrey in 1961 though it had been used by stamp collectors since 1893.



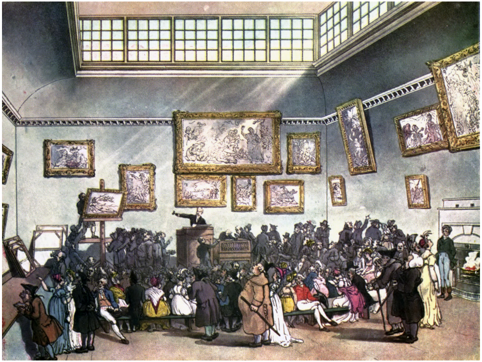
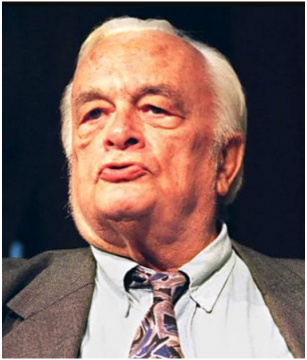


Figure: William Vickrey (1914–1996).

Sources: https://en.wikipedia.org and https://ichef.bbci.co.uk/

Vickrey's auction: remarks

- First described by William Vickrey in 1961 though it had been used by stamp collectors since 1893.



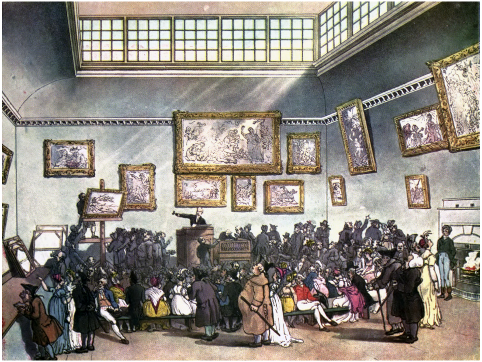


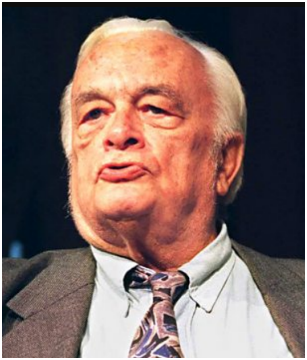
Figure: William Vickrey (1914–1996).

Sources: https://en.wikipedia.org and https://ichef.bbci.co.uk/

- Vickrey's auction is used also in, for example, network routing.

Vickrey's auction: remarks

- First described by William Vickrey in 1961 though it had been used by stamp collectors since 1893.



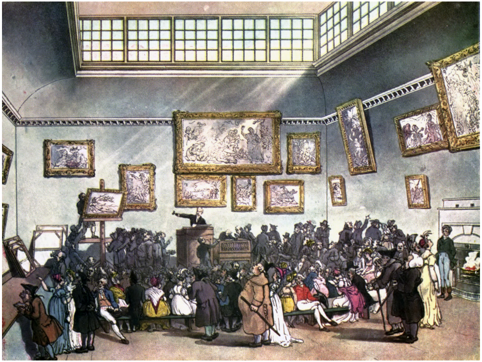


Figure: William Vickrey (1914–1996).

Sources: https://en.wikipedia.org and https://ichef.bbci.co.uk/

- Vickrey's auction is used also in, for example, network routing.
- Vickrey posthumously received a Nobel prize in Economic Sciences.

Single parameter environments

Single parameter environments

- Now that we have suceeded in single-item auction setting, can we design awesome mechanisms in more general settings?

Single parameter environments

- Now that we have suceeded in single-item auction setting, can we design awesome mechanisms in more general settings? We consider the following environments.

Single parameter environments

- Now that we have suceeded in single-item auction setting, can we design awesome mechanisms in more general settings? We consider the following environments.
- In a single-parameter environment, there are n bidders, each bidder i has a private valuation v_i (a value "per unit of the goods").

Single parameter environments

- Now that we have suceeded in single-item auction setting, can we design awesome mechanisms in more general settings? We consider the following environments.
- In a single-parameter environment, there are n bidders, each bidder i has a private valuation v_i (a value "per unit of the goods"). There is a feasible set $X \subseteq \mathbb{R}^n$ (feasible outcomes)

Single parameter environments

- Now that we have suceeded in single-item auction setting, can we design awesome mechanisms in more general settings? We consider the following environments.
- In a single-parameter environment, there are n bidders, each bidder i has a private valuation v_i (a value "per unit of the goods"). There is a feasible set $X \subseteq \mathbb{R}^n$ (feasible outcomes) containing vectors $x = (x_1, \ldots, x_n)$, where x_i denotes the part of the outcome that bidder i is interested in.

Single parameter environments

- Now that we have suceeded in single-item auction setting, can we design awesome mechanisms in more general settings? We consider the following environments.
- In a single-parameter environment, there are n bidders, each bidder i has a private valuation v_i (a value "per unit of the goods"). There is a feasible set $X \subseteq \mathbb{R}^n$ (feasible outcomes) containing vectors $x = (x_1, \ldots, x_n)$, where x_i denotes the part of the outcome that bidder i is interested in.
- The sealed-bid auction in this environment then proceeds in three steps.

