Practical Budget Pacing Algorithms and Simulation Test Bed for eBay Marketplace Sponsored Search

Phuong Ha Nguyen, Djordje Gligorijevic, Arnab Borah,

Gajanan Adalinge and Abraham Bagherjeiran





AdKDD 2023

Goals

I Technical goal:

- ➔ **Simple but efficient simulator**
  - ◆ Detailed Implementation flow
  - ◆ New click value generator (for PPC sponsored search program)

II Research goal:

- ➔ **Evaluate throttling budget pacing algorithms in constrained environments**
- ➔ **Constraints:**
  - ◆ (1) many high performing campaigns with small budget,
  - ◆ (2) high external ads competition,
  - ◆ (3) second price auction policy in use,
  - ◆ (4) only know [remaining budget information] or [remaining budget and time information]

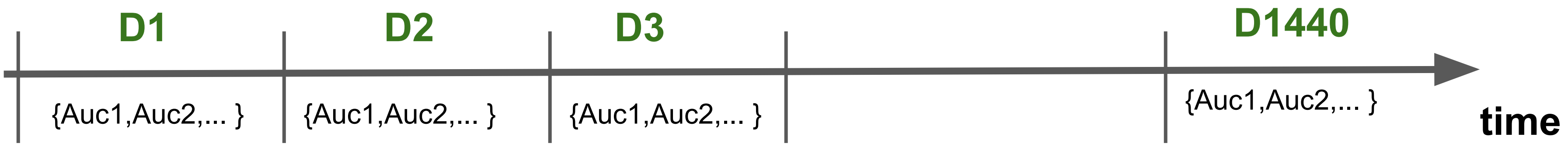
Conclusion: Greedy strategy based algorithms seem to be good candidates

Simulator

1. Implementation flow
2. Running mode
3. Click value generator
4. Evaluation

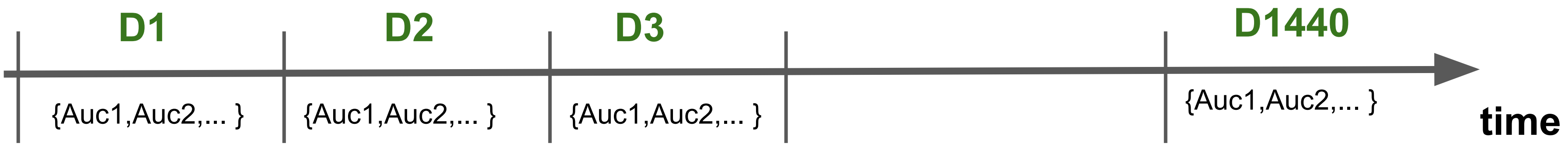
Simulator: Implementation Flow

Simulator: Implementation Flow



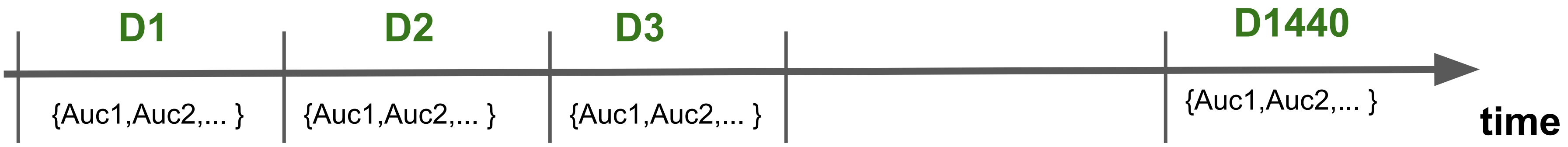
Define time buckets

Simulator: Implementation Flow



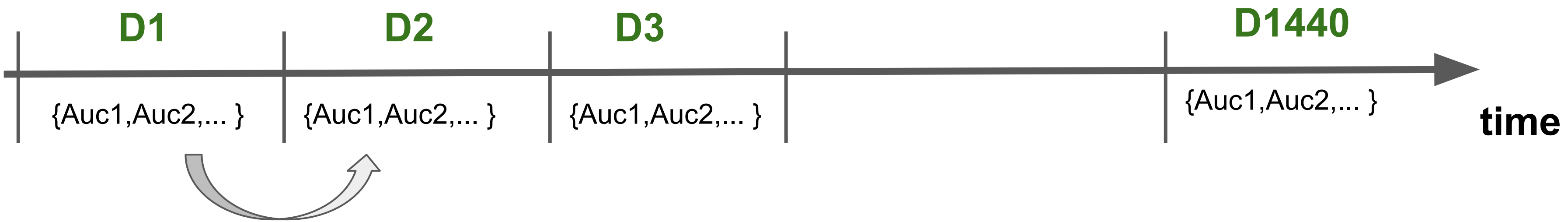
★ Auc(tion) = Query Q + Targeting campaigns A, B, C, … + historical data (shown items and their click values)

