A Primal-Dual Approach for Online Problems

Nikhil Bansal

Online Algorithms

Input revealed in parts.

Algorithm has no knowledge of future.

Scheduling, Load Balancing, Routing, Caching
Finance, Machine Learning …

Competitive ratio = $\max_I \frac{On(I)}{Opt(I)}$

Expected Competitive ratio = $\max_I \frac{E[On(I)]}{Opt(I)}$

Some classic problems

The Ski Rental Problem

- Buying costs $B.
- Renting costs $1 per day.

Problem:

- Number of ski days is not known in advance.

Goal: Minimize the total cost.

Deterministic: 2
Randomized: $e/(e-1) \approx 1.58$

Online Virtual Circuit Routing

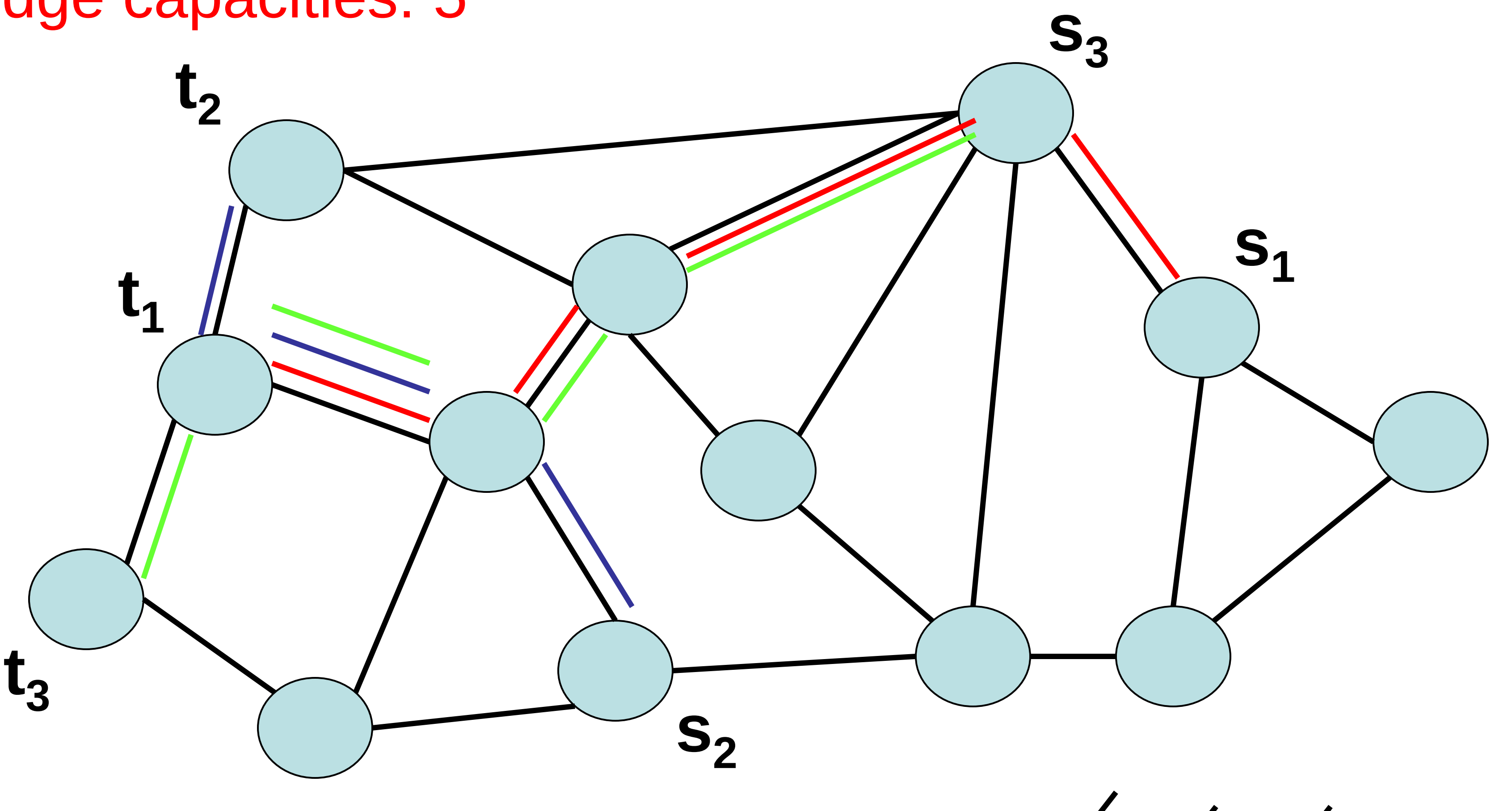
Network graph $G=(V, E)$
capacity function $u: E \rightarrow Z^+$