Single parameter environments

- Now that we have suceeded in single-item auction setting, can we design awesome mechanisms in more general settings? We consider the following environments.
- In a single-parameter environment, there are n bidders, each bidder i has a private valuation v_i (a value "per unit of the goods"). There is a feasible set $X \subseteq \mathbb{R}^n$ (feasible outcomes) containing vectors $x = (x_1, \ldots, x_n)$, where x_i denotes the part of the outcome that bidder i is interested in.
- The sealed-bid auction in this environment then proceeds in three steps.
  - Collect bids $b = (b_1, \ldots, b_n)$, where b_i is the bid of bidder i.

Single parameter environments

- Now that we have suceeded in single-item auction setting, can we design awesome mechanisms in more general settings? We consider the following environments.
- In a single-parameter environment, there are n bidders, each bidder i has a private valuation v_i (a value "per unit of the goods"). There is a feasible set $X \subseteq \mathbb{R}^n$ (feasible outcomes) containing vectors $x = (x_1, \ldots, x_n)$, where x_i denotes the part of the outcome that bidder i is interested in.
- The sealed-bid auction in this environment then proceeds in three steps.
  - Collect bids $b = (b_1, \ldots, b_n)$, where b_i is the bid of bidder i.
  - Allocation rule: Choose a feasible outcome allocation $x = x(b)$ from X as a function of the bids b.

Single parameter environments

- Now that we have suceeded in single-item auction setting, can we design awesome mechanisms in more general settings? We consider the following environments.
- In a single-parameter environment, there are n bidders, each bidder i has a private valuation v_i (a value "per unit of the goods"). There is a feasible set $X \subseteq \mathbb{R}^n$ (feasible outcomes) containing vectors $x = (x_1, \ldots, x_n)$, where x_i denotes the part of the outcome that bidder i is interested in.
- The sealed-bid auction in this environment then proceeds in three steps.
  - Collect bids $b = (b_1, \ldots, b_n)$, where b_i is the bid of bidder i.
  - Allocation rule: Choose a feasible outcome allocation $x = x(b)$ from X as a function of the bids b.
  - Payment rule: Choose payments $p(b) = (p_1(b), \ldots, p_n(b)) \in \mathbb{R}^n$ as a function of the bids b.

Single parameter environments

- Now that we have suceeded in single-item auction setting, can we design awesome mechanisms in more general settings? We consider the following environments.
- In a single-parameter environment, there are n bidders, each bidder i has a private valuation v_i (a value "per unit of the goods"). There is a feasible set $X \subseteq \mathbb{R}^n$ (feasible outcomes) containing vectors $x = (x_1, \ldots, x_n)$, where x_i denotes the part of the outcome that bidder i is interested in.
- The sealed-bid auction in this environment then proceeds in three steps.
  - Collect bids $b = (b_1, \ldots, b_n)$, where b_i is the bid of bidder i.
  - Allocation rule: Choose a feasible outcome allocation $x = x(b)$ from X as a function of the bids b.
  - Payment rule: Choose payments $p(b) = (p_1(b), \ldots, p_n(b)) \in \mathbb{R}^n$ as a function of the bids b.
- The pair (x, p) then forms a mechanism.

Single parameter environments

- Now that we have suceeded in single-item auction setting, can we design awesome mechanisms in more general settings? We consider the following environments.
- In a single-parameter environment, there are n bidders, each bidder i has a private valuation v_i (a value "per unit of the goods"). There is a feasible set $X \subseteq \mathbb{R}^n$ (feasible outcomes) containing vectors $x = (x_1, \ldots, x_n)$, where x_i denotes the part of the outcome that bidder i is interested in.
- The sealed-bid auction in this environment then proceeds in three steps.
  - Collect bids $b = (b_1, \ldots, b_n)$, where b_i is the bid of bidder i.
  - Allocation rule: Choose a feasible outcome allocation $x = x(b)$ from X as a function of the bids b.
  - Payment rule: Choose payments $p(b) = (p_1(b), \ldots, p_n(b)) \in \mathbb{R}^n$ as a function of the bids b.
- The pair (x, p) then forms a mechanism.
- The utility $u_i(b)$ of bidder i is $u_i(b) = v_i \cdot x_i(b) - p_i(b)$.

Single parameter environments: remarks

Single parameter environments: remarks

- We consider only payments $p_i(b) \in [0, b_i \cdot x_i(b)]$ for every bidder i and all bids b.

Single parameter environments: remarks

- We consider only payments $p_i(b) \in [0, b_i \cdot x_i(b)]$ for every bidder i and all bids b.
  - Since $p_i(b) \geq 0$, the seller never pays the bidders.

Single parameter environments: remarks

- We consider only payments $p_i(b) \in [0, b_i \cdot x_i(b)]$ for every bidder i and all bids b.
  - Since $p_i(b) \geq 0$, the seller never pays the bidders.
  - The condition $p_i(b) \leq b_i \cdot x_i(b)$ says that we never charge a bidder more than his value b_i per good (that they told us) times the amount $x_i(b)$ of stuff that we gave them.

Single parameter environments: remarks

- We consider only payments $p_i(b) \in [0, b_i \cdot x_i(b)]$ for every bidder i and all bids b.
  - Since $p_i(b) \geq 0$, the seller never pays the bidders.
  - The condition $p_i(b) \leq b_i \cdot x_i(b)$ says that we never charge a bidder more than his value b_i per good (that they told us) times the amount $x_i(b)$ of stuff that we gave them.
  - It ensures that a truthtelling bidder receives non-negative utility.