Simulator: Implementation Flow



- ★ Auc(tion) = Query Q + Retrieved* items from campaigns + historical data (shown items and their click values)
  - Simulated retrieval based on historical retrieval for simulation acceleration.
- ★ Campaigns’ items = predicted CTR (pCTR) and budget B

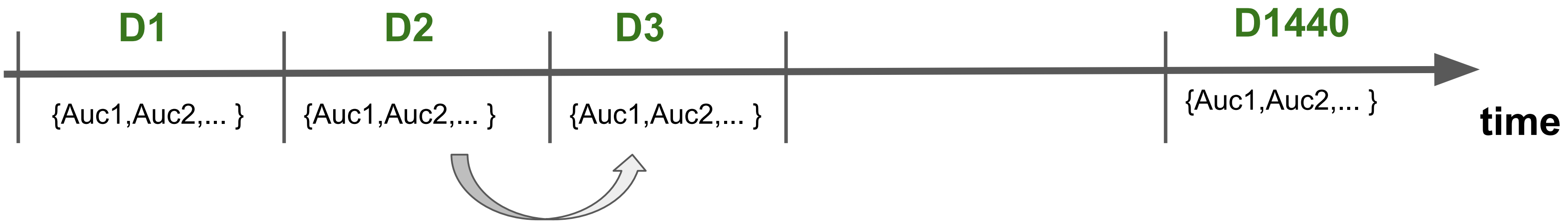
Simulator: running mode



- ➔ Run auctions independently in parallel
- ➔ Compute the spending of campaigns (need click value generator)
- ➔ Recompute the remaining budgets of campaigns

- ➔ Click value generator simulates the click behavior of buyers when seeing the ads

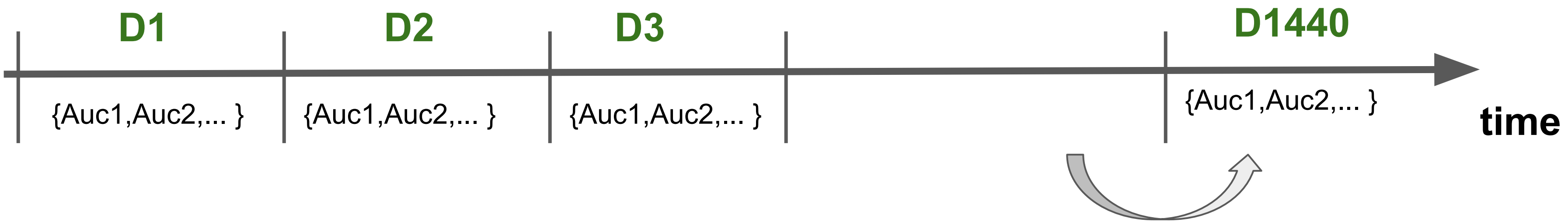
Simulator: running mode



- ➔ Run auctions independently in parallel
- ➔ Compute the spending of campaigns (need click value generator)
- ➔ Recompute the remaining budgets of campaigns

- ➔ Click value generator simulates the click behavior of buyers when seeing the ads

Simulator: running mode



- ➔ Run auctions independently in parallel
- ➔ Compute the spending of campaigns (generate counterfactual user clicks)
- ➔ Recompute the remaining budgets of campaigns

- ➔ Click value generator simulates the click behavior of buyers when seeing the ads

Simulator: click value generator

Q ⇐ $Item_1$, $pCTR_1$ and click value C1 [Historical log]

Simulator: click value generator

Q ⇐ $Item_1$, $pCTR_1$ and click value C1 **[Historical log]**

Q ⇐ $Item_2$, $pCTR_2$ and what is click value C2? **[Occurs when running simulator]**

Simulator: click value generator

Q ⇐ $Item_1$, $pCTR_1$ and click value C1 **[Historical log]**

Q ⇐ $Item_2$, $pCTR_2$ and what is click value C2? **[Simulator generated]**

Case 1. $pCTR_2 < pCTR_1$ and $C_1 = 1$.

- Generate a random number R in $[0, 1]$
- If $R < pCTR_2/pCTR_1$, then $C_2 = 1$ otherwise $C_2 = 0$.

