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

1-day historical log of sponsored program

\{Auc1.1,Auc2.1,\ldots\} \quad \{\ldots\} \quad \{\ldots\} \quad \{AucN.1,AucN.2,\ldots\}
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

\[ Q \leftarrow \text{Item}_1, \ p\text{CTR}_1 \text{ and click value } C1 \] [Historical log]
Simulator: click value generator

\[ Q \leftarrow \text{Item}_1, \ pCTR_1 \ \text{and click value } C_1 \ \text{[Historical log]} \]

\[ Q \leftarrow \text{Item}_2, \ pCTR_2 \ \text{and what is click value } C_2? \ \text{[Occurs when running simulator]} \]
Simulator: click value generator

Q $\leftarrow$ Item$_1$, pCTR$_1$ and click value C1  [Historical log]

Q $\leftarrow$ 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

<table>
<thead>
<tr>
<th></th>
<th>Total Impressions</th>
<th>Total Clicks</th>
<th>Total Spend</th>
<th>CTR</th>
<th>CPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive</td>
<td>8.5%</td>
<td>-10.12%</td>
<td>-14.91%</td>
<td>-17.16%</td>
<td>-5.33%</td>
</tr>
<tr>
<td>New</td>
<td>13.5%</td>
<td>-5.15%</td>
<td>-4.91%</td>
<td>-16.16%</td>
<td>+1.08%</td>
</tr>
</tbody>
</table>

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.

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

Campaign C → Join? → Auction A
Throttling budget pacing algorithms

Generate a random number $R$ in $[0, 1]$

- if $R \leq \theta$ then
  - Do not throttle spend
- else
  - Throttle spend

Campaign C → Join? → 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\textsubscript{rem} and budget\textsubscript{orig}
  \( x = \frac{\text{budget\textsubscript{rem}}}{\text{budget\textsubscript{orig}}} \)
  \( \theta = \frac{\Psi(x)}{\Psi(1)} \) where \( \Psi(x) = 1 - e^{-x} \)
end if

Spend control based on remaining budget
Simple throttling budget pacing algorithms

if methodName == RST then
    θ = 1
end if

Spend control based on budget reset time

if methodName == ClkOp then
    Input: budget_{rem}, maxBid and ClickOpp_{rem}
    θ = budget_{rem}/(maxBid × 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}
    θ = Ψ(x)/Ψ(1) where Ψ(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 = \frac{budget\(_{rem}\)}{budget\(_{orig}\)} \)
    \( \theta = \frac{\Psi(x)}{\Psi(1)} \) 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 = \frac{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 = \frac{budget\(_{rem}\)}{budget\(_{orig}\)} \)
    \( rT = \frac{time\(_{rem}\)}{time\(_{orig}\)} \)
    \( ratio = \frac{rB}{rT} \)
    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

\[ PE = \left( \frac{1}{1440} \right) \sum_{j=1}^{1440} \frac{|pS_j - pT_j|}{pT_j} \]

\[ WPE = \sum_{i=1}^{n} \left( \sum_{i=1}^{1440} \left( CS_{i,t} - t \times \frac{S_i}{1440} \right) \times \frac{S_i}{S} \right) \]
# Results

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

<table>
<thead>
<tr>
<th></th>
<th>RST750</th>
<th>Budget</th>
<th>BudgetTime</th>
<th>ClkOp</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PE</strong></td>
<td>-35.40%</td>
<td>+3.10%</td>
<td>-52.20%</td>
<td>+1.70%</td>
</tr>
<tr>
<td><strong>WPE</strong></td>
<td>+32.30%</td>
<td>-10.40%</td>
<td>-39.50%</td>
<td>+2.20%</td>
</tr>
<tr>
<td><strong>totalImps</strong></td>
<td>-8.50%</td>
<td>-3.20%</td>
<td>-3.90%</td>
<td>-2.30%</td>
</tr>
<tr>
<td><strong>totalClks</strong></td>
<td>+8.10%</td>
<td>+0.70%</td>
<td>+5.60%</td>
<td>+3.90%</td>
</tr>
<tr>
<td><strong>totalSpends</strong></td>
<td>+7.60%</td>
<td>+0.60%</td>
<td>+4.10%</td>
<td>+3.80%</td>
</tr>
<tr>
<td><strong>CTR</strong></td>
<td>+18.30%</td>
<td>+4.00%</td>
<td>+9.90%</td>
<td>+6.40%</td>
</tr>
<tr>
<td><strong>CPC</strong></td>
<td>-0.50%</td>
<td>0.00%</td>
<td>-1.40%</td>
<td>0.00%</td>
</tr>
</tbody>
</table>
Results

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)

\[
PE = \frac{1}{1440} \sum_{j=1}^{1440} \left| \frac{pS_j - pT_j}{pT_j} \right|
\]

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

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