Requests: $r_i = (s_i, t_i)$

- **Problem:** Connect s_i to t_i by a path, or reject the request.
- Reserve one unit of bandwidth along the path.
- **No re-routing is allowed**.
- **Load:** ratio between reserved edge bandwidth and edge capacity.
- **Goal:** Maximize the total throughput.

Virtual Circuit Routing - Example

Edge capacities: 5



Maximum Load: ~~0~~ ~~1/5~~ ~~2/5~~ 3/5

Virtual Circuit Routing

Key decision:

1) Whether to choose request or not?
2) How to route request?

O(log n)-congestion, O(1)-throughput [Awerbuch Azar Plotkin 90's]

Various other versions and tradeoffs.

Main idea: Exponential penalty approach

length (edge) = exp (congestion)

Decisions based on length of shortest (s_i,t_i) path

Clever potential function analysis

The Paging/Caching Problem

Set of pages {1,2,…,n} , cache of size k<n.

Request sequence of pages 1, 6, 4, 1, 4, 7, 6, 1, 3, …

a) If requested page already in cache, no penalty.

b) Else, cache miss. Need to fetch page in cache (possibly) evicting some other page.

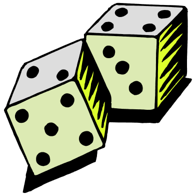
Goal: Minimize the number of cache misses.

Key Decision: Upon a request, which page to evacuate?

Previous Results: Paging

Paging (Deterministic) [Sleator Tarjan 85]:

- Any det. algorithm $\geq$ k-competitive.
- LRU is k-competitive (also other algorithms)



Paging (Randomized):

- Rand. Marking O(log k) [Fiat, Karp, Luby, McGeoch, Sleator, Young 91].
- Lower bound H_k [Fiat et al. 91], tight results known.

Do these problems have anything in common?

An Abstract Online Problem

min $3x_1 + 5x_2 + x_3 + 4x_4 + \ldots$

$$\left.\begin{aligned} 2x_1 + x_3 + x_6 + \ldots &\geq 3 \\ x_3 + x_{14} + x_{19} + \ldots &\geq 8 \\ x_2 + 7x_4 + x_{12} + \ldots &\geq 2 \end{aligned}\right\}$$

Covering LP
(non-negative entries)

Goal: Find feasible solution x^* with min cost.

Requirements:

1) Upon arrival constraint must be satisfied
2) Cannot decrease a variable.

Example

$\min \; x_1 + x_2 + \ldots + x_n$

$$
\begin{aligned}
x_1 + x_2 + x_3 + \ldots + x_n &\geq 1 \\
x_2 + x_3 + \ldots + x_n &\geq 1 \\
x_3 + \ldots + x_n &\geq 1 \\
\ldots \\
x_n &\geq 1
\end{aligned}
$$

Set all x_i to 1/n

Increase $x_2, x_3, \ldots, x_n$ to 1/n-1

…

Increase x_n to 1

Online $\geq$ ln n $\quad (1+1/2+1/3+\ldots+1/n)$

Opt = 1 ($x_n=1$ suffices)

The Dual Problem

$\max \quad 3\,y_1 + 5\,y_2 + y_3 + 4\,y_4 + \ldots$

$$
\begin{aligned}
2\,y_1 &\leq 3 \\
y_1 &\leq 8 \\
y_1 &\leq 2
\end{aligned}
$$

Packing LP
(non-negative entries)

Goal: Find y^* with max cost.