Single parameter environments: remarks

- We consider only payments $p_i(b) \in [0, b_i \cdot x_i(b)]$ for every bidder i and all bids b.
  - Since $p_i(b) \geq 0$, the seller never pays the bidders.
  - The condition $p_i(b) \leq b_i \cdot x_i(b)$ says that we never charge a bidder more than his value b_i per good (that they told us) times the amount $x_i(b)$ of stuff that we gave them.
  - It ensures that a truthtelling bidder receives non-negative utility.
- The basic dilemma of mechanism design is that the mechanism designer wants to optimize some global objective such as the social surplus $\sum_{i=1}^{n} v_i \cdot x_i(b)$.

Single parameter environments: remarks

- We consider only payments $p_i(b) \in [0, b_i \cdot x_i(b)]$ for every bidder i and all bids b.
  - Since $p_i(b) \geq 0$, the seller never pays the bidders.
  - The condition $p_i(b) \leq b_i \cdot x_i(b)$ says that we never charge a bidder more than his value b_i per good (that they told us) times the amount $x_i(b)$ of stuff that we gave them.
  - It ensures that a truthtelling bidder receives non-negative utility.
- The basic dilemma of mechanism design is that the mechanism designer wants to optimize some global objective such as the social surplus $\sum_{i=1}^{n} v_i \cdot x_i(b)$.
- We now illustrate single-parameter environments with a few specific examples.

Single parameter environments: single-item auctions

Single parameter environments: single-item auctions

- Singe-parameter environments comprise single-item auctions.

Single parameter environments: single-item auctions

- Singe-parameter environments comprise single-item auctions.
- In single-item auctions, n bidders compete for a single item of the seller.

Single parameter environments: single-item auctions

- Singe-parameter environments comprise single-item auctions.
- In single-item auctions, n bidders compete for a single item of the seller.
- Each bidder either gets the item or not, but only one bidder can get it.

Source: https://www.widewalls.ch

Single parameter environments: single-item auctions

- Singe-parameter environments comprise single-item auctions.
- In single-item auctions, n bidders compete for a single item of the seller.
- Each bidder either gets the item or not, but only one bidder can get it.

Source: https://www.widewalls.ch

- This can be modelled by setting $x_i \in \{0, 1\}$ for each bidder i and choosing the feasible set $X = \{(x_1, \ldots, x_n) \in \{0, 1\}^n \colon \sum_{i=1}^{n} x_i \leq 1\}$.

Single parameter environments: single-item auctions

- Singe-parameter environments comprise single-item auctions.
- In single-item auctions, n bidders compete for a single item of the seller.
- Each bidder either gets the item or not, but only one bidder can get it.

Source: https://www.widewalls.ch

- This can be modelled by setting $x_i \in \{0, 1\}$ for each bidder i and choosing the feasible set $X = \{(x_1, \ldots, x_n) \in \{0, 1\}^n \colon \sum_{i=1}^{n} x_i \leq 1\}$.
- The goal is to design the auction so that the bidder with the highest valuation v_i wins.

Single parameter environments: sponsored-search

Single parameter environments: sponsored-search

- Consider the following real-world motivation.

Single parameter environments: sponsored-search

- Consider the following real-world motivation.
- A Web search results page contains a list of organic search results and a list of k sponsored links, which have been paid for by advertisers.

Single parameter environments: sponsored-search

- Consider the following real-world motivation.
- A Web search results page contains a list of organic search results and a list of k sponsored links, which have been paid for by advertisers.
- Every time a search query is typed into a search engine, an auction is run in real-time to decide which sponsored links are shown, in what order, and how they are charged.

Single parameter environments: sponsored-search

- Consider the following real-world motivation.
- A Web search results page contains a list of organic search results and a list of k sponsored links, which have been paid for by advertisers.
- Every time a search query is typed into a search engine, an auction is run in real-time to decide which sponsored links are shown, in what order, and how they are charged.
- The positions for sale are the k “slots” for sponsored links and slots with higher positions on the search page are more valuable than lower ones, since people generally scan the page from top to bottom.

Single parameter environments: sponsored-search

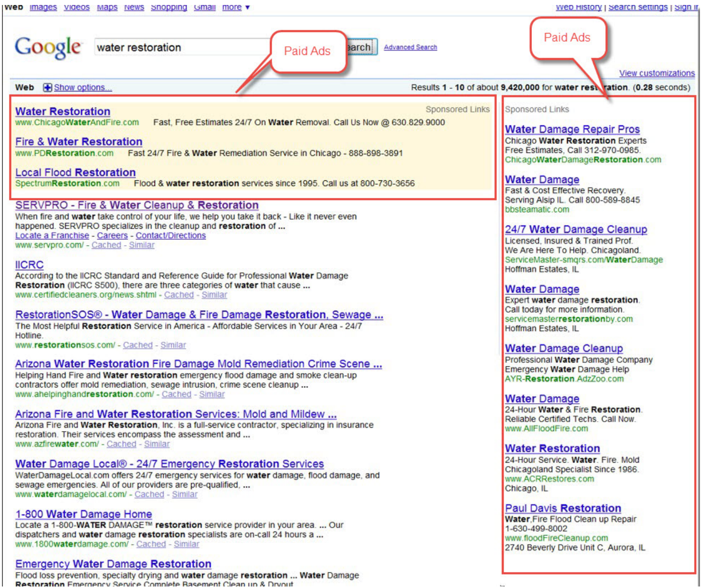
- Consider the following real-world motivation.
- A Web search results page contains a list of organic search results and a list of k sponsored links, which have been paid for by advertisers.
- Every time a search query is typed into a search engine, an auction is run in real-time to decide which sponsored links are shown, in what order, and how they are charged.
- The positions for sale are the k "slots" for sponsored links and slots with higher positions on the search page are more valuable than lower ones, since people generally scan the page from top to bottom.
- The bidders are the advertisers who have a standing bid on the keyword that was searched on.

