Intro to Ad-Click Simulator

Castle Lab @ Princeton U.

Apr 11, 2018

Donghun Lee

Department of
Computer Science

Princeton University

Weidong Han

Department of
Operations Research and
Financial Engineering

Princeton University

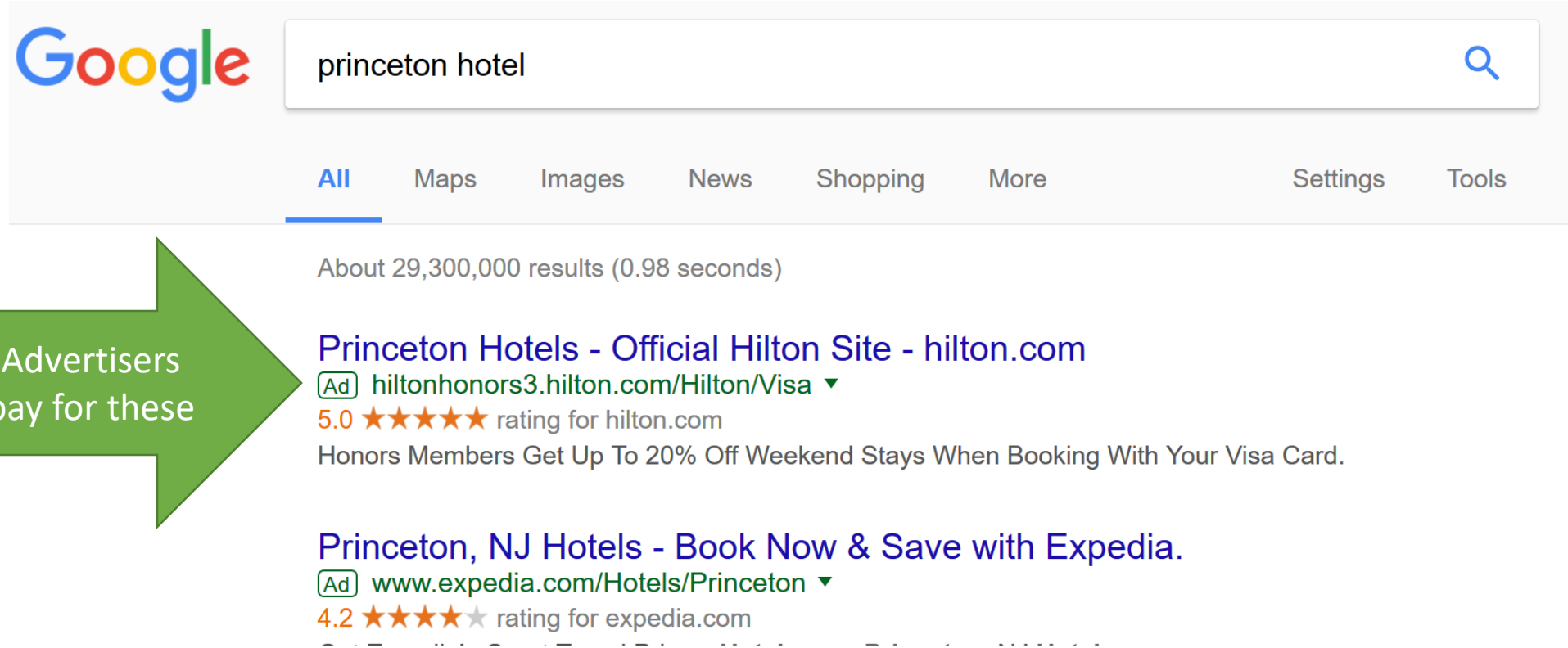
PRINCETON
UNIVERSITY

1. Sponsored Search Auction
2. Simulator Description
3. Simulator Code Structure
4. Dive to Code

Sponsored Search Auction

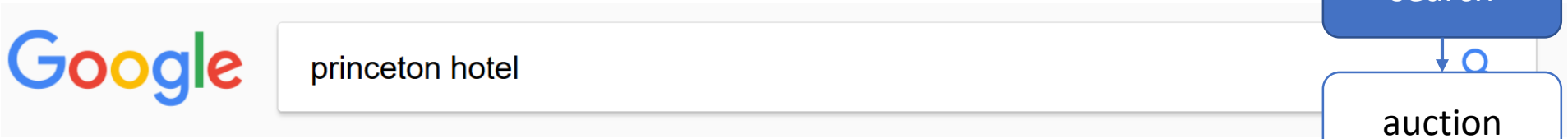
Sponsored Search Auction (SSA)