Requirements:
1) Variables arrive sequentially
2) At step t, can only modify y(t)

All previous problems can be expressed as Covering/Packing LP

Ski Rental – Integer Program

$$x = \begin{cases} 1 \text{ - Buy} \\ 0 \text{ - Don't Buy} \end{cases} \qquad z_i = \begin{cases} 1 \text{ - Rent on day i} \\ 0 \text{ - Don't rent on day i} \end{cases}$$

$$\min Bx + \sum_{i=1}^{k} z_i$$

Subject to:

For each day i: $x + z_i \geq 1$ (either buy or rent)

$$x, z_i \in \{0,1\}$$

Routing – Linear Program

$y(r_i, p)$ = Amount of bandwidth allocated for r_i on path p

$P(r_i)$ - Available paths to serve request r_i

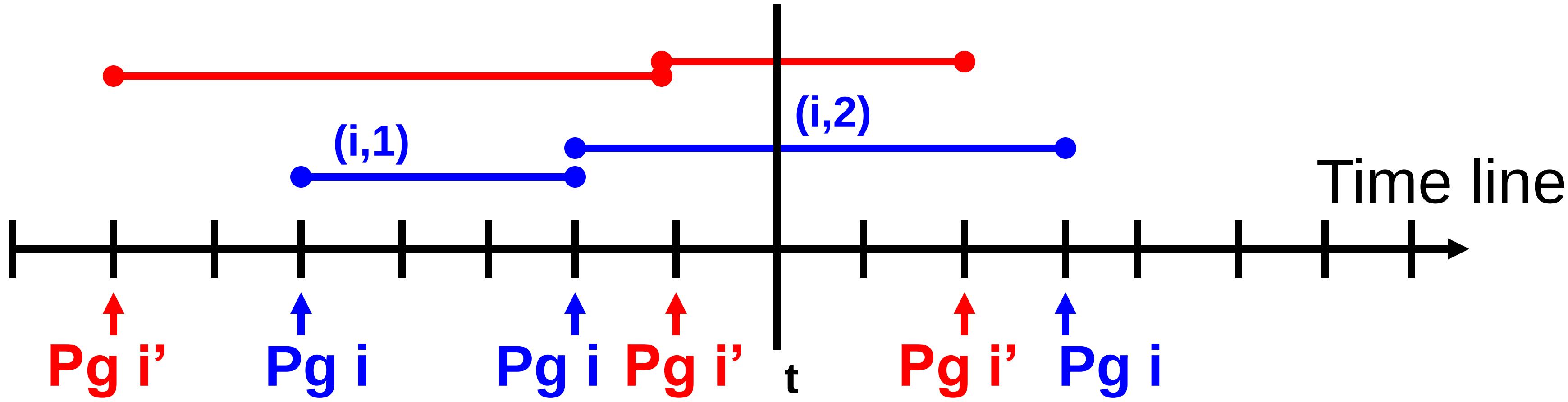
$$\max \sum_{r_i} \sum_{p \in P(r_i)} y(r_i, p)$$

s.t:

For each r_i: $\displaystyle\sum_{p \in P(r_i)} y(r_i, p) \leq 1$

For each edge e: $\displaystyle\sum_{r_i} \sum_{p \in P(r_i) \mid e \in p} y(r_i, p) \leq u(e)$

Paging – Linear Program



If interval not present, then cache miss.

At any time t, can have at most k such intervals.

□ at least n-k intervals must be absent

n: number of distinct pages

x(i,j): How much interval (i,j) evacuated thus far

$0 \leq x(i,j) \leq 1$

Cost $= \sum_i \sum_j x(i,j)$

$\sum_{i:\, i \neq p_t} x(i,r(i,t)) \geq n-k \quad \forall t$

What can we say about the abstract problem ?

General Covering/Packing Results

LP Connection → Powerful Techniques (duality)

For a **general covering/packing** matrix [BN05] :

Covering:

- Competitive ratio O(log n) (n – number of variables).

Packing:

- Competitive ratio O(log n + log [a(max)/a(min)])
  a(max), a(min) – max/min non-zero entry

Remarks:

- Results are tight.
- Can add “box” constraints to covering LP (e.g. $x \le 1$)

General Covering/Packing Results

For a {0,1} covering/packing matrix: [Buchbinder Naor 05]

- **Competitive ratio O(log D)**
- Can get e/e-1 for ski rental and other problems.

