Online Algorithms

Anil Maheshwari

anil@scs.carleton.ca
School of Computer Science
Carleton University
Canada

Outline

Problem Definition

Competitive Ratio

Bipartite Matching

Primal-Dual LP

Fractional Matching

Randomized Matching

BALANCE Algorithm

Complementary Slackness

Problem Definition

Basic Problem

A problem motivated from which advertisement to display on the web

Manufacturer	Products	Ad Amount	Total Budget
I	A	\$1	100
II	A,B	\$ 2	100

Online Queries for Products A and B.

Question: Which Manufacturer's Ad should be shown given that we can display exactly one advertisement at a time?

Complication: We don't know how many queries, and with what distribution, for each product we will receive.

Example Query Sequences

Manufacturer	Products	Ad Amount	Total Budget
I	A	\$1	100
II	A,B	\$ 2	100

Sample Query Sequences:

1. 50 for B followed by 100 for A
2. 100 for A followed by 50 for B
3. intermix of 100 for A and 50 for B
4. intermix of ? for A and ? for B

In Cases 1-3, it is best to assign all B's to Manufacturer II and all A's to Manufacturer I, with a total revenue of \$200.

What to do in Case 4?

Competitive Ratio

Competitive Ratio

Ratio of the value returned by an Online Algorithm in comparison to the (best) Offline algorithm.

What is the largest value of $c \leq 1$, such that

$$\frac{\textbf{Value (online algorithm)}}{\textbf{Value (off-line algorithm)}} \geq c$$

Online Algorithmic Problems

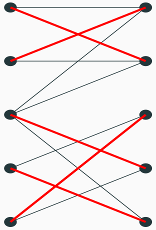
1. Bipartite Matching
2. Fractional Bipartite Matching
3. Randomized Bipartite Matching
4. b-Matching - Adwords

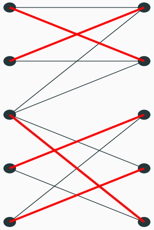
Bipartite Matching

Bipartite Matching

Let $G = (V = L \cup R, E)$ be a bipartite graph where the vertex set V consists of the sets L and R (referred to as 'left' and 'right' sets) and a set E of edges (v, w) where $v \in L$ and $w \in R$.

The set $M \subseteq E$ is a matching in G if no two edges in M share a vertex.

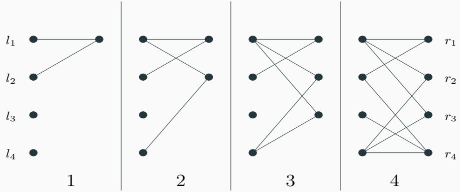




Online Matching Problem

Input: All the vertices in the set L are known in advance, but the vertices in R and the edges are presented over time. At each time instant $t \in \{1, 2, 3, \dots\}$, a new vertex $r_t \in R$ and all its incident edges arrive.

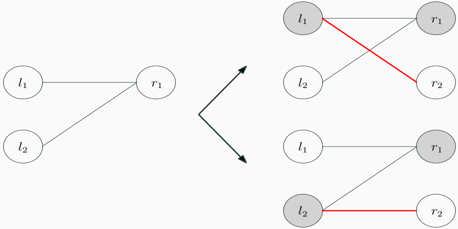
Task: The online matching algorithm needs to decide among all the currently unmatched neighbors of r_t in the set L to which vertex (if any) r_t should be matched. The vertex r_t remains matched to that vertex for the rest of the algorithm.



Output: Find a matching M of the largest possible size. Find M such that the ratio $\frac{|M|}{|M^*|}$ is as large as possible, where M^* is (offline) maximum matching in G.

Lower Bound on Deterministic Algorithms

Consider a bipartite graph on 4 vertices, where $L = \{l_1, l_2\}$ and $R = \{r_1, r_2\}$. At the first time step the algorithm is presented with the vertex r_1 and the two incident edges (r_1, l_1) and (r_1, l_2).



Adversary chooses what to do in the next time stamp and therefore the Competitive Ratio $= \frac{1}{2}$

Greedy Bipartite Matching Algorihm

Greedy Online Matching Algorithm:

At time step t:

Match r_t to any of the unmatched neighbors in the set L

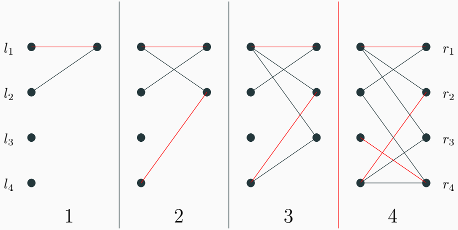


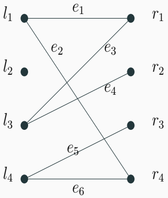
Figure 1: Competitive Ratio: $\frac{|M|}{|M^*|} = \frac{3}{4}$

We will show that the Greedy Online Matching Algorithm has a competitive ratio $\geq \frac{1}{2}$

Linear Programs and Dual Linear Programs

Definitions

1. For each edge $e \in E$, let $x_e \geq 0$. Let $x = \left[x_1, \ldots, x_{|E|}\right]$ be the vector of variables corresponding to the edges.
2. $\mathcal{N}(v)$, the neighborhood set of v, is the set of edges incident to the vertex v.
3. $c = \left[1, \ldots, 1\right]$ is a vector of all 1s of length $|E|$
4. $b = \left[1, \ldots, 1\right]$ is a vector of all 1s of length $|V|$
5. A is a $|V| \times |E|$ matrix and its ij-th entry is 1 if the edge corresponding to the column j is incident on the vertex corresponding to the row i, otherwise 0.



$$A = \begin{array}{c} \\ l_1 \\ l_2 \\ l_3 \\ l_4 \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Primal LP

The Primal Linear Program can be stated as follows:

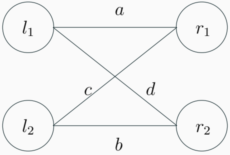
Primal LP:

Objective function:	
$\max \sum_{e \in E} x_e$	$\max c^T x$
Subject to:	
For all $v \in L \cup R : \sum_{e \in \mathcal{N}(v)} x_e \leq 1$	$Ax \leq b$
For all $e \in E : x_e \geq 0$	$x \geq 0$.

Objective function

The size of the maximum matching is the optimal value of the objective function of the LP

Example: Primal LP



Bipartite Matching LP:

$\max a + b + c + d$,

Subject to:

$l_1 : a + d \leq 1$

$l_2 : b + c \leq 1$

$r_1 : a + c \leq 1$

$r_2 : b + d \leq 1$

$a, b, c, d \geq 0$

Maximum value of the objective function of LP is 2.

For example, set $a = b = c = d = \frac{1}{2}$

Alternatively, set $a = b = 1$ and $c = d = 0$

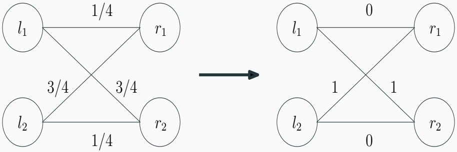
Matching LP

Integral Solution

For the bipartite matching LP

$$\boxed{\begin{array}{l} \max \sum_{e \in E} x_e, \\ \forall v \in L \cup R : \sum_{e \in \mathcal{N}(v)} x_e \leq 1, \\ \forall e \in E : x_e \geq 0 \end{array}}$$

there is always an integral solution (i.e. each variable taking integral values) that achieves an optimal value.