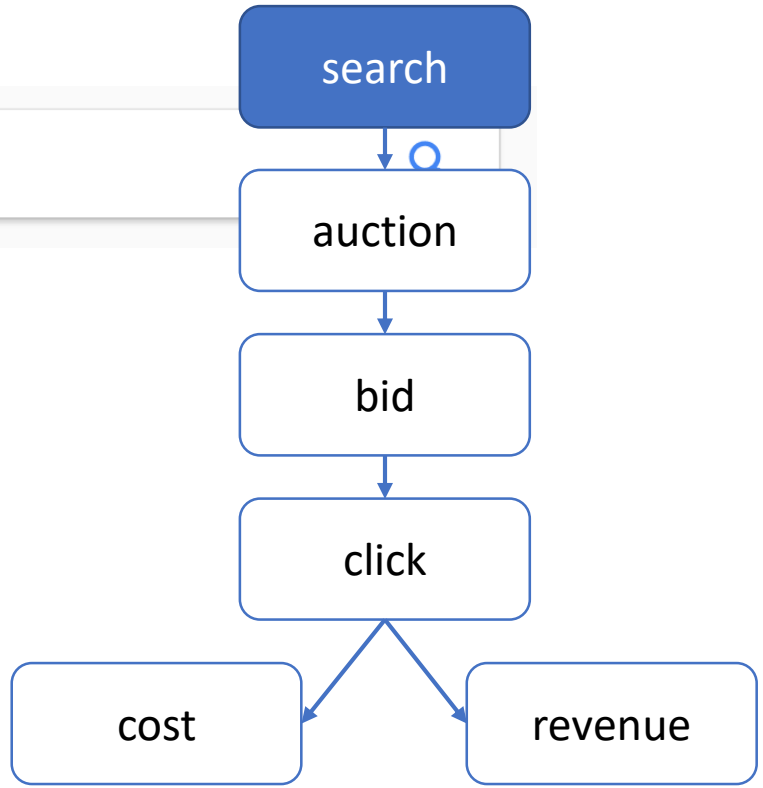
- Sponsored search auction result



How to Bid in SSA

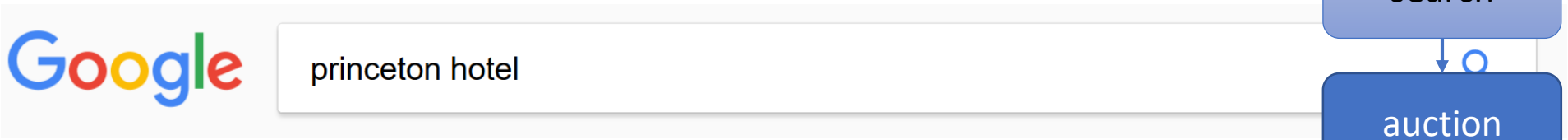
- When someone sends in a search query



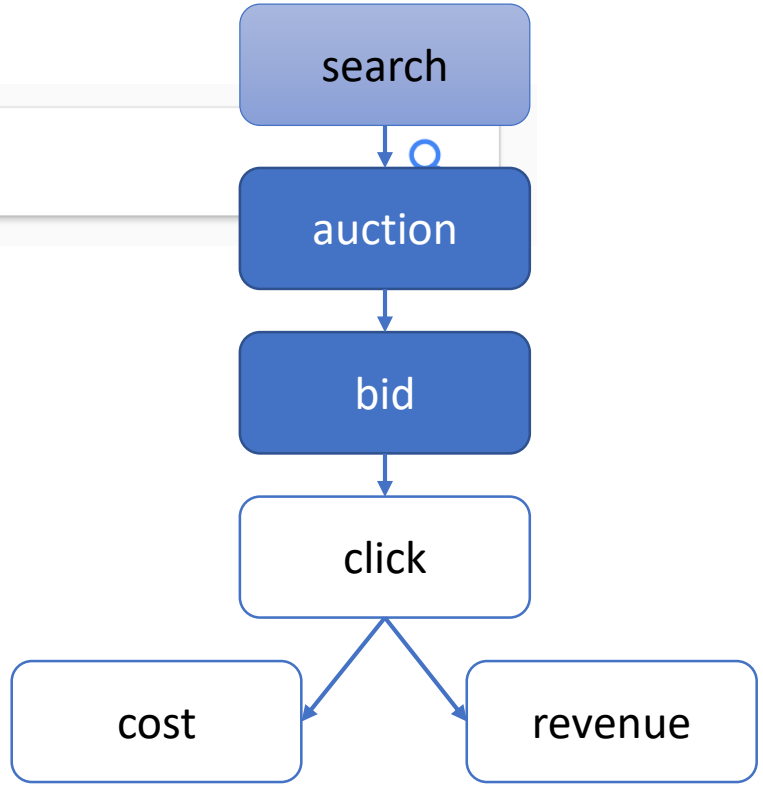


How to Bid in SSA

- Auction happens behind the curtain

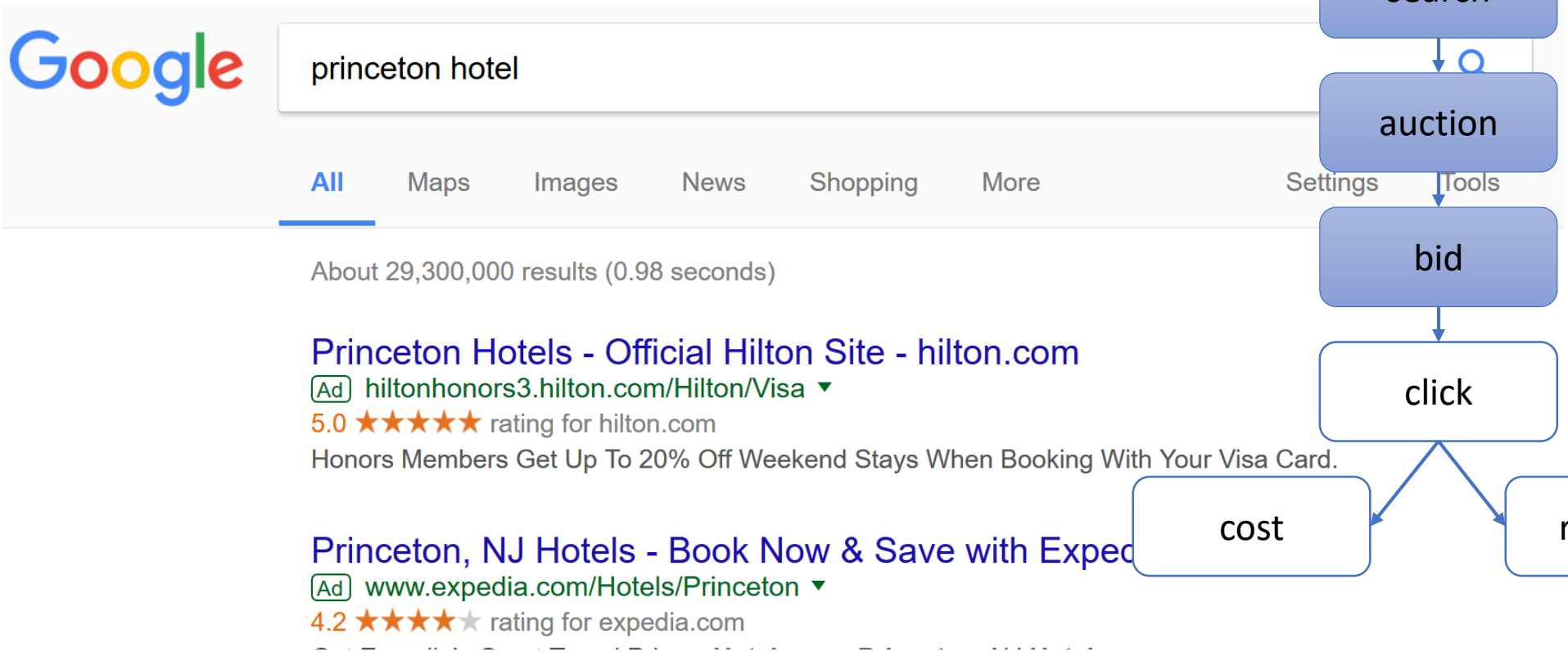


- Advertisers bid to get their ads placed



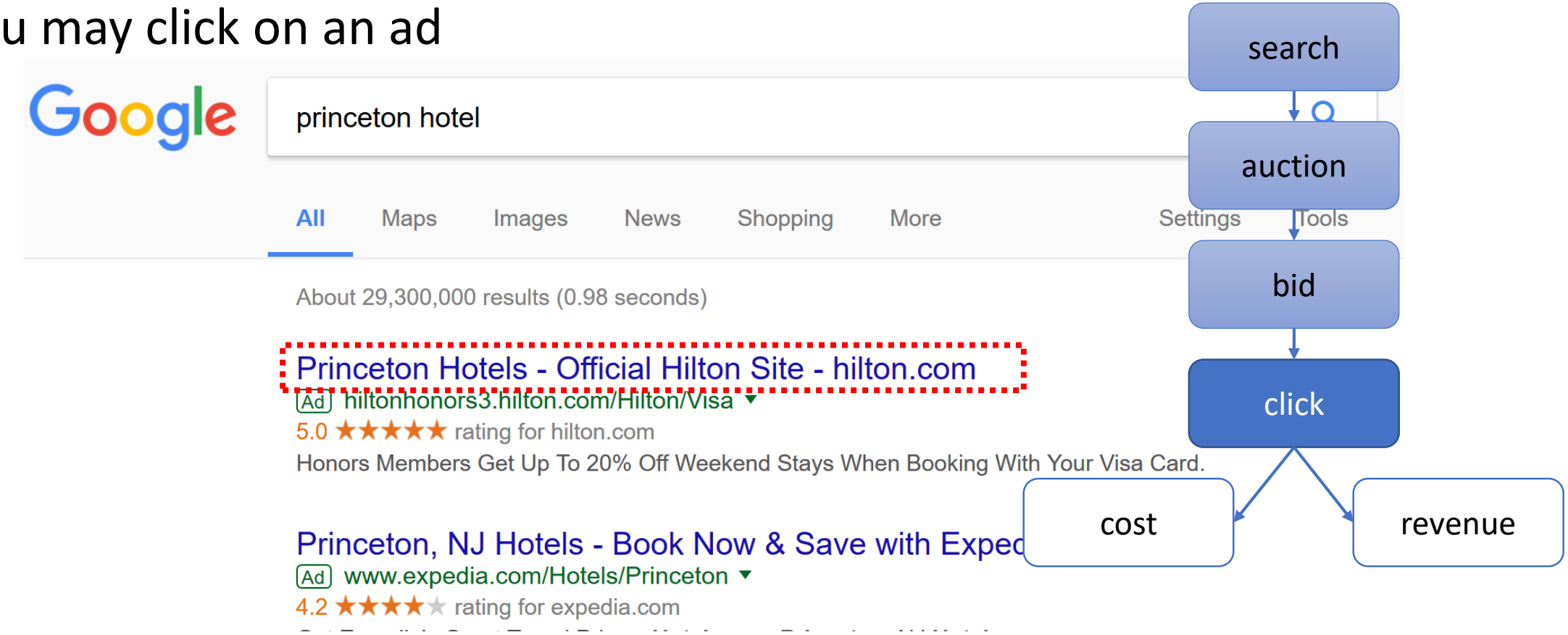
How to Bid in SSA

- The ads from the winners appear in the result



How to Bid in SSA

- You may click on an ad

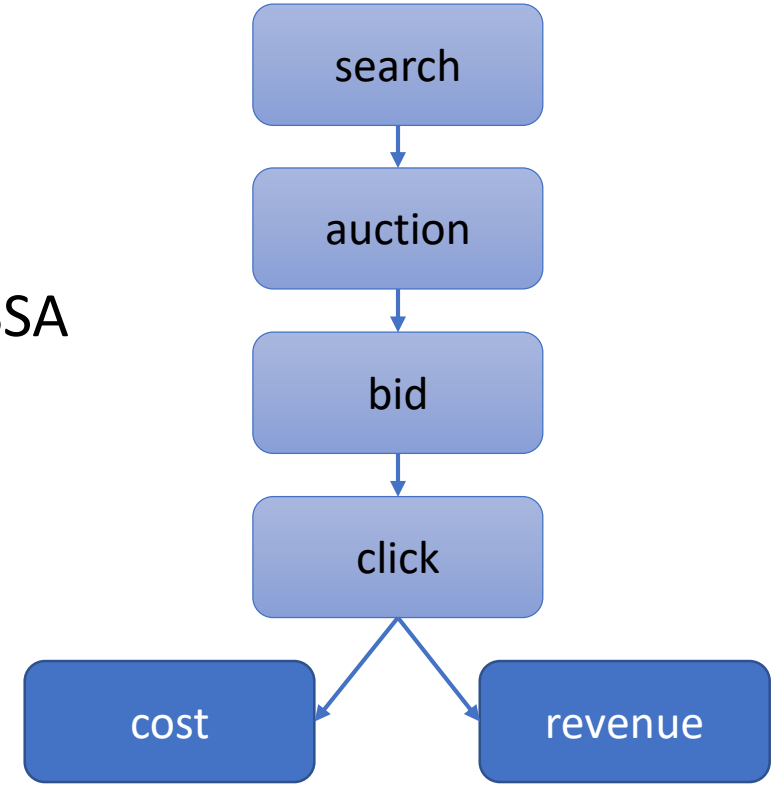


How to Bid in SSA

- Then that advertiser pay for your click
- You may generate some revenue for that advertiser