(D – max number of non-zero entries in a constraint).

Remarks:

- Number of constraints/variables can be exponential.
- There can be a tradeoff between the competitive ratio and the factor by which constraints are violated.

Fractional solution $\rightarrow$ randomized algorithm (online rounding)

Consequences

Very powerful framework.

Unified and improved several previous results.

Weighted Paging: O(log k) guarantee [B., Buchbinder, Naor 07]

Previously, o(k) known only for the case of 2 weights [Irani 02]

$O(\log^2 k)$ for Generalized Paging (arbitrary weights and sizes)

[B., Buchbinder, Naor 08]

A poly-logarithmic guarantee for the k-server problem

[B., Buchbinder, Madry, Naor 11]

Rest of the Talk

1) Overview of LP Duality, offline P-D technique
2) Derive Online Primal Dual (very natural)
3) Further Extensions + k-server Problem

Duality

Min $3 x_1 + 4 x_2$

$x_1 + x_2 >= 3$
$x_1 + 2 x_2 >= 5$

Want to convince someone that there is a solution of value 12.

Easy, just demonstrate a solution,
$x_2 = 3$

Duality

Min $3x_1 + 4x_2$

$x_1 + x_2 >= 3$

$x_1 + 2x_2 >= 5$

Want to convince someone that there is no solution of value 10.

How?

2 * first eqn + second eqn

$3x_1 + 4x_2 >= 11$

LP Duality Theorem: This seemingly ad hoc trick always works!

LP Duality

Min $c_j\, x_j$

$\sum_j a_{ij}\, x_j \geq b_i$ $\xrightarrow[(y \geq 0)]{\text{Linear combination}}$

So, for any $y \geq 0$ satisfying $\sum_i a_{ij}\, y_i \leq c_j$ for all i

$\sum_j x_j\, c_j \geq \sum_i y_i\, b_i$

Dual cost

Dual LP

Equality when Complementary Slackness

i.e. $y_i > 0$ (only if corresponding primal constraint is tight)

$x_i > 0$ (only if corresponding dual constraint is tight)

Offline Primal-Dual Approach

min cx | max b y
$Ax \geq b$ | $A^t y \leq c$
$x \geq 0$ | $y \geq 0$

Generic Primal Dual Algorithm:

0) Start with x=0, y=0   (primal infeasible, dual feasible)

1) Increase dual and primal together,

   s.t. if dual cost increases by 1, primal increases by $\leq$ c

2) If both dual and primal feasible $\Rightarrow$ c approximate solution

Key Idea for Online Primal Dual

Primal: Min $\sum_i c_i x_i$

Dual

Step t, new constraint:

$a_1x_1 + a_2x_2 + \ldots + a_jx_j \geq b_t$

New variable y_t

$+ b_t y_t$ in dual objective

How much: Δx_i ?

$y_t \rightarrow y_t + 1$ (additive update)

Δ primal cost = $\sum_i c_i(\Delta x_i) \; =$

$\leq b_t$ = Δ Dual Cost

dx/dy proportional to x so, x varies as exp(y)

How to initialize

A problem: dx/dy is proportional to x, but $x=0$ initially.

So, x will remain equal to 0 ?

Answer: Initialize to $1/n$.

When: Complementary slackness tells us that $x > 0$ only if dual constraint corresponding to x is tight.

Set $x=1/n$ when its dual constraint becomes tight.

The Algorithm

Min $\sum_j c_j x_j$
$\sum_j a_{ij} x_j \geq b_i$

Max $\sum_i b_i y_i$
$\sum_i a_{ij} y_i \leq c_j$

On arrival of i-th constraint, Initialize y_i=0 (dual var. for constraint)