Primal-Dual LP

Primal-Dual LP

Consider the following Linear Program:

$$\max x_1 + x_2$$
$$x_1 + 2x_2 \leq 4$$
$$2x_1 + x_2 \leq 6$$
$$x_1, x_2 \geq 0$$

Since $x_1 \geq 0$ and $x_2 \geq 0$: we have

Observation 1: Value of objective function $x_1 + x_2 \leq 2x_1 + x_2 \leq 6$

Observation 2: Value of objective function $x_1 + x_2 \leq x_1 + 2x_2 \leq 4$

Observation 3: Value of objective function
$x_1 + x_2 \leq \frac{1}{2}(x_1 + 2x_2) + \frac{1}{4}(2x_1 + x_2) = x_1 + \frac{5}{4}x_2 \leq \frac{1}{2}4 + \frac{1}{4}6 = \frac{7}{2}$

Primal-Dual LP

Observation 4: Value of objective function is upper bounded by
$x_1+x_2 \leq y_1(x_1+2x_2)+y_2(2x_1+x_2) = x_1(y_1+2y_2)+x_2(2y_1+y_2)) \leq 4y_1+6y_2$,
provided

1. $y_1, y_2 \geq 0$
2. $y_1 + 2y_2 \geq 1$ (corresponding to x_1)
3. $2y_1 + y_2 \geq 1$ (corresponding to x_2)

Finding the right upper bound can also be expressed as a **(dual)** linear program:

$$\begin{aligned} \min 4y_1 + 6y_2 \\ y_1 + 2y_2 \geq 1 \\ 2y_1 + y_2 \geq 1 \\ y_1, y_2 \geq 0 \end{aligned}$$

Example (contd.)

	Primal	Dual
Objective Value	$\max x_1 + x_2$	$\min 4y_1 + 6y_2$
Constraints	$x_1 + 2x_2 \leq 4$	$y_1 + 2y_2 \geq 1$
	$2x_1 + x_2 \leq 6$	$2y_1 + y_2 \geq 1$
	$x_1, x_2 \geq 0$	$y_1, y_2 \geq 0$

- Set $y_1 = y_2 = 1/3$. This choice satisfies Dual LP constraints and results in an upper bound of $\frac{10}{3} < \frac{7}{2}$
- Set $x_1 = \frac{8}{3}$ and $x_2 = \frac{2}{3}$ satisfies Primal LP constraints and results in the objective value of $\frac{10}{3}$

$\implies$ The upper bound using the linear combination that we obtained is the optimal value

Primal-Dual LP - Standard Form

Primal-Dual LP Pair

Primal LP: $\max c^T x$ subject to $Ax \leq b$, $x \geq 0$
Dual LP: $\min b^T y$, subject to $A^T y \geq c$, $y \geq 0$

Observations:

1. For each variable in the Primal we have a constraint in the Dual.
2. For each constraint in the Primal we have a variable in the Dual.
3. Maximization becomes a Minimization problem.
4. (Weak Duality) If x and y are feasible solutions to the Primal and Dual LPs: $c^T x \leq (A^T y)^T x = y^T (Ax) \leq y^T b = b^T y$

Strong Duality Theorem

If x and y are optimal values for the Primal and Dual LPs, then $c^T x = b^T y$

Primal-Dual LP for Matching

Primal LP:

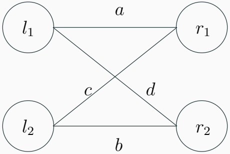
$\max \sum_{e \in E} x_e$	$\max c^T x$
Subject to:	
For all $v \in \{L \cup R\}$: $\sum_{e \in \mathcal{N}(v)} x_e \leq 1$	$Ax \leq b$
For all $e \in E : x_e \geq 0$	$x \geq 0$

For Dual LP: Introduce $|V|$ variables corresponding to each vertex constraint of the primal. We label them $p_1, \ldots, p_{|V|}$ and let $p = (p_1, \ldots, p_{|V|})^T$

Dual LP:

$\min \sum_{v \in V} p_v$	$\min b^T p$
Subject to:	
For all edges $e = (v, w) \in E : p_v + p_w \geq 1$	$A^T p \geq c$
$p_v \geq 0$, for all $v \in V$	$p \geq 0$

Example: Primal-Dual Bipartite Matching LP



	Primal LP	Dual LP
Objective Value	$\max a + b + c + d$	$\min l_1 + l_2 + r_1 + r_2$
Subject to:	$l_1 : a + d \leq 1$	$a : l_1 + r_1 \geq 1$
	$l_2 : b + c \leq 1$	$b : l_2 + r_2 \geq 1$
	$r_1 : a + c \leq 1$	$c : l_2 + r_1 \geq 1$
	$r_2 : b + d \leq 1$	$d : l_1 + r_2 \geq 1$
	$a, b, c, d \geq 0$	$l_1, l_2, r_1, r_2 \geq 0$

Greedy Bipartite Matching Algorihm

Greedy Online Matching Algorithm:
At time step t:
Match r_t to any of the unmatched neighbors in the set L

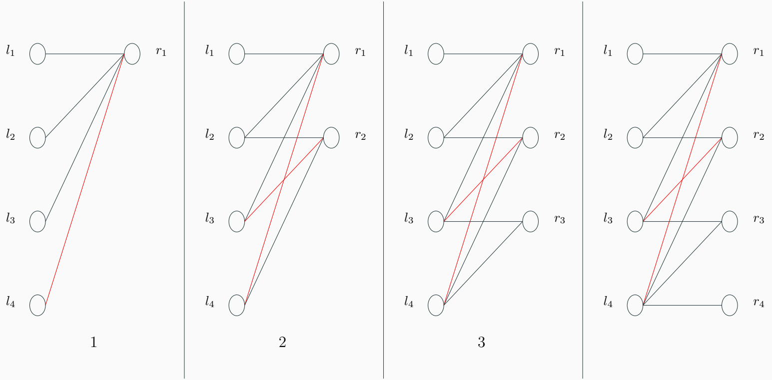


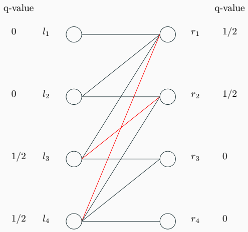
Figure 2: Red edges in greedy matching.

Analysis of Greedy Algorithm

For analysis, for each vertex $v \in V$ define the quantity $q_v \geq 0$ as follows.

Initialize: For all $v \in V$: $q_v = 0$

Update: If the greedy algorithm decides to add edge $e = (v, w)$ to the matching, set $q_v = \frac{1}{2}$ and $q_w = \frac{1}{2}$.



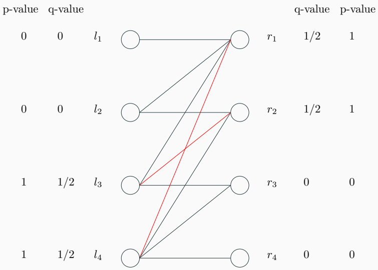
Observation: The size of the matching M reported by the greedy algorithm is given by $|M| = \sum_{v \in L \cup R} q_v$

Analysis of Greedy Algorithm (contd.)

$\min \sum_{v \in V} p_v$	$\min b^T p$
Subject to:	
For all edges $e = (v, w) \in E : p_v + p_w \geq 1$	$A^T p \geq c$
$p_v \geq 0$, for all $v \in L \cup R$	$p \geq 0.$

For all $v \in V$: set $p_v = 2q_v$

Observation: For each edge $e = (v, w) \in E$, $p_v + p_w = 2q_v + 2q_w \geq 1$ (as the greedy algorithm doesn't leave both v and w unmatched)



Competitiveness of Greedy

Greedy algorithm is $\frac{1}{2}$-competitive.