- Profit of a bid = [revenue] – [cost]

Learn How to Bid in SSA

- Learning with a goal: maximize profit
- Find the policy
  - that maximizes profit from bidding (to advertise) in SSA



Simulator Description

Discrete-time Simulator

- Time runs in discrete steps
  - Simulator's 1 step = real-world 1 hour
  - $t = 0, 1, \ldots, T$

- At time t,
  - Each policy will submit their bid
  - Bid is valid for $[t, t+1)$, fixed during that time step
  - May change bids at $t+1$

Second Price Auction

- Highest bidder wins the auction
- Highest bidder gets to put on the ad
- When the ad is clicked, the ad owner pays the second highest bid

- Example
  - Bid: $3, $5, $6, $10
  - Winner: D
  - Per click, D pays $6

Many Auctions at a Timestep

- Many searches can happen in 1 hour
- Many auctions can happen in 1 timestep

- Note: your bid is fixed for each timestep

- Tie-breaking? Fair deal.
  - Highest bid ties will be randomly broken
  - If all bids tie, then the winners will pay the bid price (not 2nd highest bid)

Not All Auctions Are the Same

- Each auction has different set of attributes
  - e.g. my search = age 30-39, gender M, platform desktop, location NJ+suburb

- Attribute Space (tentative)
  - Age: 0-19, 20-29, 30-39, 40-49, 50-59, 60+
  - Gender: M, F
  - Platform: Mobile, Tablet, Desktop
  - Location: Geo+Type