Single parameter environments: sponsored-search

- Consider the following real-world motivation.
- A Web search results page contains a list of organic search results and a list of k sponsored links, which have been paid for by advertisers.
- Every time a search query is typed into a search engine, an auction is run in real-time to decide which sponsored links are shown, in what order, and how they are charged.
- The positions for sale are the k "slots" for sponsored links and slots with higher positions on the search page are more valuable than lower ones, since people generally scan the page from top to bottom.
- The bidders are the advertisers who have a standing bid on the keyword that was searched on.
  We have two assumptions: first, the more the slot is on the top, the higher the probability α_j that the slot is clicked on, and, second, the click-through rates do not depend on the occupant of the slot.

Sponsored search

Sponsored search



Source: https://proceedinnovative.com

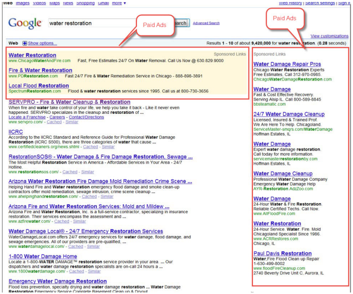
Single parameter environments: sponsored-search

Single parameter environments: sponsored-search

- Formally, we have n bidders competing for k positions.

Single parameter environments: sponsored-search

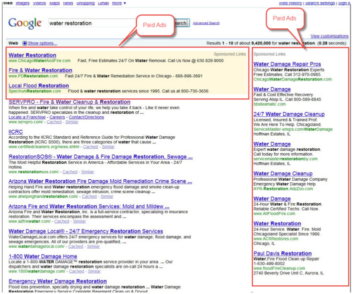
- Formally, we have n bidders competing for k positions.



Source: https://proceedinnovative.com

Single parameter environments: sponsored-search

- Formally, we have n bidders competing for k positions.

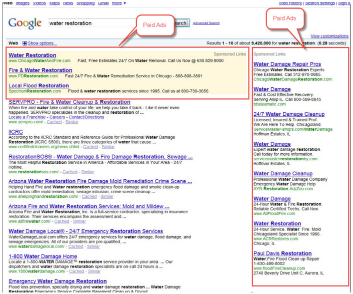


Source: https://proceedinnovative.com

- Each position j has a click-through-rate α_j, where $\alpha_1 > \cdots > \alpha_k > 0$.

Single parameter environments: sponsored-search

- Formally, we have n bidders competing for k positions.

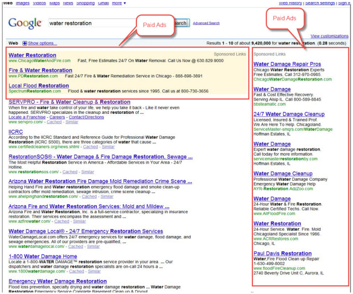


Source: https://proceedinnovative.com

- Each position j has a click-through-rate α_j, where $\alpha_1 > \cdots > \alpha_k > 0$.
- The feasible set X consists of vectors $(x_1, \ldots, x_n)$, where each x_i lies in $\{\alpha_1, \ldots, \alpha_k, 0\}$ and if $x_i = x_j$, then $x_i = 0 = x_j$.

Single parameter environments: sponsored-search

- Formally, we have n bidders competing for k positions.

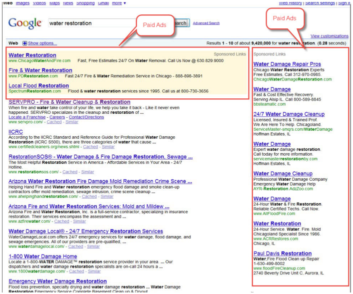


Source: https://proceedinnovative.com

- Each position j has a click-through-rate α_j, where $\alpha_1 > \cdots > \alpha_k > 0$.
- The feasible set X consists of vectors $(x_1, \ldots, x_n)$, where each x_i lies in $\{\alpha_1, \ldots, \alpha_k, 0\}$ and if $x_i = x_j$, then $x_i = 0 = x_j$.
- The value of slot j to bidder i is then $v_i \alpha_j$.

Single parameter environments: sponsored-search

- Formally, we have n bidders competing for k positions.



Source: https://proceedinnovative.com

- Each position j has a click-through-rate α_j, where $\alpha_1 > \cdots > \alpha_k > 0$.
- The feasible set X consists of vectors $(x_1, \ldots, x_n)$, where each x_i lies in $\{\alpha_1, \ldots, \alpha_k, 0\}$ and if $x_i = x_j$, then $x_i = 0 = x_j$.
- The value of slot j to bidder i is then $v_i \alpha_j$.
- The goal is to maximize the social surplus.