Proof:

1. The value of the objective function of the dual LP is given by $\sum_{v \in V} p_v = 2 \sum_{v \in V} q_v = 2|M|$.
2. From the weak/strong duality, the objective value of dual LP ($= 2|M|$) is an upper bound to the objective value of Primal LP ($= |M^*|$)
3. $2|M| \geq |M^*|$
4. Thus, Competitive Ratio $\frac{|M|}{|M^*|} \geq \frac{1}{2}$

$\square$

NOTE: Primal-Dual LP is used only for proving that the Greedy Online Bipartite Matching Algorithm is $\frac{1}{2}$-competitive.

In the algorithm, we **don't** solve an LP.

Fractional Matching

Fractional Bipartite Matching

Input: Bipartite graph $G = (V = L \cup R, E)$

- Vertices in L are known in advance and each has a unit capacity.

- Vertices in R come in an online fashion along with its incident edges.

- Each vertex in R has a unit amount of information to handout.

- At each time instant t, we transmit the information from the current vertex r_t to its neighboring vertices in the set L if:

1. Sum total of the information transmitted from r_t to its neighbors in L is at most 1.
2. One or more neighbors of r_t may receive the information provided they do not exceed their capacity of 1.
3. Once the information is transmitted it cannot be reversed.

Problem Formulation

An Observation

Let $x_e =$ Amount of information that travels on edge e.
For any vertex $v \in \{L \cup R\}$, Level$(v) = \sum_{w \in N(v)} x_{vw} \leq 1$

Fractional Matching Problem

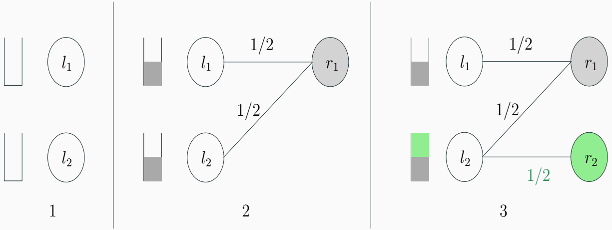
Maximize the total information received by the vertices in the set L, i.e. $\max \sum_{e \in E} x_e$

Note: For the (static) graph G the Bipartite Matching LP,
$\max \sum_{e \in E} x_e, \forall v \in L \cup R : \sum_{e \in \mathcal{N}(v)} x_e \leq 1, \forall e \in E : x_e \geq 0,$
applies to this problem formulation as x_e's can take fractional values. Value of the objective function is the size of the maximum fractional matching in G.

WATERLEVEL Algorithm

At any time step t:
Drain the water (information) from r_t to its neighbors where the preference is always given to the neighbor with the largest residual capacity remaining till
Case 1: All neighbors of r_t are saturated, or
Case 2: r_t transmits all its information

Example 1: $G = (V = L \cup R, E)$, where $L = \{l_1, l_2\}$, $R = \{r_1, r_2\}$ and $E = \{r_1 l_1, r_1 l_2, r_2 l_2\}$



Example 2

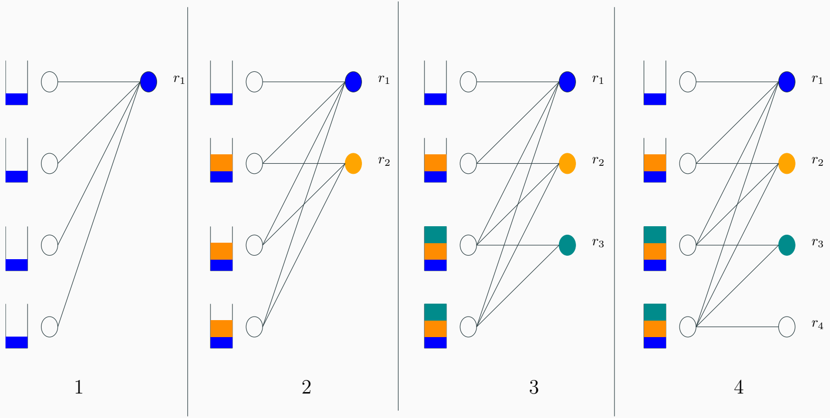
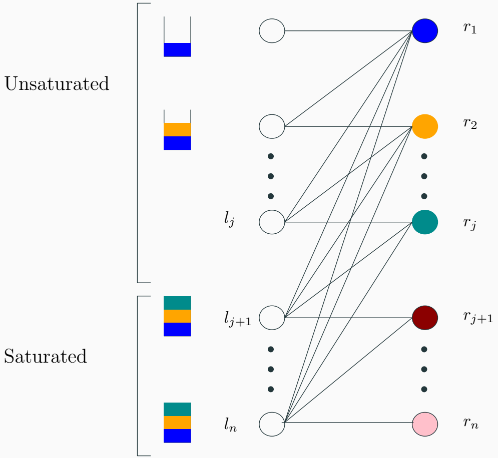


Figure 3: Fractional Flow Value $= 4 \times \frac{1}{4} + 3 \times \frac{1}{3} + 2 \times \frac{5}{12} = \frac{17}{6} > 2$. Total flow value received at vertices in L are $l_1 = \frac{1}{4}$; $l_2 = \frac{1}{4} + \frac{1}{3} = \frac{7}{12}$; and $l_3 = l_4 = \frac{1}{4} + \frac{1}{3} + \frac{5}{12} = 1$.

Example 3

Let $L = \{l_1, \ldots, l_n\}$ and $R = \{r_1, \ldots, r_n\}$
Let $E = \{(l_i, r_j) | i \geq j, \text{ for all } i, j \in \{1, \ldots, n\}\}$



Example 3 (contd.)

Let j be the first index at which there is no further flow of information from vertices in $r_t \in R$ for $t > j$.
$\implies$ All the vertices $l_{j+1}, \ldots, l_n$ are saturated

The index j must satisfy $\frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{n-j+1} \geq 1$

For what value of j, $\frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{n-j+1} \geq 1$?

Recall that $H_n = \sum_{i=1}^{n} \frac{1}{i} \approx \ln n$

Thus, $\frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{n-j+1} = H_n - H_{n-j} \approx \ln n - \ln(n-j) = \ln \frac{n}{n-j}$

If $j = n(1 - \frac{1}{e})$,

$\ln \frac{n}{n-j} = \ln \frac{n}{n-n(1-\frac{1}{e})} = \ln e = 1$

Competitive Ratio

The competitive ratio of the Waterlevel Algorithm on Example 3 is $\approx (1 - \frac{1}{e})$

Proof:

1. Vertices $r_1, \dots, r_{j-1}$ are able to send all of their information to vertices of L
2. Vertices $r_{j+1}, \dots, r_n$ aren't able to send any information
3. Total weight of the fractional matching computed by the WATERLEVEL algorithm is $\approx j \approx n(1 - \frac{1}{e})$
4. G has a perfect matching (match l_i to r_i) $\implies$ optimal offline matching has size n.
5. Competitive ratio $\approx \frac{n(1-\frac{1}{e})}{n} = (1 - \frac{1}{e}) \approx 0.63$

$\square$

Primal Dual Analysis of WATERLEVEL Algorithm

Dual LP:	$\min \sum_{v \in V} p_v$	$\min b^T p$
	Subject to:	
	For all edges $e = (v, w) \in E : p_v + p_w \geq 1$	$A^T p \geq c$
	For all $v \in V : p_v \geq 0$	$p \geq 0$

Recall the analysis of Greedy Algorithm:
By setting $p_v = 2q_v$, we had

1. For all vertices v, $p_v \geq 0$
2. For all edges $e = (v, w) \in E$, $p_v + p_w \geq 1$
3. $\sum_{v \in V} p_v = 2 \sum_{v \in V} q_v = 2|M|$
4. Upper Bound: $2|M| \geq |M^*| \implies \frac{|M|}{|M^*|} \geq \frac{1}{2}$

Question: Can we follow the same analysis?

