The Balance Algorithm in Ad Placement

Justin Yu - 2025
COMP 4900G

Motivation

- Everytime you go online, your activity affects the ads you see
- Understanding how these ads are assigned is essential to optimizing advertising revenue
- This problem is formally known as the AdWords Problem
- The Balance Algorithm is a strong method used to solve the AdWords Problem

AdWords Problem

- Variant of the Bipartite Matching Problem
- Input: Bipartite Graph $G = (V = L \cup R, E)$
- Vertices in R come in an Online Manner along with the edges incident to them
- Match vertices in R to vertices in L
- Can match at most *b* vertices to any vertex in L

Adwords Problem - Advertising Lens

- L -> Advertisers
- b = $1
- R -> Online Queries
- ε -> Bid Price for a Query
- Goal: Maximize the budget spent by the advertisers, the revenue

Balance Algorithm

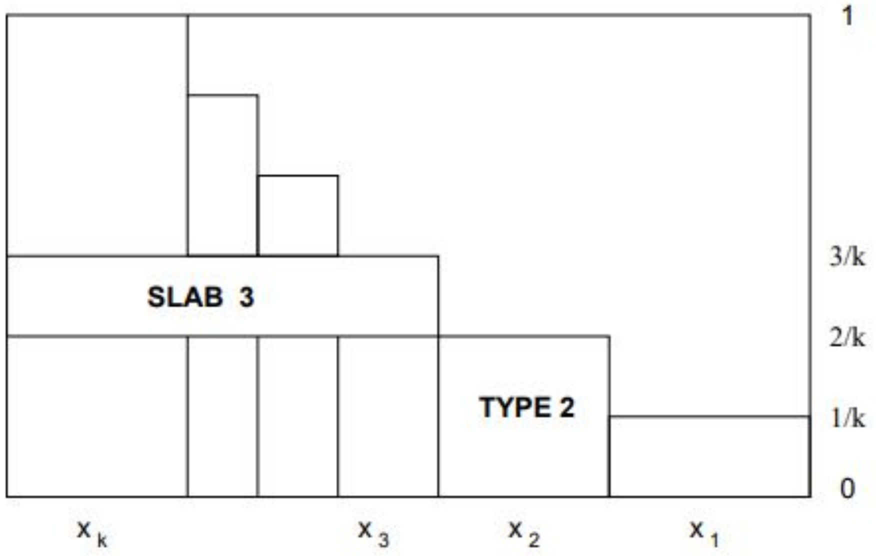
- For each vertex $w \in R$ revealed:
  - If there exists any neighbours of w in L that have been matched less than b times, match w to the neighbour that is matched to the fewest.

- Assigns query to the advertiser who has bid for it, has the budget to bid for it, and has the largest remaining budget

Analysis Preliminaries

- Each advertiser's budget is split into $k > 0$ slabs
- Optimal solution has every advertiser spending all their budget, resulting in a revenue of $1 * |L| = N$
- Each query can be paid for by a single slab or less
- Type i advertisers spend $\frac{i}{k}$ of their budget
- For $i = 1, \ldots, k$, x_i represents the number of advertisers of Type i
- β_j represents the amount spent from slab j for all advertisers, with $j = 1, \ldots, k$

Analysis Preliminaries



Example Figure from A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. “AdWords and generalized on-line matching”

Observation 1

In the Balance Algorithm, all the queries that are assigned optimally to a Type i, $i < k$, advertiser are paid by slabs ≤ i

- If a query q is assigned to an advertiser of Type i, $i < k$, which slab will be used to pay for it?
- Cannot be the last slab k
- Cannot be paid for by any slab $> i$

Observation 2

$$|\beta_1| = \frac{|L|}{k} = \frac{N}{k} \text{ and } |\beta_j| = \frac{N}{k} - \sum_{i=1}^{j-1} \frac{x_i}{k}$$

- Consider the definitions of x_i and β_j
- As *j* increases, the amount of advertisers that have Type $i \geq j$ decreases

Observation 3 / Lemma 1

$$\text{For } 1 \le i \le k-1, \sum_{j=1}^{i} x_j \le \sum_{j=1}^{i} \beta_j.$$

- Base Case: $i = 1$, $\beta_1 = \frac{N}{k}$
- Recursive Case: $2 \le i \le k - 1$, $x_1 + \ldots + x_i \le \beta_1 + \ldots + \beta_i$