Designing awesome mechanisms

Designing awesome mechanisms

- We would like to design sensible mechanisms (x, p) for a given single-parameter environment that are, ideally, awesome.

Designing awesome mechanisms

- We would like to design sensible mechanisms (x, p) for a given single-parameter environment that are, ideally, awesome.
- In particular, we should ensure that (x, p) has the DSIC property.

Designing awesome mechanisms

- We would like to design sensible mechanisms (x, p) for a given single-parameter environment that are, ideally, awesome.
- In particular, we should ensure that (x, p) has the DSIC property. Thus, we want to identify allocation rules x for which we can find payment rules p such that (x, p) is DSIC.

Designing awesome mechanisms

- We would like to design sensible mechanisms (x, p) for a given single-parameter environment that are, ideally, awesome.
- In particular, we should ensure that (x, p) has the DSIC property. Thus, we want to identify allocation rules x for which we can find payment rules p such that (x, p) is DSIC.
- An allocation rule x for a single-parameter environment is implementable if there is a payment rule p such that (x, p) is DSIC.

Designing awesome mechanisms

- We would like to design sensible mechanisms (x, p) for a given single-parameter environment that are, ideally, awesome.
- In particular, we should ensure that (x, p) has the DSIC property. Thus, we want to identify allocation rules x for which we can find payment rules p such that (x, p) is DSIC.
- An allocation rule x for a single-parameter environment is implementable if there is a payment rule p such that (x, p) is DSIC.
    - The allocation rule "give the item to the bidder with the highest bid" is implementable in the case of single-item auctions, as the second-price rule provides DSIC mechanism.

Designing awesome mechanisms

- We would like to design sensible mechanisms (x, p) for a given single-parameter environment that are, ideally, awesome.
- In particular, we should ensure that (x, p) has the DSIC property. Thus, we want to identify allocation rules x for which we can find payment rules p such that (x, p) is DSIC.
- An allocation rule x for a single-parameter environment is implementable if there is a payment rule p such that (x, p) is DSIC.
    - The allocation rule "give the item to the bidder with the highest bid" is implementable in the case of single-item auctions, as the second-price rule provides DSIC mechanism.
    - The situation is much less clear for the allocation rule "give the item to the bidder with the second highest bid".

Designing awesome mechanisms

- We would like to design sensible mechanisms (x, p) for a given single-parameter environment that are, ideally, awesome.
- In particular, we should ensure that (x, p) has the DSIC property. Thus, we want to identify allocation rules x for which we can find payment rules p such that (x, p) is DSIC.
- An allocation rule x for a single-parameter environment is implementable if there is a payment rule p such that (x, p) is DSIC.
    - The allocation rule "give the item to the bidder with the highest bid" is implementable in the case of single-item auctions, as the second-price rule provides DSIC mechanism.
    - The situation is much less clear for the allocation rule "give the item to the bidder with the second highest bid".
- An allocation rule x is monotone if, for every bidder i and all bids b_{-i} of the other bidders, the allocation $x_i(z; b_{-i})$ to i is nondecreasing in his bid z.

Designing awesome mechanisms

- We would like to design sensible mechanisms (x, p) for a given single-parameter environment that are, ideally, awesome.
- In particular, we should ensure that (x, p) has the DSIC property. Thus, we want to identify allocation rules x for which we can find payment rules p such that (x, p) is DSIC.
- An allocation rule x for a single-parameter environment is implementable if there is a payment rule p such that (x, p) is DSIC.
    - The allocation rule "give the item to the bidder with the highest bid" is implementable in the case of single-item auctions, as the second-price rule provides DSIC mechanism.
    - The situation is much less clear for the allocation rule "give the item to the bidder with the second highest bid".
- An allocation rule x is monotone if, for every bidder i and all bids b_{-i} of the other bidders, the allocation $x_i(z; b_{-i})$ to i is nondecreasing in his bid z.
    - The first rule above is monotone while the other one is not.

Myerson's lemma

Myerson's lemma

- These two notions coincide!

Myerson's lemma

- These two notions coincide! Follows from Myerson's lemma, a powerful tool for designing DSIC mechanisms.

Myerson's lemma

- These two notions coincide! Follows from Myerson's lemma, a powerful tool for designing DSIC mechanisms.

Figure: Roger Myerson (born 1951) receiving a Nobel prize in economics.

Sources: https://en.wikipedia.org and https://twitter.com

Myerson's lemma

Myerson's lemma

Myerson's lemma (Theorem 3.8)

In a single-parameter environment, the following three claims hold.

Myerson's lemma

Myerson's lemma (Theorem 3.8)

In a single-parameter environment, the following three claims hold.

(*a*) An allocation rule is implementable if and only if it is monotone.

Myerson's lemma

Myerson's lemma (Theorem 3.8)

In a single-parameter environment, the following three claims hold.

(*a*) An allocation rule is implementable if and only if it is monotone.

(*b*) If an allocation rule x is monotone, then there exists a unique payment rule p such that the mechanism (x, p) is DSIC (assuming that $b_i = 0$ implies $p_i(b) = 0$).