1st Approach

For all $v \in V$, intialize $q_v = 0$

After the execution of the WATERLEVEL algorithm, for each edge $e = vw \in E$, set $q_v = q_v + \frac{1}{2}x_{vw}$ and $q_w = q_w + \frac{1}{2}x_{vw}$, where x_{vw} is the amount of flow on edge vw.

Size of the fractional matching: $|M| = \sum\limits_{v \in V} q_v$

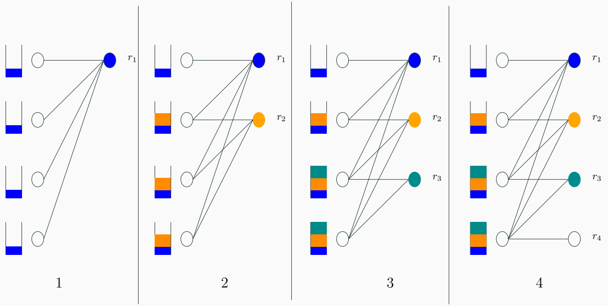
What about the constraint $p_v + p_w \geq 1$ for all edges $e = (v, w) \in E$ in the Dual LP?

Can we get away by setting $p_v = cq_v$, where $c < 2$?

If so, competitive ratio will be $\frac{1}{c}$

1st Approach (contd.)

Consider the edge l_4r_4 in Example 2. Edges incident to r_4 have $x_e = 0$



For any of those edges the sum total of the q values of their end points is at most $\frac{1}{2}$

$\implies$ to satisfy the Dual LP constraints we need to set $p_v = 2q_v$

2nd Approach

Idea: Uneven Split

Instead of splitting the value of the flow x_e on each edge $e = vw \in E$ between its endpoints evenly, split in such a way that $q_v + q_w \geq 1 - \frac{1}{e}$

$p_v = \frac{e}{e-1} q_v$

All constraints of the Dual LP are satisfied

The competitive ratio will be $\geq 1 - \frac{1}{e} \approx 0.63$

A Key Observation

Observation

Assume that after the algorithm has terminated, the vertex $v \in L$ isn't saturated. Let the information content that v has received during the entire execution of the algorithm equals Level$(v) < 1$. Moreover, assume that $vw \in E$. Now consider the step in the online algorithm when $w \in R$ was revealed. In that step w routed the information to its neighbors (including v) in L whose Level's were at most Level(v).

Proof: Follows from the water-filling analogy since v finished with Level(v) at the termination and w can only send information to its neighbors up to Level(v) upon its arrival.

Partition Function $f(x) = e^{x-1}$

Let $f(x) = e^{x-1}$ for $0 \leq x \leq 1$

Let $w \in R$ and let $vw \in E$.
Let a 'small' amount dx of information flows on the edge vw when w arrives.

Uneven Split

Partition the increase $x_{vw} = dx$ among q_v and q_w by the function f as follows:

$$\begin{aligned} q_v &= q_v + f(\text{Level}(v))dx \\ q_w &= q_w + (1 - f(\text{Level}(v)))dx \end{aligned}$$

Observation: The increase in the value of $q_v + q_w$ is dx.

If Level$(v) \approx 1$ then a large proportion of dx is assigned to q_v as $f(\text{Level}(v)) = e^{\text{Level}(v)-1} \approx 1$
$\implies$ function f doesn't split x_{vw} evenly among q_v and q_w

Objective: Evaluate q values for all vertices in $L \cup R$ after the Waterlevel Algorithm has terminated.

Processing of $w \in R$ resulted in
Case 1: Level$(v) = \sum\limits_{z \in R} x_{vz} = 1$ (v gets saturated)
Case 2: $\sum\limits_{v \in L} x_{vw} = 1$ (w is completely drained)

Next we analyze the sum $q_v + q_w$ for both the cases:

Case 1: v is saturated

We know that $v \in L$ is saturated on the termination of the algorithm, i.e. Level$(v) = 1$

During the course of the algorithm Level of vertex v increased from 0 to 1

Thus for the edge vw: $q_v + q_w \geq q_v = \int\limits_0^1 f(x)dx = \int\limits_0^1 e^{x-1}dx = 1 - \frac{1}{e}$

Case 2: w is drained out

- Vertex w has sent all of its information to its neighbors including v

- Suppose Level$(v) = X$, where $0 \leq X \leq 1$, at the termination of the algorithm.

- By the Key Observation we know that when w was sending information to its neighbors, their Level's were at most X.

- Thus using the fact that f is increasing (therefore, $1 - f$ is decreasing), we have

$$q_w \geq \int\limits_0^1 (1 - f(X))dx = (1 - e^{X-1}) \int\limits_0^1 dx = 1 - e^{X-1}$$

-Therefore, $q_v + q_w \geq \int\limits_0^X f(x)dx + 1 - e^{X-1} = e^{X-1} - \frac{1}{e} + 1 - e^{X-1} = 1 - \frac{1}{e}$

Analysis of Waterlevel Algorithm

Claim

Waterlevel algorithm is $1 - \frac{1}{e}$-competitive for fractional bipartite matching.

Proof: For any edge $e = (vw) \in E$, $q_v + q_w \geq 1 - \frac{1}{e}$

Set $p_v = \frac{e}{e-1} q_v$ for all $v \in L \cup R$

All the constraints of the Dual LP are satisfied.

We know that $\sum_{e=vw \in E} (q_v + q_w) = |M|$ and the objective value of the Dual LP is an upper bound to the objective value of the Primal LP.
Optimal value of the Primal is the size of the optimal fractional matching M^*.
Thus, $\sum_{e=vw \in E} p_v + p_w = \frac{e}{e-1} \sum_{e=vw \in E} q_v + q_w = \frac{e}{e-1} |M| \geq |M^*|$

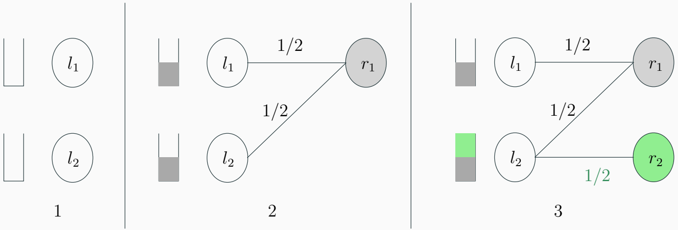
Equivalently, $\frac{|M|}{|M^*|} \geq 1 - \frac{1}{e}$

$\square$

Randomized Matching

Randomized Online Bipartite Matching

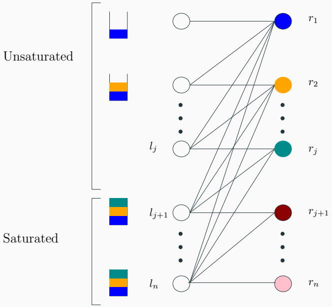
Reconsider Example 1: $G = (V = L \cup R, E)$, where $L = \{l_1, l_2\}$, $R = \{r_1, r_2\}$ and $E = \{r_1 l_1, r_1 l_2, r_2 l_2\}$



1. Greedy Matching: Competitive Ratio $= \frac{1}{2}$
2. Fractional Matching: Competitive Ratio $= \frac{3}{4}$

Randomized Online Bipartite Matching

Reconsider Example 3: $G = (V = L \cup R, E)$, where $L = \{l_1, \ldots, l_n\}$, $R = \{r_1, \ldots, r_n\}$, and $E = \{(l_i, r_j) | i \geq j, \text{ for all } i, j \in \{1, \ldots, n\}\}$



1. Greedy Matching: Competitive Ratio $= \frac{1}{2}$
2. Fractional Matching: Competitive Ratio $= 1 - \frac{1}{e}$

Randomized Online Bipartite Matching

Bipartite graph be $G = (V = L \cup R, E)$
Vertices in $L = \{l_1, \ldots, l_n\}$ are known in advance
Vertices in $R = \{r_1, \ldots, r_n\}$ arrive online along with its incident edges in increasing order of their indices

An edge is either in the matching or it isn't

RANKING Algorithm

The RANKING randomized matching algorithm on the bipartite graph $G = (L \cup R, E)$ is as follows.