Observation 4 / Lemma 2

$$\text{Revenue from the Balance Algorithm} \geq N - \sum_{i=1}^{k-1} x_i - \frac{N}{k} + \sum_{i=1}^{k-1} \frac{i}{k} x_i$$

- From the revenue of Type *k* advertisers - $N - \sum_{i=1}^{k-1} x_i$
- From the loss resulting from assumptions - $-\frac{N}{k}$
- From the Total Revenue of Type *i* advertisers - $+\sum_{i=1}^{k-1} \frac{i}{k} x_i$ where *i* < *k*
- Simplifies to $= N(1 - \frac{1}{k}) - \sum_{i=1}^{k-1} \frac{k-i}{k} x_i$

Bounding the Revenue - Primal LP

Primal LP

Maximize $\sum_{i=1}^{k-1} \frac{k-i}{k} x_i$

Subject to:

For all $i \in 1, ..., k-1 : \sum_{j=1}^{i} x_j \leq \sum_{j=1}^{i} \beta_j$ According to Observation 4 / Lemma 2

For all $i \in 1, ..., k : x_i \geq 0$

- To prove the competitive ratio, we must place a lower bound on the revenue of the Balance Algorithm
- Since $N(1-\frac{1}{k})$ is fixed, this is equivalent to placing upper bound on $\sum_{i=1}^{k-1} \frac{k-i}{k} x_i$

Complementary Slackness

The Complementary Slackness Theorem states that if we have feasible solutions x and y to Primal and Dual LP's respectively and they satisfy $\forall i : (b_i - \sum_j a_{ij}x_j)y_i = 0$ and $\forall j : (\sum_j a_{ij}y_j - c_j)x_j = 0$ then they are also optimal.

- In general, if we can show that the inequalities in our Primal and Dual LPs can be shown as equalities, they pass the complementary slackness test and are both feasible and optimal

Bounding the Revenue - Dual LP

Dual LP
Minimize $\sum_{i=1}^{k-1}(\frac{i}{k}N)y_i$

Subject to:
For all $i \in 1,...,k-1 : \sum_{j=i}^{k-1}(1+\frac{j-i}{k})y_j \geq \frac{k-i}{k}$
For all $i \in 1,...,k : y_i \geq 0$

- Primal LPs are of the form *max c · x*, where *Ax ≤ b* and *x ≥ 0*
- Dual LPs are of the form *min b · y*, where $A^Ty \geq c$ and *y ≥ 0*
- Obtain Dual LP through conversion of $\sum_{j=1}^{i} x_j \leq \sum_{j=1}^{i} \beta_j$ to $\sum_{j=1}^{i}(1+\frac{i-j}{k})x_j \leq \frac{i}{k}N$ through rearrangement

Feasible Solution - Primal LP

Primal LP

Maximize $\sum_{i=1}^{k-1} \frac{k-i}{k} x_i$

Subject to:

For all $i \in 1, ..., k-1$: $\sum_{j=1}^{i} (1 + \frac{i-j}{k}) x_j \leq \frac{i}{k} N$

For all $i \in 1, ..., k$: $x_i \geq 0$

- Set $\sum_{j=1}^{i} (1 + \frac{i-j}{k}) x_j = \frac{i}{k} N$ and solve for x_i for all *i = 1, ..., k-1*
- Obtain: $x_1 = \frac{N}{k}, x_2 = \frac{N}{k}(1 - \frac{1}{k}), ..., x_i = \frac{N}{k}(1 - \frac{1}{k})^{i-1}, ..., x_k = \frac{N}{k}(1 - \frac{1}{k})^{k-1}$

Feasible Solution - Dual LP

Dual LP
Minimize $\sum_{i=1}^{k-1}(\frac{i}{k}N)y_i$

Subject to:
For all $i \in 1, ..., k-1 : \sum_{j=i}^{k-1}(1+\frac{j-i}{k})y_j \geq \frac{k-i}{k}$
For all $i \in 1, ..., k : y_i \geq 0$