Attribute-specific Bidding

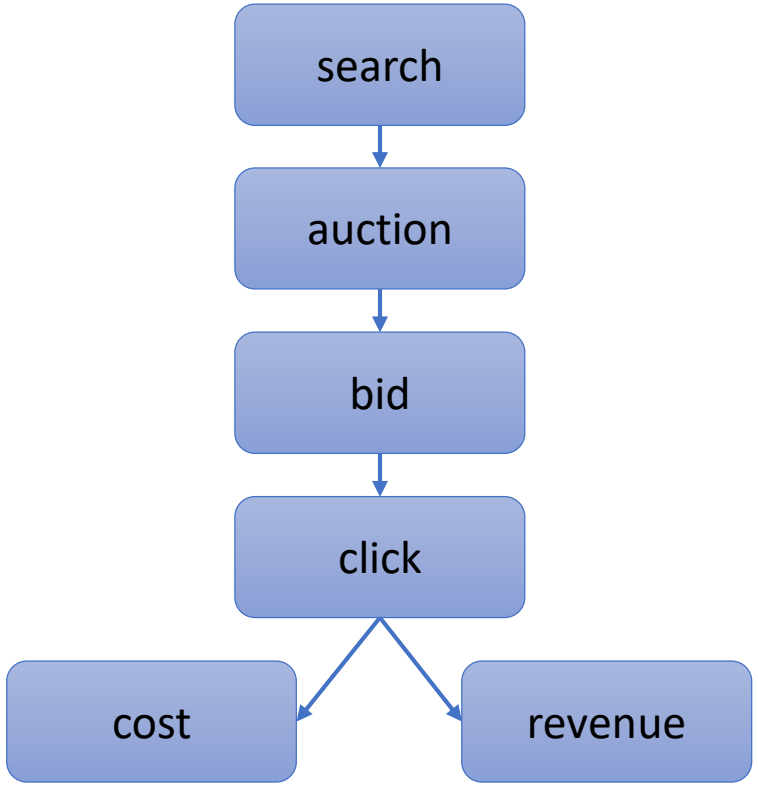
- For each attribute, you may bid differently
  - bid() function in your policy

Learning to Bid

- Bidding policy can learn from outcome
  - learn() function in your policy
- Information is provided every iteration
  - Your policy can learn and change bidding strategy for the next iteration

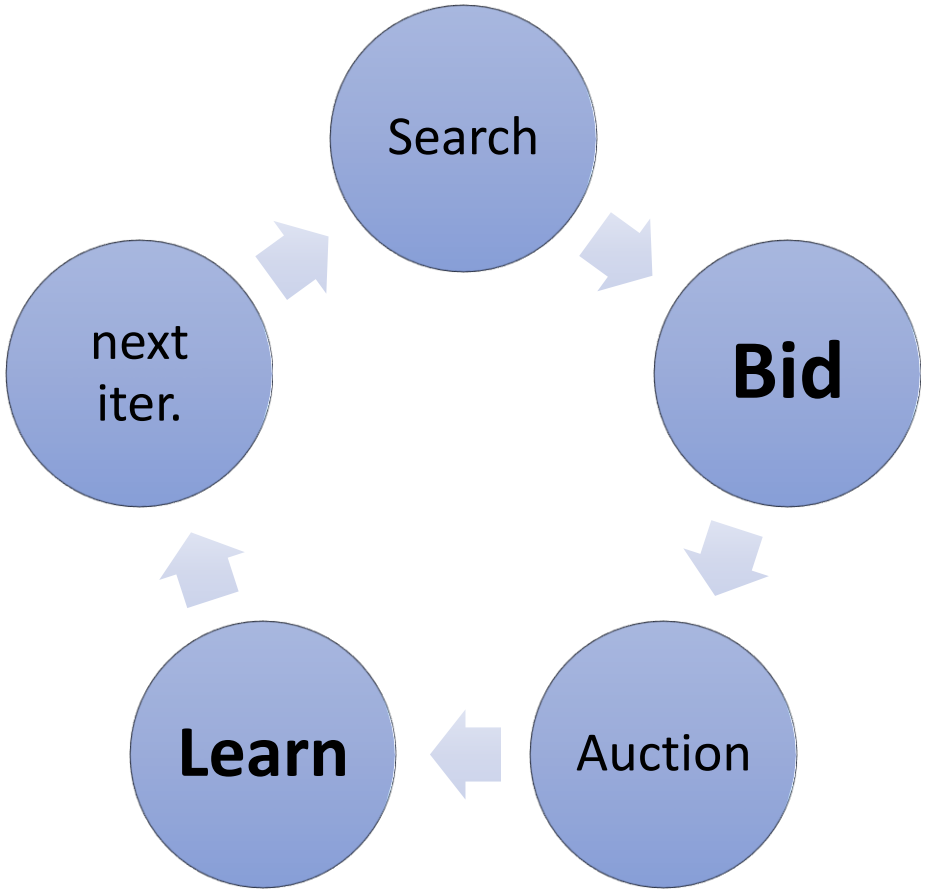
Simulator Operation

- For each auction
  - Get bids from each policies
  - Determine winner
  - Randomly determine click counts
  - Compute cost per click
  - Randomly determine conversion & revenue
  - Report individualized results to each policies