RANKING Algorithm [KVV90]

Step 1: For each vertex $v \in L$:
Assign a rank (i.e. a real number) rank(v) selected independently and uniformly at random from $[0, 1]$

Step 2: For each vertex $w \in R$ in order of its appearance:
Match w to its lowest ranked unmatched neighbor (if any) in L

Analysis Framework

1. Analysis via Primal-Dual LP Framework
2. Construct a Dual LP solution (that is randomized)
3. Constraints of the Dual LP may not be satisfied. But they will hold in expectation, i.e. $\sum_{e=(v,w)} E[p_v + p_w] \geq 1$
4. On expected the value of the dual solution is at least the size of an optimum matching $|M^*|$ (Note: Objective value of Dual LP $\geq$ Objective value of Primal LP (and that equals $|M^*|$))

Key Idea 1

Consider the execution of the RANKING.
Let $e = (l_i r_j) \in E$.
When the vertex $r_j \in R$ is considered by RANKING it may or may not be matched to $l_i \in L$ as that depends on whether

1. l_i is unmatched at that moment and
2. among all the unmatched neighbors of r_j, rank(l_i) is the lowest

Key Idea 1 (contd.)

Claim 1

Consider the set $L' = L \setminus \{l_i\}$ and the graph $G' = (L' \cup R, E')$, where E' is obtained from E by excluding the edges incident on l_i. Assume that when RANKING was executed on G', the ranks assigned to each vertex in L' is the same as the ranks assigned to the full set L. Suppose RANKING when executed on G' matches r_j to $l_{i'} \in L$. Let $\Gamma = \text{rank}(l_{i'})$. If $\text{rank}(l_i) < \Gamma$, the vertex $l_i \in L$ is matched in the execution of RANKING to some vertex of R.

Proof: If l_i is already matched in G before the vertex r_j is processed by RANKING, then there is nothing to prove.
For the rest of the proof assume that l_i is not matched even after r_j has been processed by RANKING.
Since the ranks of each vertex in L' is the same as that in L, it follows that the (partial) matching computed by RANKING in G' and G are identical till the vertex r_j is considered.
We know that in G' RANKING matches r_j to $l_{i'} \implies$ In G, RANKING will match r_j to l_i as $\text{rank}(l_i) < \text{rank}(l_{i'}) = \Gamma$. Hence l_i is matched.

□

Key Idea 2

Assume the rank of each vertex in $L' = L \setminus \{l_i\}$ to be same as the rank of the corresponding vertices in L (as generated by RANKING in Step 1), and we assume that $l_i r_j$ is an edge in G.

Claim 2

Execute RANKING on the graphs $G' = (L' \cup R, E')$ and $G = (L \cup R, E)$ in parallel. The set of unmatched vertices in L' is subset of the set of unmatched vertices in L at the start of any step of the 'parallel' execution.

Proof: This is true at the start as the set of matched vertices is empty and $L' \subset L$.
Assume that it holds true when RANKING considered the vertices $r_1, r_2, \ldots, r_{j-1}$.
Consider the step when RANKING is going to consider r_j.
We ask the following question: For two distinct vertices $l_k (\neq l_i)$ and $l_{k'}$ that are among the set of unmatched vertices for both L and L' before r_j was considered, can r_j be matched to l_k in G and to $l_{k'}$ in G' by RANKING?

Proof contd.

For two distinct vertices $l_k (\neq l_i)$ and $l_{k'}$ that are among the set of unmatched vertices for both L and L' before r_j was considered, can r_j be matched to l_k in G and to $l_{k'}$ in G' by RANKING?

This cannot occur.
Before r_j was considered, $l_{k'}$ and l_k are among the set of unmatched vertices for both L and L'.
If $l_{k'}$ is chosen by RANKING in G' as a match for r_j, then rank$(l_{k'}) <$ rank(l_k).
But for G, as $l_{k'}$ was available as an unmatched vertex when r_j was considered by RANKING, there is no reason to match it to l_k which is a higher ranked vertex than $l_{k'}$.
In this step in G either r_j gets matched to l_i or to $l_{k'}$.

$\square$

1. Initialize all $q_v = 0$, where $v \in L \cup R$
2. If an edge $e = vw \in E$, where $v \in L$ and $w \in R$, is identified to be in the matching by RANKING, we set $q_v = f(\text{rank}(v)) = e^{\text{rank}(v)-1}$ and $q_w = 1 - q_v$

Recall that $\Gamma = \text{rank}(l_{i'})$ is the rank of the vertex $l_{i'} \in L'$ that is matched to w in the graph G'.

Claim 3

Let the execution of RANKING on G matches $r_j \in R$ to some vertex $v \in L$. Then $q_{r_j} = 1 - e^{\text{rank}(v)-1} \geq 1 - e^{\Gamma-1}$

Proof of Claim 3

Claim 3

Let the execution of RANKING on G matches $r_j \in R$ to some vertex $v \in L$. Then $q_{r_j} = 1 - e^{\text{rank}(v)-1} \geq 1 - e^{\Gamma - 1}$

Proof: Consider the step when RANKING considers r_j.
As discussed in Claim 2, before r_j is considered, the set of unmatched vertices in L' is a subset of the set of unmatched vertices in L.

This implies that r_j has a unmatched neighbor in G whose rank is at most Γ.

Thus r_j will be matched to a vertex $v \in L$ (may be l_i) with a rank at most Γ.

Since f is an increasing function (and $1 - f$ is decreasing),
$q_{r_j} = 1 - f(\text{rank}(v)) \geq 1 - f(\Gamma)$.

$\square$

Setup for Dual LP (contd.)

As before, set $p_v = \frac{e}{e-1} q_v$ for all vertices $v \in L \cup R$.
Note: $p_v \geq 0$ for all $v \in \{L \cup R\}$ as $q_v \geq 0$

Claim: In expectation, all the Dual LP constraints are satisfied, i.e. for each edge $e = (vw)$, where $v \in L$ and $w \in R$, $E[p_v + p_w] \geq 1$

Two cases to consider: Either e is in the matching reported by RANKING or it isn't.