- Set $\sum_{j=1}^{k-1}(1+\frac{j-i}{k}N)y_j = \frac{k-i}{k}$ and solve for all *i = 1, ..., k-1*
- Obtain: $y_{k-1} = \frac{1}{k}, y_{k-2} = \frac{1}{k}(1-\frac{1}{k}), y_{k-3} = \frac{1}{k}(1-\frac{1}{k})^2, ..., y_{k-i} = \frac{1}{k}(1-\frac{1}{k})^{i-1}, ..., y_1 = \frac{1}{k}(1-\frac{1}{k})^{k-2}$

Optimal Objective Function Value

- Complementary Slackness conditions hold
- $x_i = \frac{N}{k}(1 - \frac{1}{k})^{i-1}$ is a feasible and optimal solution for the Primal LP
- All y_i are feasible and optimal for the Dual LP
- Can now substitute the optimal x in the Primal LP to find the competitive ratio

Optimal Objective Function Value

Optimal Objective Function Value $\phi = c \cdot x^*$
$\sum_{i=1}^{k-1}(\frac{k-i}{k})x_i = \sum_{i=1}^{k-1}(\frac{k-i}{k})(\frac{N}{k})(1-\frac{1}{k})^{i-1}$

Multiplying the first two terms of the summation together, factoring out the $\frac{N}{k^2}$ and applying linearity of summation
$= \frac{N}{k^2}[\sum_{i=1}^{k-1} k(1-\frac{1}{k})^{i-1} - \sum_{i=1}^{k-1} i(1-\frac{1}{k})^{i-1}]$

Applying the geometric series formula $\sum_{i=1}^{n} a_i r^{i-1} = a_1(\frac{1-r^n}{1-r})$
and the arithmetic geometric series formula $\sum_{i=1}^{n} i r^{i-1} = \frac{1-r^n}{(-r)^2} - \frac{nr^n}{1-r}$
$= \frac{N}{k^2}[k\frac{1-(1-\frac{1}{k})^{k-1}}{1-(1-\frac{1}{k})} - \frac{k^2}{k-1}((1-\frac{1}{k})^k - k(2(1-\frac{1}{k})^k - 1]$

Distribute the k-value in the $\frac{k^2}{k-1}$ term and simplify
$= \frac{N}{k^2}[k^2(1-(1-\frac{1}{k})^{k-1} - \frac{k^2}{k-1}((1-2k)(1-\frac{1}{k})^k - k + 1]$

Optimal Objective Function Value

Factor k^2 out and cancel it out with the denominator, then combine the remaining terms

$$= N[\frac{(k-1)(1-(1-\frac{1}{k})^{k-1}-(1-2k)(1-\frac{1}{k})^{k}-k+1}{k-1}$$

Distribute $(k-1)$ to $(1-(1-\frac{1}{k})^{k-1}$ and cancel out the terms

$$= N[\frac{-(k-1)(1-\frac{1}{k})^{k-1}-(1-2k)(1-\frac{1}{k})^{k}}{k-1}]$$

Distribute $-(k-1)$ to $(1-\frac{1}{k})^{k-1}$

$$= N[\frac{-k(1-\frac{1}{k})^{k}-(1-2k)(1-\frac{1}{k})^{k}}{k-1}]$$

Factor out $(1-\frac{1}{k})^{k}$ and simplify

$$= N(1-\frac{1}{k})^{k}$$

Optimal Objective Function Value

- As *k* approaches infinity, the upper bound on $\sum_{i=1}^{k-1}(\frac{k-i}{k})x_i$, $N(1-\frac{1}{k})^k$ becomes $\frac{N}{e}$
- By applying this upper bound to the revenue of the Balance Algorithm Equation:

$$N(1-\frac{1}{k}) - \sum_{i=1}^{k-1}\frac{k-i}{k}x_i \geq N(1-\frac{1}{k}) - \frac{N}{e} \approx N(1-\frac{1}{e})$$

- Therefore the Competitive Ratio of the Balance Algorithm is $1-\frac{1}{e}$

References

- A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. “AdWords and generalized on-line matching”
- R. M. Karp, U. Vazirani, and V. V. Vazirani, “An optimal algorithm for on-line bipartitematching”
- A. Mehta, “Online matching and ad allocation”