Myerson's lemma

Myerson's lemma (Theorem 3.8)

In a single-parameter environment, the following three claims hold.

(a) An allocation rule is implementable if and only if it is monotone.

(b) If an allocation rule x is monotone, then there exists a unique payment rule p such that the mechanism (x, p) is DSIC (assuming that $b_i = 0$ implies $p_i(b) = 0$).

(c) The payment rule p is given by the following explicit formula

$$p_i(b_i; b_{-i}) = \int_0^{b_i} z \cdot \frac{\mathrm{d}}{\mathrm{d}z} x_i(z; b_{-i}) \, \mathrm{d}z$$

for every $i \in \{1, \ldots, n\}$.

Myerson's lemma

Myerson's lemma (Theorem 3.8)

In a single-parameter environment, the following three claims hold.

(a) An allocation rule is implementable if and only if it is monotone.

(b) If an allocation rule x is monotone, then there exists a unique payment rule p such that the mechanism (x, p) is DSIC (assuming that $b_i = 0$ implies $p_i(b) = 0$).

(c) The payment rule p is given by the following explicit formula

$$p_i(b_i; b_{-i}) = \int_0^{b_i} z \cdot \frac{\mathrm{d}}{\mathrm{d}z} x_i(z; b_{-i}) \, \mathrm{d}z$$

for every $i \in \{1, \ldots, n\}$.

- We will see the proof next week, now we show some applications.

Appliactions of Myerson's lemma I

Appliactions of Myerson's lemma I

- We start with single-item auctions.

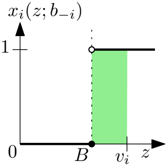
Appliactions of Myerson's lemma I

- We start with single-item auctions.
- Let bidder i and bids b_{-i} of the other bidders be fixed and set $B = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$.

Appliactions of Myerson's lemma I

- We start with single-item auctions.
- Let bidder i and bids b_{-i} of the other bidders be fixed and set $B = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$. If we allocate the item to the highest bidder, then the allocation function $x_i(\,\cdot\,; b_{-i})$ is 0 up to B and 1 thereafter.

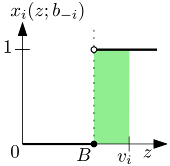
Bidder i is the highest bidder:



Appliactions of Myerson's lemma I

- We start with single-item auctions.
- Let bidder i and bids b_{-i} of the other bidders be fixed and set $B = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$. If we allocate the item to the highest bidder, then the allocation function $x_i(\,\cdot\,; b_{-i})$ is 0 up to B and 1 thereafter.

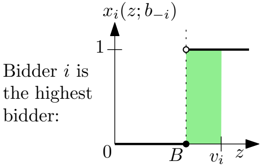
Bidder i is the highest bidder:



Clearly, this allocation rule is monotone.

Appliactions of Myerson's lemma I

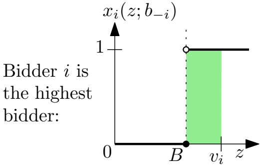
- We start with single-item auctions.
- Let bidder i and bids b_{-i} of the other bidders be fixed and set $B = \max_{j \in \{1,\ldots,n\} \setminus \{i\}} b_j$. If we allocate the item to the highest bidder, then the allocation function $x_i(\,\cdot\,; b_{-i})$ is 0 up to B and 1 thereafter.



  Clearly, this allocation rule is monotone.
- If i is the highest bidder, then $b_i > B$ and the (unique) payment formula from Myerson's lemma becomes $p_i(b_i; b_{-i}) = B$ and the utility of i is $v_i \cdot x_i(b_i; b_{-i}) - B = v_i - B$.

Appliactions of Myerson's lemma I

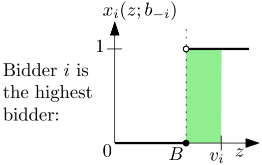
- We start with single-item auctions.
- Let bidder i and bids b_{-i} of the other bidders be fixed and set $B = \max_{j \in \{1,\ldots,n\} \setminus \{i\}} b_j$. If we allocate the item to the highest bidder, then the allocation function $x_i(\,\cdot\,; b_{-i})$ is 0 up to B and 1 thereafter.



  Clearly, this allocation rule is monotone.
- If i is the highest bidder, then $b_i > B$ and the (unique) payment formula from Myerson's lemma becomes $p_i(b_i; b_{-i}) = B$ and the utility of i is $v_i \cdot x_i(b_i; b_{-i}) - B = v_i - B$. Otherwise, $b_i \leq B$ and the payment function and the utility of i is zero.

Appliactions of Myerson's lemma I

- We start with single-item auctions.
- Let bidder i and bids b_{-i} of the other bidders be fixed and set $B = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$. If we allocate the item to the highest bidder, then the allocation function $x_i(\,\cdot\,; b_{-i})$ is 0 up to B and 1 thereafter.

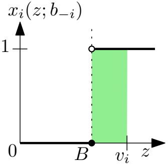


  Clearly, this allocation rule is monotone.
- If i is the highest bidder, then $b_i > B$ and the (unique) payment formula from Myerson's lemma becomes $p_i(b_i; b_{-i}) = B$ and the utility of i is $v_i \cdot x_i(b_i; b_{-i}) - B = v_i - B$. Otherwise, $b_i \leq B$ and the payment function and the utility of i is zero.
- If $v_i > B$, then his utility is positive and the utility of all other bidders is zero.