Case 2. $pCTR_2 < pCTR_1$ and $C_1 = 0$. C_2 is assigned to 0.
Case 3. $pCTR_2 > pCTR_1$ and $C_1 = 1$. C_2 is assigned to 1.
Case 4. $pCTR_2 > pCTR_1$ and $C_1 = 0$.

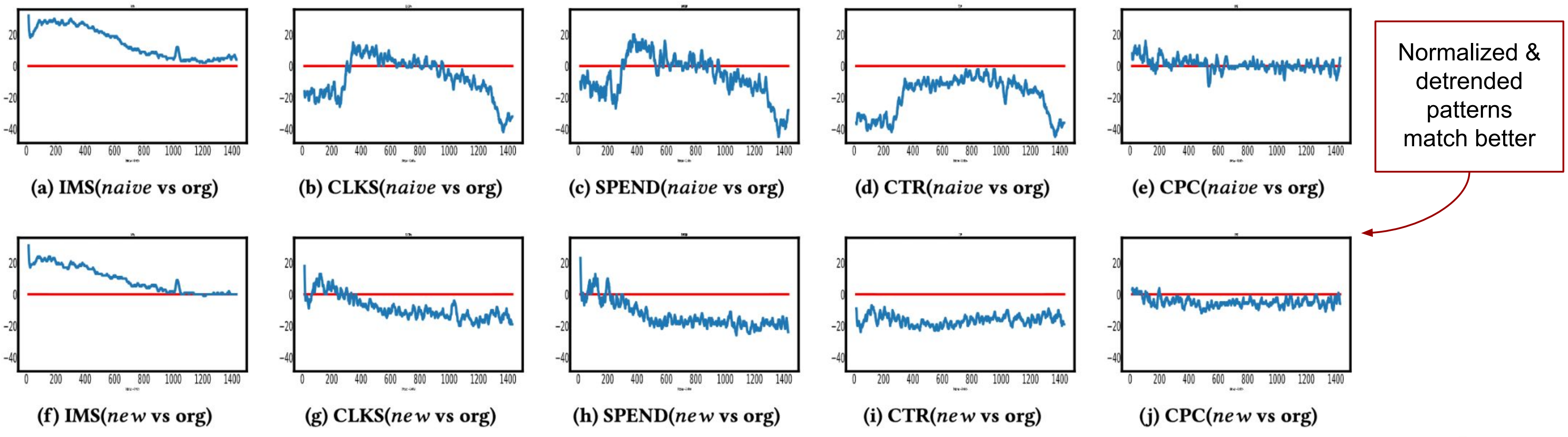
- Generate a random number R in $[0, 1]$
- If $R < (1 - pCTR_2)/(1 - pCTR_1)$, then $C_2 = 0$. Else, $C_2 = 1$.

Counterfactual click value generator algorithm

Simulator and Click Value Generator: Evaluation

	Total Impressions	Total Clicks	Total Spend	CTR	CPC
Naive	**8.5%**	-10.12%	-14.91%	-17.16%	-5.33%
New	13.5%	**-5.15%**	**-4.91%**	**-16.16%**	**+1.08%**

Table 1: Evaluation of the test bed according to changes in key metrics compared to the original data collected in logs when naive and the proposed click value generators are used.



(a) IMS(*naive* vs org) (b) CLKS(*naive* vs org) (c) SPEND(*naive* vs org) (d) CTR(*naive* vs org) (e) CPC(*naive* vs org)

(f) IMS(*new* vs org) (g) CLKS(*new* vs org) (h) SPEND(*new* vs org) (i) CTR(*new* vs org) (j) CPC(*new* vs org)

Figure 3: Comparison between 1) The simulator with naive click value generator vs Real system 2) The simulator with new click value generator vs Real system

Throttling budget pacing algorithms

1. **Pacing algorithms**
2. **Smoothness and Pacing Error Curve metrics**
3. **Results**

Throttling budget pacing algorithms

Throttling budget pacing algorithms

Campaign C → **Join ?** → **Auction A**

```
Generate a random number R in [0, 1]
if R ≤ θ then
    Do not throttle spend
else
    Throttle spend
end if
```

Campaign C → **Auction A**

Simple throttling budget pacing algorithms

if $methodName == RST$ **then**
    $\theta = 1$
end if

Spend control based on budget reset time

Simple throttling budget pacing algorithms

if $methodName == RST$ **then**
    $\theta = 1$
end if

Spend control based on budget reset time

if $methodName == Budget$ **then**
    Input: $budget_{rem}$ and $budget_{orig}$
    $x = budget_{rem}/budget_{orig}$
    $\theta = \Psi(x)/\Psi(1)$ where $\Psi(x) = 1 - e^{-x}$
end if