Case 1: $e \in$ Matching

By definition, $q_v = e^{\text{rank}(v)-1}$ and $q_w = 1 - q_v$. Then $q_v + q_w = 1$ and therefore $p_v + p_w = \frac{e}{e-1} \geq 1$

$\square$

Case 2: $e \notin$ Matching

The analysis is analogous to Case 2 of the WATERLEVEL algorithm.
We need to show that $E[p_v + p_w] \geq 1$

Consider the sets L and L' and the parameter Γ used in Claim 1.
Assume $l_i = v$ and $r_j = w$.
We know that if rank$(v) < \Gamma$ then v is matched by RANKING.

$$\implies E[q_v] \geq \int_0^\Gamma e^{x-1} dx = e^{\Gamma - 1} - \frac{1}{e}$$

By Claim 3 we know that $q_w \geq 1 - e^{\Gamma-1}$

$$E[q_v + q_w] = E[q_v] + E[q_w] \geq e^{\Gamma-1} - \frac{1}{e} + 1 - e^{\Gamma-1} = 1 - \frac{1}{e}$$

Therefore $E[p_v + p_w] = \frac{e}{e-1} E[q_v + q_w] \geq 1$.

$\square$

Main Result

Theorem [KVV90, DJK13]

RANKING Algorithm is $(1 - \frac{1}{e})$-competitive

Proof:

1. For any edge $e \in E$: $E[p_v + p_w] = \frac{e}{e-1} E[q_v + q_w] \geq 1$
2. $\sum_{v \in L \cup R} q_v = |M|$
3. Cost of Primal is the size of an optimal matching M^*
4. Cost of the Dual LP is an upper bound to the cost of the Primal.
5. Cost of Dual: $\sum_{v \in L \cup R} p_v = \frac{e}{e-1} \sum_{v \in L \cup R} q_v = \frac{e}{e-1} |M|$
6. We have $\frac{e}{e-1} |M| \geq |M^*|$
7. $\frac{|M|}{|M^*|} \geq 1 - \frac{1}{e}$

$\square$

BALANCE Algorithm

b-Matching

A bipartite graph $G = (L \cup R, E)$

Vertices in R come in an online manner along with the edges incident to them.

Parameter $b > 0$ is a fixed positive integer.

When a vertex $w \in R$ is revealed to the algorithm, possibly match it one of its neighbors $v \in L$ provided that the number of vertices matched to v so far by the algorithm is $< b$

Whatever decision that we make for w cannot be altered on the arrival of future vertices of R.

BALANCE Algorithm

BALANCE Algorithm

For each vertex $w \in R$ in order of its appearance:

Among all the neighbors of w in L that have been matched $< b$ times, match w to that neighbor (if any) that is matched to the fewest.

AdWords Problem

An alternate view of the problem:

1. Vertices in $L = \{1, 2, \ldots, N\}$ are advertisers, where each of them have a daily budget of $1
2. Each advertiser bids a small amount $\epsilon > 0$ for a set of keywords of their liking.
3. The set R comprises of keyword queries that arrive in an online manner.
4. Each query keyword needs to be assigned to an advertiser (if any) who has bid for that keyword and has some remaining budget $\geq \epsilon$.
5. If the query is assigned to an advertiser, its budget is decreased by ϵ and we generate a revenue of ϵ.

BALANCE algorithm assigns the query to the advertiser who has

1. Bid for that keyword
2. Has remaining budget $\geq \epsilon$
3. Among all those advertisers has the largest remaining budget.

Problem: Maximize the revenue generated by the algorithm, i.e., maximize the sum total of the budget spent by the advertisers.

An Example

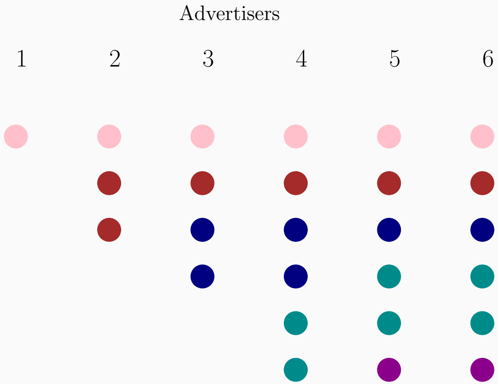


Figure 4: BALANCE with 6 advertisers numbered 1 to 6. Each has a budget of \$1 and can pay for 6 queries. Advertiser i bids for keywords $\{K_1, \ldots, K_i\}$. Thirty-six online queries arrive: first 6 for K_1 (pink dots), followed by next 6 for K_2 (dark red),. . . BALANCE handles 26 queries whereas optimal can handle all 36 queries.

Extending the Example

Setup:

- L has N vertices (advertisers) $1, \ldots, N$, each with a budget of \$1
- N keywords $K_1, \ldots, K_N$
- Advertiser i bids only for the keywords $\{K_1, \ldots, K_i\}$
- Set $\epsilon = \frac{1}{N}$
- Each advertiser can pay for at most N queries

Query Sequence:

- Total of N^2 queries
- First N queries are for the keyword K_1
- Next N queries are for the keyword K_2
- $\ldots$
- $\ldots$
- Last N queries are for the keyword K_N

Offline Revenue $= N$

Analysis of BALANCE

- First N queries corresponding to the keyword K_1 are distributed evenly among all the advertisers

- Next N queries corresponding to the keyword K_2 are distributed among the advertisers $2, \ldots, N$

- In general, N queries for the keyword K_i are distributed evenly among advertisers $i, \ldots, n$ provided that they have sufficient remaining budget

Which queries the advertiser N receives?

- at least one query of type K_1
- at least one query of type K_2
- $\ldots$
- at least $\lfloor \frac{N}{N-i} \rfloor$ queries of type K_i

Analysis of BALANCE (contd.)

When does N-th advertiser runs out of budget ?

$$
\begin{aligned}
N &\leq \left\lfloor \frac{N}{N} \right\rfloor + \left\lfloor \frac{N}{N-1} \right\rfloor + \cdots + \left\lfloor \frac{N}{N-i} \right\rfloor \\
&\leq N \left(\frac{1}{N} + \frac{1}{N-1} + \cdots + \frac{1}{N-i} \right)
\end{aligned}
$$

Condition on i

$i \approx N(1 - \frac{1}{e})$

Recall that n-th Harmonic Number $H_n = \sum_{i=1}^{n} \frac{1}{i} \approx \ln n$.

Express $\frac{1}{N} + \frac{1}{N-1} + \cdots + \frac{1}{N-i}$ as the difference of two Harmonic numbers.

Analysis of BALANCE (contd.)

Competitive Ratio

BALANCE algorithm's competitive ratio is at most $1 - \frac{1}{e}$

Proof: The above example illustrates that the revenue of Balance is $N(1 - \frac{1}{e})$.

Offline, we will assign first N queries to Advertiser 1, next N queries to Advertiser 2, and so on.

Total offline revenue $= N$.

Thus, Competitive Ratio = $\frac{N(1-\frac{1}{e})}{N} = 1 - \frac{1}{e}$.

$\square$

Adwords with BALANCE

BALANCE Algorithm

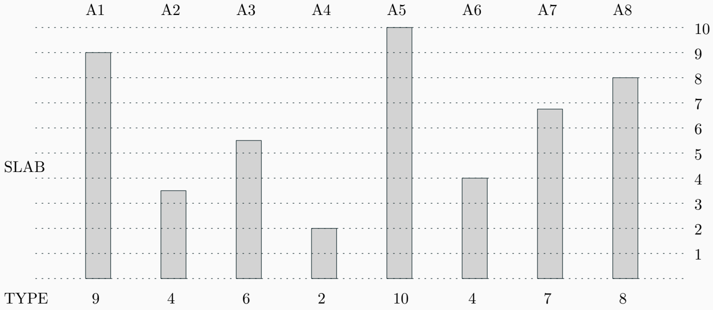
For each vertex $w \in R$ in order of its appearance:
Among all the neighbors of w in L that have been matched $< b$ times, match w to that neighbor (if any) that is matched to the fewest.

