CS 396: Online Markets

Lecture 1: Ride Sharing

Topics:

- ride sharing problem
- algorithms, online algorithms, mechanisms
- first price auction
- ascending auction
- second price auction
- secretary algorithm
- secretary pricing

Exercise: Elevator Problem

Setup:

- two elevators: floors 0 and 7
- three riders:
  - Alice from 1 to 4
  - Bob from 5 to 6
  - Charlie from 3 to 2
- cost 1 to move elevator each floor.

Find plan for elevators to minimize total cost

Example: Ride Sharing

"a.k.a., the Uber problem"

input:

- k drivers at initial locations $(d_1, \ldots, d_k)$
- n rider request (one at a time):
  - origin s_i
  - destination f_i
- driving cost $|f_i - s_i|$

output:

- choice of driver to pick up each rider
- goal: minimize total costs.

Note: any driver pays cost $|f_i - s_i|$,

$\Rightarrow$ minimizing total cost = minimize pickup cost.

History: original CS motivation reading hard drive with multiple read heads

Algorithm design:

- efficiently compute the optimal assignment.

Two twists:

- drivers need to be incentivized to give rides.

  $\Rightarrow$ mechanism design

- riders are not known in advance.

  $\Rightarrow$ online algorithms

A.k.a.: online ride share problem is "k-server problem"

Allocating a Single-item

input:

- n buyers with values $(v_i[1], \ldots, v_i[n])$

output:

- winner
- goal: maximize value of winner.

Four paradigms:

- algorithms.
- mechanisms.
- online algorithms.
- online mechanisms.

Algorithms

Alg: Maximum Value

1. loop over buyers.
2. keep track of buyer and value that is "maximum so far".
3. output "maximum so far" buyer.

Mechanisms

"buyers are strategic"

Mech 0:

1. ask buyer to report values
2. run maximum value algorithm

Q: how would you bid?

Q: what happens?

A: arbitrary outcome.

Mech 1: first-price auction

1. ask buyers to report values.
2. winner is highest bidder.
3. winner pays their bid.

Q: how would you bid?

Q: what happens?

A: later in course.

Exercise: Place Your Bids

Setup:

- you are bidding in two auctions A and B.
- your opponents' value are $U[0, 100]$
- your values v_A and v_B are $U[0, 100]$
- your utility is value minus bid if you win, zero otherwise.

Given your values, determine bids to place in the auctions.

Mech 2: ascending auction

1. price ascends
2. until second to last bidder drops
3. remaining bidder wins, pays this price.

Q: how would you bid?

A: drop when price > value

Q: what happens?

A:

- highest valued agent wins
- pays second highest value

Thm: ascending price auction maximizes social welfare

Proof:

social surplus

= total utility of all participants

= seller + winner + losers

$= v_{(2)} + (v_{(1)} - v_{(2)}) + 0$

$= v_{(1)}$

Challenge: generalization to complex environments like ride sharing.

Idea: [Vickrey '61; Nobel Prize] simulate ascending auction with sealed bids.

Mech 3: second-price auction

1. ask buyers to report values.
2. winner is highest bidder.
3. winner pays second-highest bid.

Q: how would you bid?

A: bid your value.

Q: what happens?

A: same as ascending auction.

Thm: bid = value is dominant strategy in second-price auction

Proof:

Consider bidder i:

- $\hat{v}_i = \max_{j \neq i} b_j$
- if $b_i > \hat{v}_i$:
  - i wins and pays $\hat{v}_i$
  - $u_i = v_i - \hat{v}_i$
- if $b_i < \hat{v}_i$:
  - i loses and pays 0
  - $u_i = 0$
- consider cases
  - $v_i > \hat{v}_i$
  - $v_i < \hat{v}_i$

  [PICTURE]
- $b_i = v_i$ is "best" in both cases.
- thus, dominant strategy.

Online Algorithms

"buyers show up one at a time, must make decision for each buyer before next buyer arrives"

Note: no good algorithm without more information.

Example:

- $\mathbf{v} = (1, 0, \ldots, 0)$
- $\mathbf{v} = (1, 0, \ldots, 0, 100)$

Assumption: values are drawn uniformly between 0 and 100

A.k.a.: online allocation with random order is "secretary problem"

Example: $n = 3$

3 2 1	3 1 2	1 3 2	2 3 1	2 1 3	1 2 3
(a)	(a)	(b)	(b)	(b)	

Two algs for example:

(a) give item to buyer i for some i

$\Rightarrow$ **Pr**[success] $= 1/3$

(b) look at 1st, condition choice of 2nd or 3rd.

- if 2nd greater than 1st, give to 2nd
- else, give to 3rd.

  $\Rightarrow$ **Pr**[success] $= 1/2$

Algorithm: Secretary Alg

1. solicit bids from k buyers, but do not sell.
2. give item to first remaining buyer with value greater than any of first k.

Lemma: For $k = n/2$ alg is selects highest value w.p. 1/4.

Proof:

- select highest when 2nd best in first half and 1st best in second half.
- Recall "conditional probability":

  $\mathbf{Pr}[A \,\&\, B] = \mathbf{Pr}[A \mid B]\mathbf{Pr}[B]$.
- **Pr**[2nd best in first half] $= 1/2$
- **Pr**[1st best in second half | 2nd best in first half] $= \frac{n/2}{n-1} \geq 1/2$

$\Rightarrow$ **Pr**[select highest]

$\geq$ **Pr**[2nd in 1st 1/2]**Pr**[1st in 2nd 1/2 | 2nd in 1st 1/2] $\geq 1/4$.

Q: what is best k?

Thm: for $k = n/e$ alg selects highest buyer with probability $1/e$

(and this is best possible; recall $e = 2.718$)

Online Mechanisms

Mech: Secretary Pricing

1. solicit bids from k buyers, but do not sell.
2. to remaining buyers, offer price equal to max of first k values, until sold.

Thm: bid = value is dominant strategy.

Thm: $k = n/e$ sells to highest valued buyer with probability $1/e$

Summary

Four paradigms for allocating a single-item

- algorithms.
- mechanisms.
- online algorithms.
- online mechanisms.