Appliactions of Myerson's lemma I

- We start with single-item auctions.
- Let bidder i and bids b_{-i} of the other bidders be fixed and set $B = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$. If we allocate the item to the highest bidder, then the allocation function $x_i(\,\cdot\,; b_{-i})$ is 0 up to B and 1 thereafter.

Bidder i is the highest bidder:



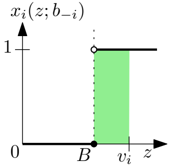
Clearly, this allocation rule is monotone.

- If i is the highest bidder, then $b_i > B$ and the (unique) payment formula from Myerson's lemma becomes $p_i(b_i; b_{-i}) = B$ and the utility of i is $v_i \cdot x_i(b_i; b_{-i}) - B = v_i - B$. Otherwise, $b_i \leq B$ and the payment function and the utility of i is zero.
- If $v_i > B$, then his utility is positive and the utility of all other bidders is zero. It follows from the form of p_i that the utility of i is maximized when $v_i = b_i$.

Appliactions of Myerson's lemma I

- We start with single-item auctions.
- Let bidder i and bids b_{-i} of the other bidders be fixed and set $B = \max_{j \in \{1,\dots,n\} \setminus \{i\}} b_j$. If we allocate the item to the highest bidder, then the allocation function $x_i(\,\cdot\,; b_{-i})$ is 0 up to B and 1 thereafter.

Bidder i is the highest bidder:



Clearly, this allocation rule is monotone.

- If i is the highest bidder, then $b_i > B$ and the (unique) payment formula from Myerson's lemma becomes $p_i(b_i; b_{-i}) = B$ and the utility of i is $v_i \cdot x_i(b_i; b_{-i}) - B = v_i - B$. Otherwise, $b_i \le B$ and the payment function and the utility of i is zero.
- If $v_i > B$, then his utility is positive and the utility of all other bidders is zero. It follows from the form of p_i that the utility of i is maximized when $v_i = b_i$. Altogether, we obtain the second-price payment rule.

Appliactions of Myerson's lemma II

Appliactions of Myerson's lemma II

- We continue with sponsored-search auctions.

Appliactions of Myerson's lemma II

- We continue with sponsored-search auctions.
- Let x be the allocation rule that assigns the ith best slot to the ith highest bidder.

Appliactions of Myerson's lemma II

- We continue with sponsored-search auctions.
- Let x be the allocation rule that assigns the ith best slot to the ith highest bidder. The rule x is then monotone, as one can easily verify, and, assuming truthful bids, x is also maximizing social surplus.

Appliactions of Myerson's lemma II

- We continue with sponsored-search auctions.
- Let x be the allocation rule that assigns the ith best slot to the ith highest bidder. The rule x is then monotone, as one can easily verify, and, assuming truthful bids, x is also maximizing social surplus.
- By Myerson's lemma, there is a unique and explicit formula for a payment rule p such that the mechanism (x, p) is DSIC.

Appliactions of Myerson's lemma II

- We continue with sponsored-search auctions.
- Let x be the allocation rule that assigns the ith best slot to the ith highest bidder. The rule x is then monotone, as one can easily verify, and, assuming truthful bids, x is also maximizing social surplus.
- By Myerson's lemma, there is a unique and explicit formula for a payment rule p such that the mechanism (x, p) is DSIC.
- Assume without loss of generality that bidder i bids the ith highest bid, that is, $b_1 \geq \cdots \geq b_n$.

Appliactions of Myerson's lemma II

- We continue with sponsored-search auctions.
- Let x be the allocation rule that assigns the ith best slot to the ith highest bidder. The rule x is then monotone, as one can easily verify, and, assuming truthful bids, x is also maximizing social surplus.
- By Myerson's lemma, there is a unique and explicit formula for a payment rule p such that the mechanism (x, p) is DSIC.
- Assume without loss of generality that bidder i bids the ith highest bid, that is, $b_1 \geq \cdots \geq b_n$. Consider bidder 1.

Appliactions of Myerson's lemma II

- We continue with sponsored-search auctions.
- Let x be the allocation rule that assigns the ith best slot to the ith highest bidder. The rule x is then monotone, as one can easily verify, and, assuming truthful bids, x is also maximizing social surplus.
- By Myerson's lemma, there is a unique and explicit formula for a payment rule p such that the mechanism (x, p) is DSIC.
- Assume without loss of generality that bidder i bids the ith highest bid, that is, $b_1 \geq \cdots \geq b_n$. Consider bidder 1. Imagine that he increases his bid z from 0 to b_1, while other bids are fixed.

Appliactions of Myerson's lemma II

- We continue with sponsored-search auctions.
- Let x be the allocation rule that assigns the ith best slot to the ith highest bidder. The rule x is then monotone, as one can easily verify, and, assuming truthful bids, x is also maximizing social surplus.
- By Myerson's lemma, there is a unique and explicit formula for a payment rule p such that the mechanism (x, p) is DSIC.
- Assume without loss of generality that bidder i bids the ith highest bid, that is, $b_1 \geq \cdots \geq b_n$. Consider bidder 1. Imagine that he increases his bid z from 0 to b_1, while other bids are fixed. The allocation function $x_1(z; b_{-1})$ increases from 0 to α_1 as z increases from 0 to b_1, with a jump of $\alpha_j - \alpha_{j+1}$ at the point where z becomes the jth highest bid in the profile $(z; b_{-1})$, namely b_{j+1}.