Spend control based on remaining budget

Simple throttling budget pacing algorithms

if $methodName == RST$ **then**
  $\theta = 1$
end if

Spend control based on budget reset time

if $methodName == ClkOp$ **then**
  Input: $budget_{rem}$, $maxBid$ and $ClickOpp_{rem}$
  $\theta = budget_{rem}/(maxBid \times ClickOpp_{rem})$
end if

Spend control based on remaining budget and remaining click opportunities

if $methodName == Budget$ **then**
  Input: $budget_{rem}$ and $budget_{orig}$
  $x = budget_{rem}/budget_{orig}$
  $\theta = \Psi(x)/\Psi(1)$ where $\Psi(x) = 1 - e^{-x}$
end if

Spend control based on remaining budget

Simple throttling budget pacing families of algorithms

if $methodName == RST$ **then**
    $\theta = 1$
end if

Spend control based on budget reset time

if $methodName == Budget$ **then**
    Input: $budget_{rem}$ and $budget_{orig}$
    $x = budget_{rem}/budget_{orig}$
    $\theta = \Psi(x)/\Psi(1)$ where $\Psi(x) = 1 - e^{-x}$
end if

Spend control based on remaining budget

if $methodName == ClkOp$ **then**
    Input: $budget_{rem}$, $maxBid$ and $ClickOpp_{rem}$
    $\theta = budget_{rem}/(maxBid \times ClickOpp_{rem})$
end if

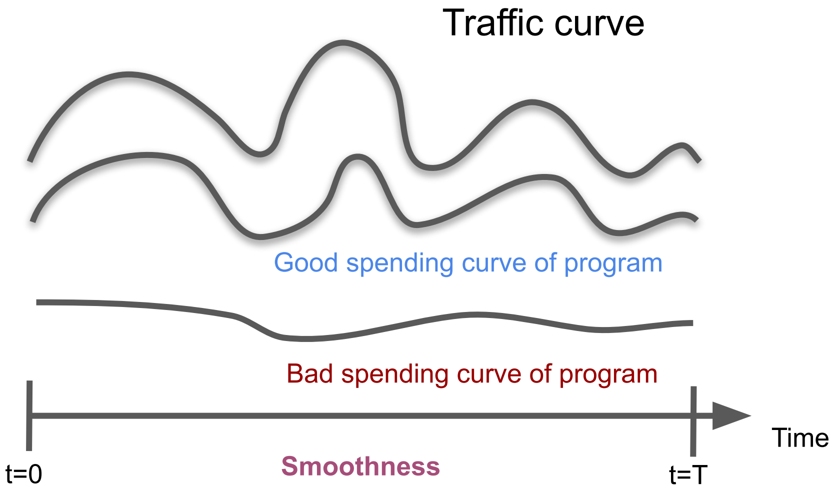
Spend control based on remaining budget and remaining click opportunities

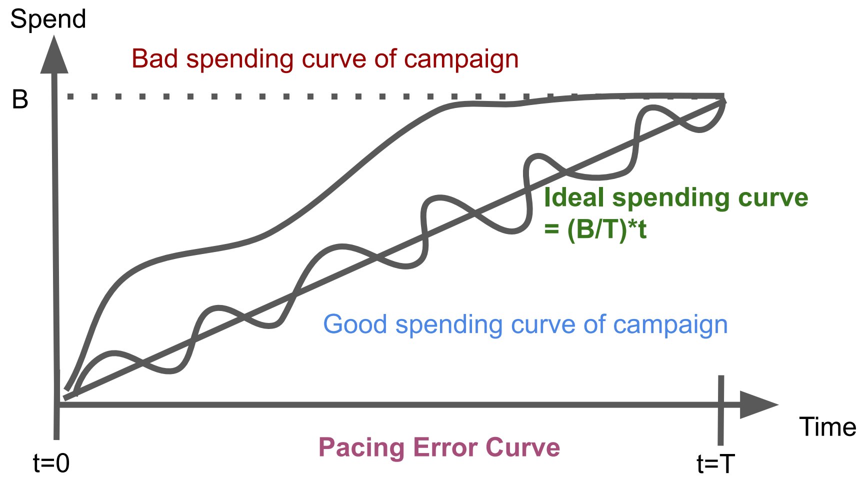
if $methodName == BudgetTime$ **then**
    Input: $budget_{rem}$,$budget_{orig}$,$time_{rem}$, $time_{orig}$
    Algo Parameters: f_{over}, f_{bound} and $bound$
    $rB = budget_{rem}/budget_{orig}$
    $rT = time_{rem}/time_{orig}$
    $ratio = rB/rT$
    if $ratio \geq 1$ **then**
        $\theta = \theta \times f_{over}$
    else
        $\theta = 1.0$
    end if
    if $\theta \leq f_{bound}$ **then**
        $\theta = 0.001$
    end if
end if