1. Advertiser-Keyword framework with bids of value ϵ
2. L corresponds to advertisers
3. R corresponds to query Keyword
4. Match the Keyword to the Advertiser who has bid for the Keyword and has the largest remaining budget
5. Quantized budgets: Each advertiser's budget is discretized in k-slabs of equal value
6. Advertiser spends their budget in increasing order of slab number

Adwords with BALANCE

Assumption: Optimal offline assignment consumes budget of all the advertisers with a total revenue of $1 \times |L| = N$

Advertiser of Type i: If the fraction of the total amount that the advertiser spends during the entire execution of BALANCE is in the range $(\frac{i-1}{k}, \frac{i}{k}]$, where $i \in \{1, \ldots, k\}$



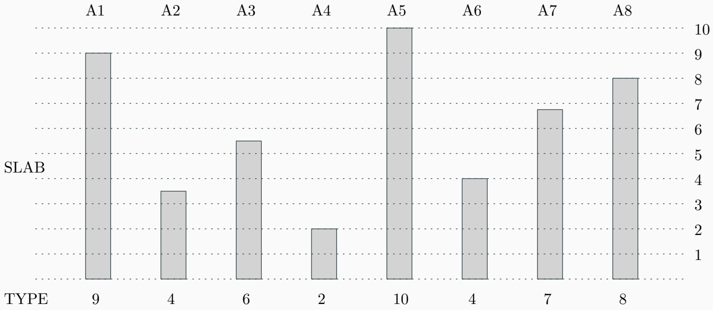
Q & A

Question: Suppose in an optimal assignment a query keyword q is assigned to an advertiser of Type i ($i < k$). From which slab the revenue with respect to q will be generated by BALANCE ?

1. q is assigned to a Type i advertiser in an optimal assignment and its budget isn't completely consumed by BALANCE (as $i < k$)
2. In BALANCE q can't be paid by any slab $> i$ since the queries are assigned to potential advertisers who have consumed the smallest amount of their budget
3. $\implies$ the contribution to the revenue of BALANCE for q comes from a slab $\leq i$

Observation 1

All the query keywords that are assigned by optimal to a Type i advertiser, for some $i < k$, are 'paid' by slabs $\leq i$ in BALANCE.



Observation 2

Let x_i = Numbers of advertisers of Type i

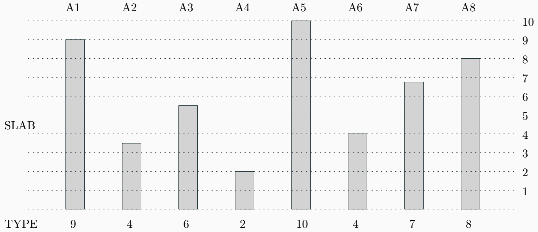
Let β_j = total amount spent from slab j of all the advertisers by BALANCE

Observation 2

$\beta_1 = \frac{|L|}{k} = \frac{N}{k}$, and

$$\beta_j = \frac{N}{k} - \sum_{i=1}^{j-1} \frac{x_i}{k}$$

(Recall that total budget of each advertiser is $1)



Observation 3

Observation 3

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

Proof: Consider $i = 1$. We need to show that $x_1 \le \beta_1$. We know that $\beta_1 = \frac{N}{k}$. All the queries that are assigned to Type 1 advertisers in an optimal assignment need to be paid by slab 1 of the advertisers according to Observation 1.
The total revenue of queries assigned to Type 1 advertisers in an optimal assignment is x_1 (initial budget of $1 times the number of Type 1 advertisers) and this need to be paid by β_1 ($=$ the total amount in Slab 1).
Thus, $x_1 \le \beta_1$.

Consider $k - 1 \ge i \ge 2$. We need to show that
$x_1 + x_2 + \cdots x_i \le \beta_1 + \beta_2 + \cdots \beta_i$.
This follows from the fact that all the queries that are assigned to Types $1, 2, \ldots, i$ advertisers in an optimal assignment need to be paid by Slabs $1, 2, \ldots, i$.

$\square$

Observation 4

The revenue generated by BALANCE is $\geq N(1-\frac{1}{k}) - \sum_{i=1}^{k-1} \frac{k-i}{k} x_i$

Proof: The revenue of BALANCE comes from advertisers of various types.
- An advertiser of Type i, where $i < k$, generates a revenue of $\frac{i}{k}$.
- There are x_i such advertisers and thus the total revenue from Type i advertisers is $\frac{i}{k} x_i$.
- Also, we obtain a revenue of $N - \sum_{i=1}^{k-1} x_i$ from the Type k advertiser.
- But we may loose a revenue of $\frac{1}{k}$ for each advertiser due to ϵ spanning consecutive slabs.
- Putting all this together, the revenue of BALANCE is

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

□

Bounding the Revenue

What is a good lower bound on $N(1-\frac{1}{k}) - \sum_{i=1}^{k-1} \frac{k-i}{k} x_i$?

$$\Leftrightarrow$$

What is an upper bound on $\sum_{i=1}^{k-1} \frac{k-i}{k} x_i$?

It results in the following Linear Program:

Primal LP

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

Subject to:

For all $i \in \{1, \ldots, k-1\}$: $\sum_{j=1}^{i} x_j \leq \sum_{j=1}^{i} \beta_j$ (Observation 4)

For all $i \in \{1, \ldots, k\} : x_i \geq 0$

Condition $\sum_{j=1}^{i} x_j \leq \sum_{j=1}^{i} \beta_j$ can be expressed as follows:

$$\begin{aligned}
\sum_{j=1}^{i} x_j &\leq \sum_{j=1}^{i} \beta_j \\
&\leq \sum_{j=1}^{i} (\frac{N}{k} - \sum_{l=1}^{j-1} \frac{x_l}{k}) \\
&= \frac{i}{k} N - \sum_{j=1}^{i} \sum_{l=1}^{j-1} \frac{x_l}{k} \\
&= \frac{i}{k} N - \sum_{j=1}^{i} \frac{i-j}{k} x_j
\end{aligned}$$

Bounding the Revenue (contd.)

By rearranging terms for x_j, we have

$$\sum_{j=1}^{i}(1+\tfrac{i-j}{k})x_j \leq \tfrac{i}{k}N$$

Thus, we can express the Primal LP as follows:

Primal LP

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

Subject to:

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

For all $i \in \{1, \ldots, k\} : x_i \geq 0$

Primal-Dual LP

Primal LP

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

Subject to:

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

For all $i \in \{1, \ldots, k\} : x_i \geq 0$

Dual LP

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

Subject to:

For all $i \in \{1, \ldots, k-1\}$: $\sum_{j=i}^{k-1} (1 + \frac{j-i}{k}) y_j \geq \frac{k-i}{k}$

For all $i \in \{1, \ldots, k-1\} : y_i \geq 0$

Reasoning

Consider the Dual LP constraint with respect to the Primal LP variable x_1. We will need that

$$y_1 + y_2(1 + \frac{1}{k}) + y_3(1 + \frac{2}{k}) + \cdots + y_{k-1}(1 + \frac{k-2}{k}) \geq \frac{k-1}{k}$$

This can be expressed as

$$\sum_{j=1}^{k-1}(1 + \frac{j-1}{k})y_j \geq \frac{k-1}{k}$$

In general, for the i-th variable x_i, we have the Dual LP constraint

$$\sum_{j=i}^{k-1}(1 + \frac{j-i}{k})y_j \geq \frac{k-i}{k}$$

Complementary Slackness

Feasible Solution $\rightarrow$ Optimal Solution

How to determine that a feasible solution is optimal in the Primal-Dual LP framework?