Simulator Operation: Policy's View

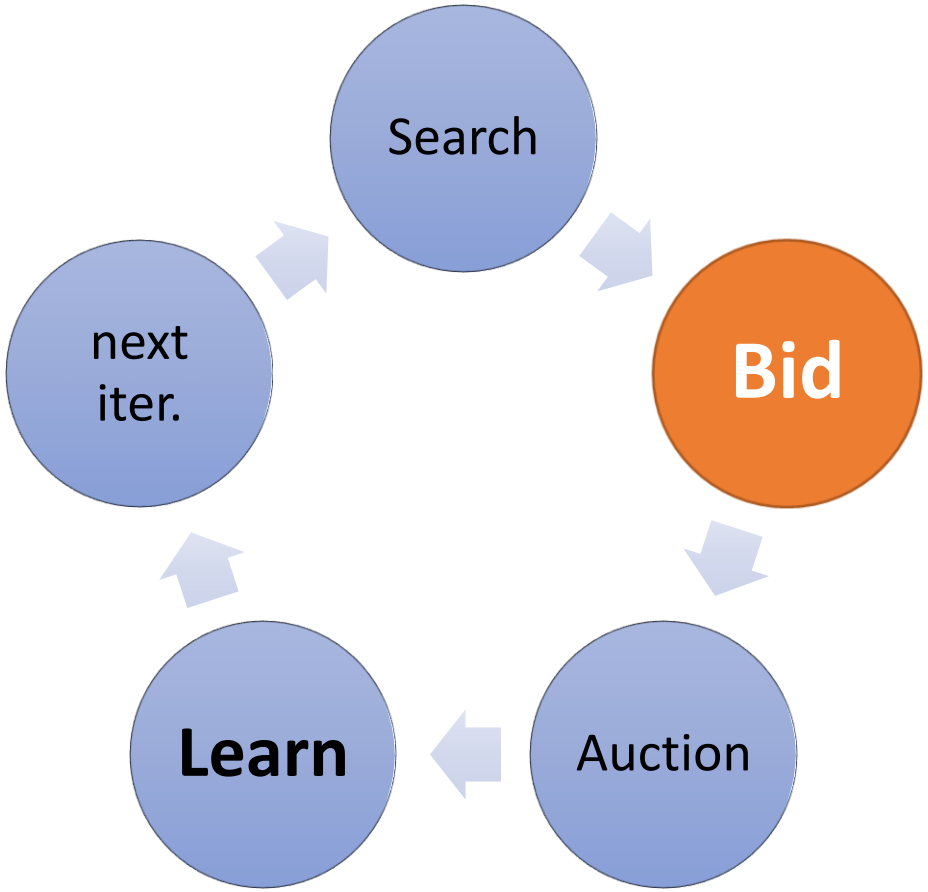
- For each iteration, each policy
  - Sends in bids
  - Gets the results & learn



Code Structure

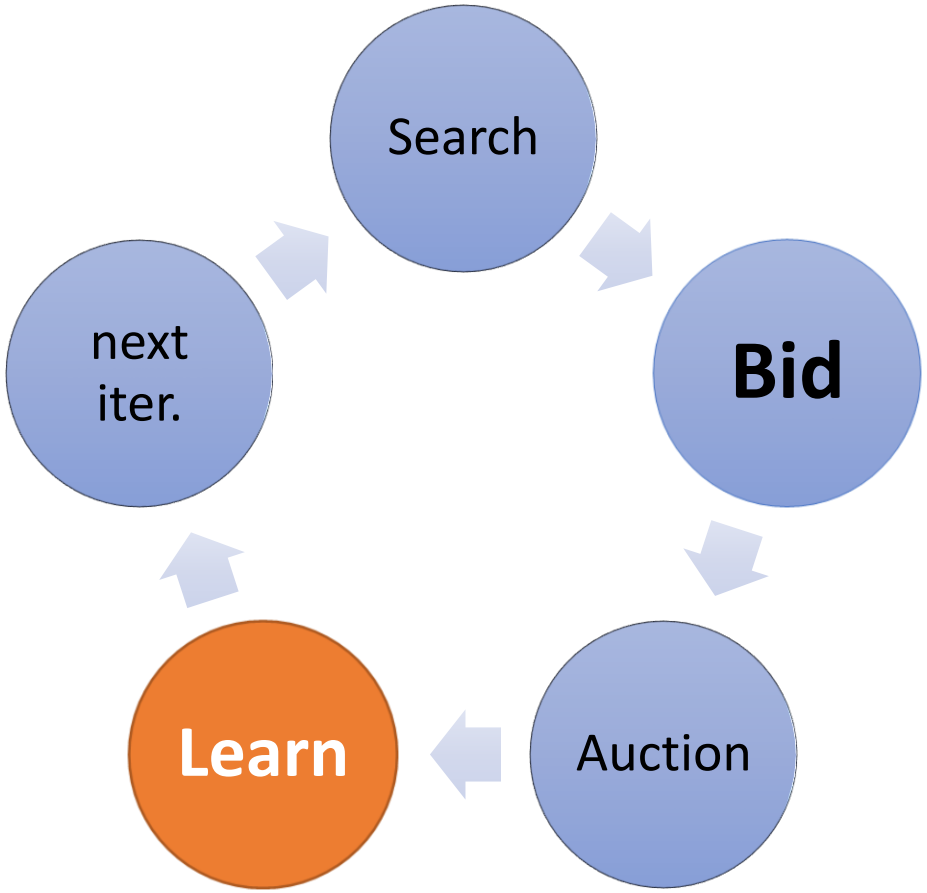
Policy.bid()

- For each iteration, each policy
  - Sends in bids
  - Gets the results & learn
- Function bid()