Spend control based on remaining budget and time information

BudgetTime variant with soft stop and hard start chosen as the best solution

Smoothness and pacing-error-curve metrics





$$\mathrm{PE} = (1/1440) \times \sum_{j=1}^{1440} \frac{|pS_j - pT_j|}{pT_j},$$

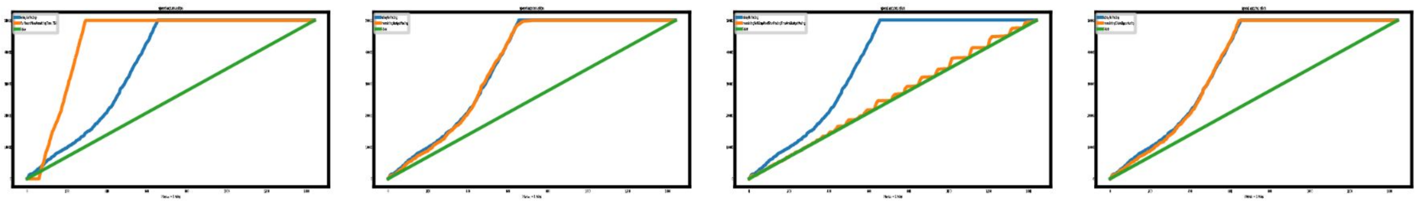
$$WPE = \sum_{i=1}^{n} \left(\sum_{t=1}^{1440} \left(\left| CS_{i,t} - t \times \frac{S_i}{1440} \right| \right) \times \frac{S_i}{S} \right)$$

Results

	RST750	Budget	BudgetTime	ClkOp
PE	-35.40%	+3.10%	**-52.20%**	+1.70%
WPE	+32.30%	-10.40%	**-39.50%**	+2.20%
totalImps	-8.50%	-3.20%	-3.90%	**-2.30%**
totalClks	**+8.10%**	+0.70%	+5.60%	+3.90%
totalSpends	**+7.60%**	+0.60%	+4.10%	+3.80%
CTR	**+18.30%**	+4.00%	+9.90%	+6.40%
CPC	-0.50%	0.00%	-1.40%	0.00%

* Budget reset at Midnight - RST0 is used as a baseline

Results



(a) $rank_1(RST_{750}$ vs $RST_0)$ (b) $rank_1(Budget$ vs $RST_0)$ (c) $rank_1(BudTime$ vs $RST_0)$ (d) $rank_1(ClkOp$ vs $RST_0)$

Figure 2: Comparison of Pacing Error Curves between 1) RST_{750} **(orange) vs** RST_0 **(blue) vs Ideal (green) 2)** $Budget$ **(orange) vs** RST_0 **(blue) vs Ideal (green) 3)** $ClkOp$ **(orange) vs** RST_0 **(blue) vs Ideal (green) 4)** $Bud(get)Time$ **(orange) vs** RST_0 **(blue) vs Ideal (green)**

$$\text{PE} = (1/1440) \times \sum_{j=1}^{1440} \frac{|pS_j - pT_j|}{pT_j},$$

$$WPE = \sum_{i=1}^{n} \left(\sum_{t=1}^{1440} \left(\left| CS_{i,t} - t \times \frac{S_i}{1440} \right| \right) \times \frac{S_i}{S} \right)$$

	RST750	Budget	BudgetTime	ClkOp
PE	-35.40%	+3.10%	**-52.20%**	+1.70%
WPE	+32.30%	-10.40%	**-39.50%**	+2.20%
totalImps	-8.50%	-3.20%	-3.90%	**-2.30%**
totalClks	**+8.10%**	+0.70%	+5.60%	+3.90%
totalSpends	**+7.60%**	+0.60%	+4.10%	+3.80%
CTR	**+18.30%**	+4.00%	+9.90%	+6.40%
CPC	-0.50%	0.00%	-1.40%	0.00%

Conclusion

1. Proposed a simple user response generator simulator
2. Evaluated four groups of throttling budget pacing algorithms
3. Greed strategy based pacing algorithms perform well in environments with high competition and with performing campaigns with small budgets