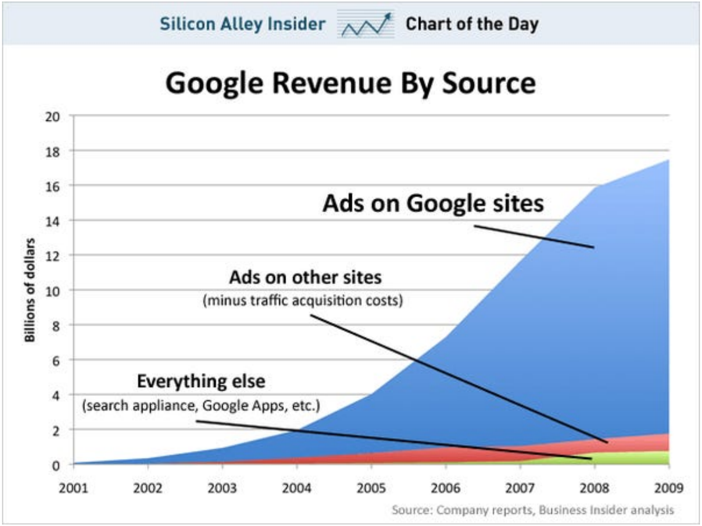
Appliactions of Myerson's lemma II

- We continue with sponsored-search auctions.
- Let x be the allocation rule that assigns the ith best slot to the ith highest bidder. The rule x is then monotone, as one can easily verify, and, assuming truthful bids, x is also maximizing social surplus.
- By Myerson's lemma, there is a unique and explicit formula for a payment rule p such that the mechanism (x, p) is DSIC.
- Assume without loss of generality that bidder i bids the ith highest bid, that is, $b_1 \geq \cdots \geq b_n$. Consider bidder 1. Imagine that he increases his bid z from 0 to b_1, while other bids are fixed. The allocation function $x_1(z; b_{-1})$ increases from 0 to α_1 as z increases from 0 to b_1, with a jump of $\alpha_j - \alpha_{j+1}$ at the point where z becomes the jth highest bid in the profile $(z; b_{-1})$, namely b_{j+1}.
- In general, for ith highest bidder, Myerson's lemma gives the payment formula (for $\alpha_{k+1} = 0$)

$$p_i(b) = \sum_{j=i}^{k} b_{j+1}(\alpha_j - \alpha_{j+1}).$$

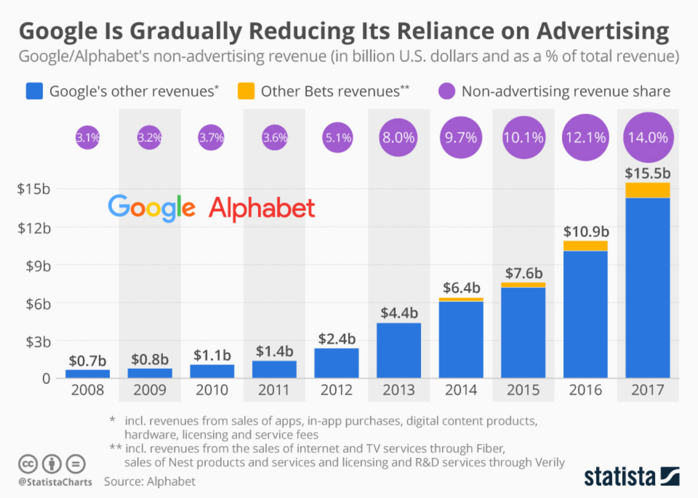
- Sponsored-search auctions were responsible for 98% revenue of Google in 2006.

- Sponsored-search auctions were responsible for 98% revenue of Google in 2006.



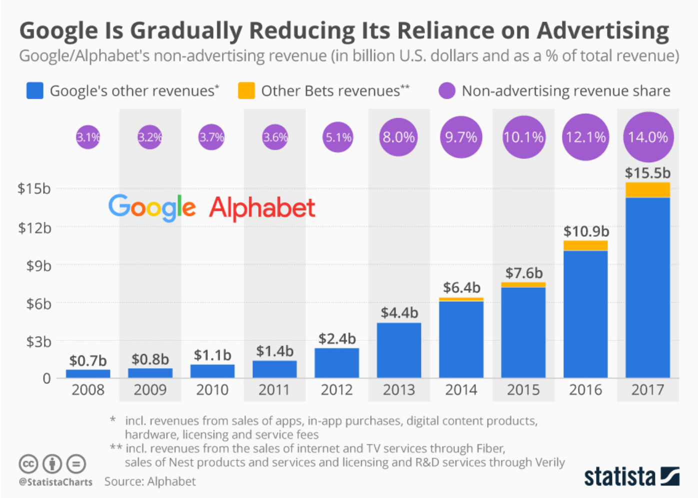
Source: https://businessinsider.com

- Sponsored-search auctions were responsible for 98% revenue of Google in 2006.



Source: https://businessinsider.com

- Sponsored-search auctions were responsible for 98% revenue of Google in 2006.



Source: https://businessinsider.com

Thank you for your attention.