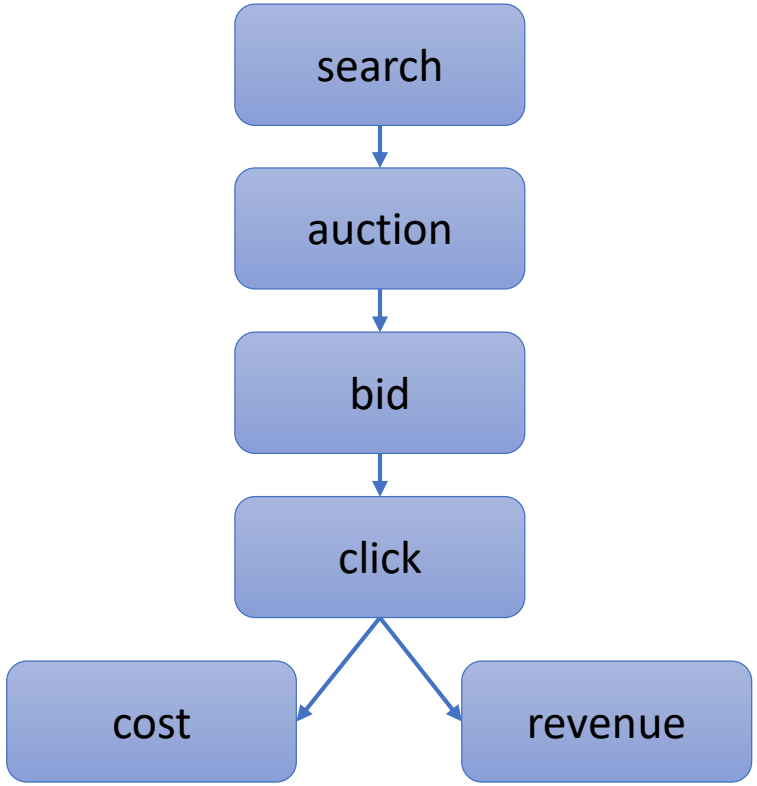
Policy.learn()

- For each iteration, each policy
  - Sends in bids
  - Gets the results & learn
- Function learn()



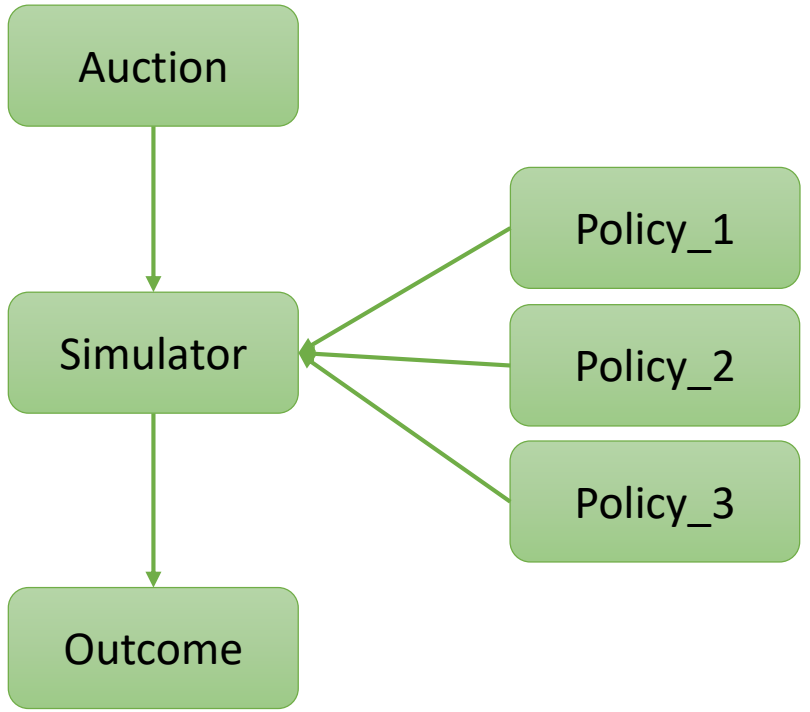
Simulator Operation, Revisited

- For each auction
  - Get bids from each policies
  - Determine winner
  - Randomly determine click counts
  - Compute cost per click
  - Randomly determine conversion & revenue
  - Report individualized results to each policies