Complementary Slackness Condition

If feasible solutions x and y to Primal LP $(\max cx, Ax \leq b, x \geq 0)$ and Dual LP $(\min by, A^T y \geq c, y \geq 0)$ satisfy $\forall i : (b_i - \sum_j a_{ij} x_j) y_i = 0$ and $\forall j : (\sum_i a_{ij} y_j - c_j) x_j = 0$ then they are also optimal.

An Example

	Primal	Dual
Objective	Max $x_1 + x_2 + x_3$	Min $6y_1 + 4y_2 + 10y_3$
Constraints	$2x_1 + 3x_2 + x_3 \leq 6$	$2y_1 + y_2 + 3y_3 \geq 1$
	$x_1 + x_2 - 7x_3 \leq 4$	$3y_1 + y_2 - y_3 \geq 1$
	$3x_1 - x_2 + 5x_3 \leq 10$	$y_1 - 7y_2 + 5y_3 \geq 1$
Non-negativity	$x_1, x_2, x_3 \geq 0$	$y_1, y_2, y_3 \geq 0$

Feasible Primal LP solution: $x = (0, \frac{5}{4}, \frac{9}{4})$
Feasible Dual LP solution $y = (\frac{3}{8}, 0, \frac{1}{8})$

Check Complementary Slackness Conditions ($\forall i : (b_i - \sum_j a_{ij}x_j)y_i = 0$ and $\forall j : (\sum_i a_{ij}y_j - c_j)x_j = 0$)

Primal: Ineq. 1 & 3 are tight. Slack in the 2nd Ineq. but $y_2 = 0$

Dual: Ineq. 2 & 3 are tight. Slack in 1st Ineq. but $x_1 = 0$

$\implies$ $x = (0, \frac{5}{4}, \frac{9}{4})$ and $y = (\frac{3}{8}, 0, \frac{1}{8})$ are optimal.

Feasible Solution for Primal LP

Primal LP: Max $\sum_{i=1}^{k-1} \frac{k-i}{k} x_i$

Subject to:

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

For all $i \in \{1, \dots, k\} : x_i \geq 0$

Set $x_1 = \frac{N}{k}$, $x_2 = \frac{N}{k}(1 - \frac{1}{k})$, $\dots$, $x_i = \frac{N}{k}(1 - \frac{1}{k})^{i-1}$, $\dots$, $x_k = \frac{N}{k}(1 - \frac{1}{k})^{k-1}$

They are derived by setting $\sum_{j=1}^{i} (1 + \frac{i-j}{k}) x_j = \frac{i}{k} N$ and solving for x_i for $i = 1, 2, \dots, k-1$.

The assignment $x_i = \frac{N}{k}(1 - \frac{1}{k})^{i-1}$ is a feasible solution for Primal LP

Feasible Solution for Dual LP

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

Subject to:

For all $i \in \{1, \ldots, k-1\}$: $\sum_{j=i}^{k-1} (1 + \frac{j-i}{k})y_j \geq \frac{k-i}{k}$

For all $i \in \{1, \ldots, k-1\} : y_i \geq 0$

Set $\sum_{j=i}^{k-1} (1 + \frac{j-i}{k})y_j = \frac{k-i}{k}$

We obtain $y_{k-1} = \frac{1}{k}$, $y_{k-2} = \frac{1}{k}(1 - \frac{1}{k})$, $y_{k-3} = \frac{1}{k}(1 - \frac{1}{k})^2, \ldots,$
$y_{k-i} = \frac{1}{k}(1 - \frac{1}{k})^{i-1}, \ldots, y_1 = \frac{1}{k}(1 - \frac{1}{k})^{k-2}$

All y_i's are feasible.

Optimality

The assignments x and y are optimal solutions for the Primal and Dual LPs.

Proof: Assignment of x and y are feasible for Primal and Dual LPs. Since all the inequalities are equalities, there is no slack, and thus the complementary slackness conditions hold. This implies that not only x and y are feasible, but they are also optimal solutions for Primal and Dual LPs.

A Summation

$$\sum_{i=1}^{k-1} \left(\frac{k-i}{k}\right) x_i = N\left(1 - \frac{1}{k}\right)^k$$

Proof:

$$\begin{aligned}
\sum_{i=1}^{k-1} \left(\frac{k-i}{k}\right) x_i &= \sum_{i=1}^{k-1} \left(\frac{k-i}{k}\right)\left(\frac{N}{k}\right)\left(1 - \frac{1}{k}\right)^{i-1} \\
&= \frac{N}{k^2}\left[\sum_{i=1}^{k-1} k\left(1 - \frac{1}{k}\right)^{i-1} - \sum_{i=1}^{k-1} i\left(1 - \frac{1}{k}\right)^{i-1}\right] \\
&= \frac{N}{k^2}\left[k\left(\frac{1 - \left(1 - \frac{1}{k}\right)^{k-1}}{1 - \left(1 - \frac{1}{k}\right)}\right) - \frac{k^2}{k-1}\left(\left(1 - \frac{1}{k}\right)^k - k\left(2\left(1 - \frac{1}{k}\right)^k - 1\right) - 1\right)\right] \\
&= \frac{N}{k^2}\left[k^2\left(1 - \left(1 - \frac{1}{k}\right)^{k-1}\right) - \frac{k^2}{k-1}\left((1-2k)\left(1 - \frac{1}{k}\right)^k + k - 1\right)\right] \\
&= N\left[\frac{(k-1)\left(1 - \left(1 - \frac{1}{k}\right)^{k-1}\right) - (1-2k)\left(1 - \frac{1}{k}\right)^k - k + 1}{k-1}\right] \\
&= N\left[\frac{-(k-1)\left(1 - \frac{1}{k}\right)^{k-1} - (1-2k)\left(1 - \frac{1}{k}\right)^k}{k-1}\right] \\
&= N\left(1 - \frac{1}{k}\right)^k
\end{aligned}$$

Cost of Optimal Solution

Main Claim

BALANCE is $(1 - \frac{1}{e})$-competitive

Proof:

1. As $k \to \infty$, $\left(1 - \frac{1}{k}\right)^k \to \frac{1}{e}$
2. Upper bound on the value of $\sum_{i=1}^{k-1} \frac{k-i}{k} x_i = \frac{N}{e}$
3. Revenue of BALANCE is atleast $N(1 - \frac{1}{k}) - \sum_{i=1}^{k-1} \frac{k-i}{k} x_i \geq N(1 - \frac{1}{k}) - \frac{N}{e} \approx N(1 - \frac{1}{e})$ for large values of k.
4. Competitive ratio is $1 - \frac{1}{e}$

References

1. Richard Karp, Umesh Vazirani, and Vijay Vazirani, An optimal algorithm for on-line bipartite matching. ACM-STOC, 1990.
2. Nikhil Devanur, Kamal Jain, and Robert Kleinberg, Randomized primal-dual analysis of RANKING for online bipartite matching, ACM-SIAM SODA 2013.
3. Aranyak Mehta, Amin Saberi, Umesh V. Vazirani, Vijay V. Vazirani: AdWords and generalized online matching. Journal of ACM 54(5): 22 (2007)
4. Bala Kalyanasundaram, Kirk Pruhs: An optimal deterministic algorithm for online b-matching. Theoretical Computer Science 233(1-2): 319-325 (2000)