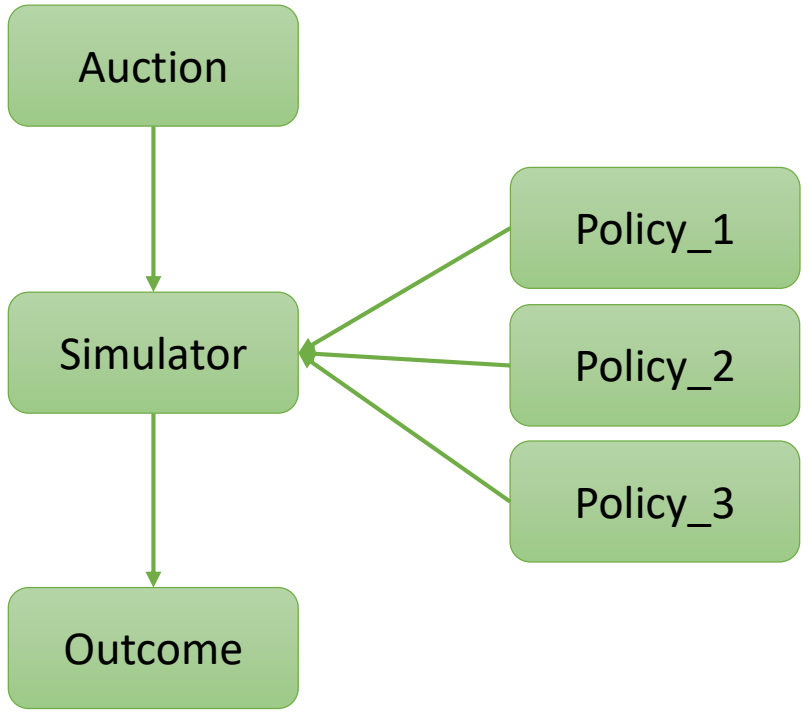
Simulator Code Structure

- Auctions are pregenerated
  - For all possible attributes



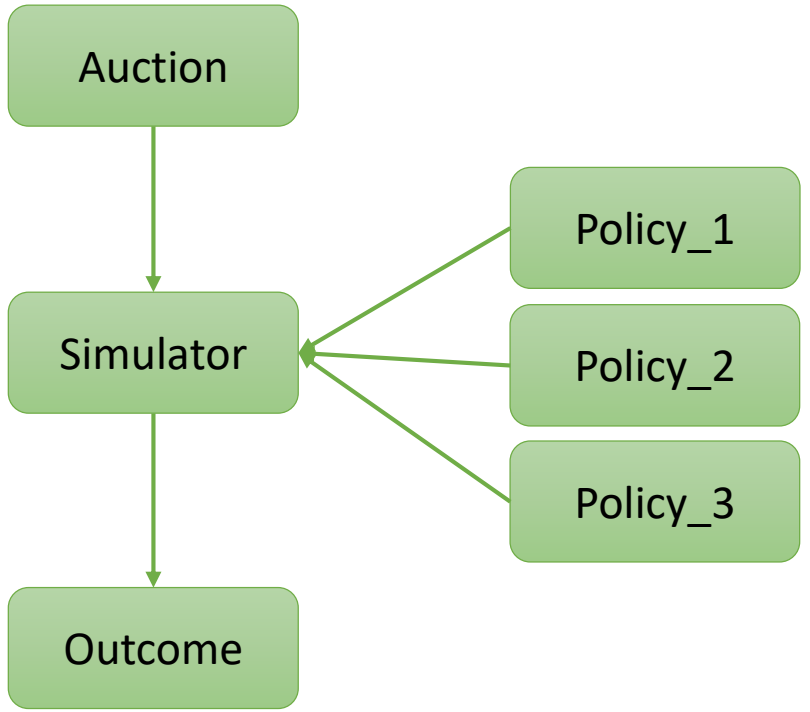
Simulator Code Structure

- Auctions are pregenerated
  - For all possible attributes
- For each iteration t
  - For each auction attribute a
    - Get bid from each policy



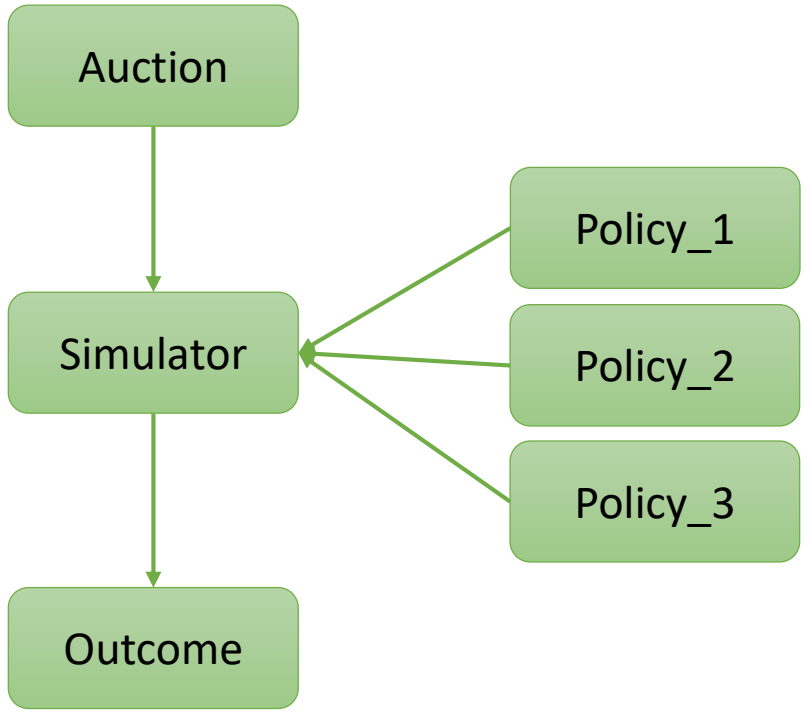
Simulator Code Structure

- Auctions are pregenerated
  - For all possible attributes
- For each iteration t
  - For each auction attribute a
    - Get bid from each policy
  - Each policy learns from result



Simulator Code Structure

- Auctions are pregenerated
  - For all possible attributes
- For each iteration t
  - For each auction attribute a
    - Get bid from each policy
  - Each policy learns from result
- Outcome is packed as xlsx



Dive to Code