If current constraint unsatisfied, gradually increase y_i

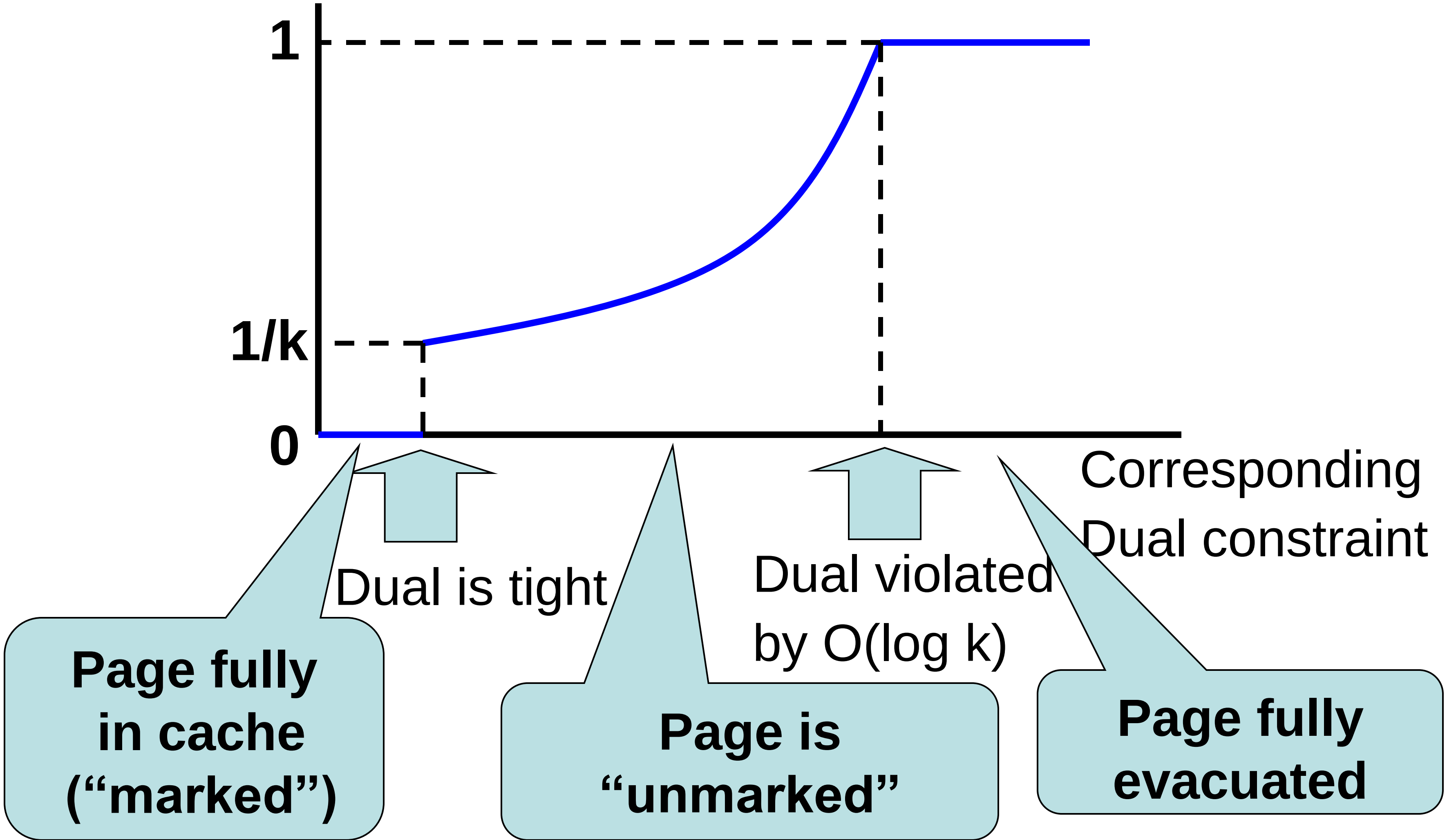
If x_j =0, set $x_j = 1/n$ when $\sum_i a_{ij} y_i = c_j$

else update x_j as $1/n \cdot \exp((\sum_i a_{ij} y_i / c_j) - 1)$

1) Primal Cost $\leq$ Dual Cost
2) Dual solution violated by at most O(log n) factor.

Example: Caching

x_p: fraction of p missing from cache



Part 2: Rounding

Primal dual technique gives fractional solution.

Problem specific rounding/interpretation:

1) Easy for ski rental (value of x, is prob. of buying by then)

2) Routing: Can derandomize online using pessimistic estimator or other techniques

3) Caching (tricky): Gives probability distribution on pages,
Actually want probability distribution on cache states.

Beyond Packing/Covering LPs

Extended Framework

Limitations of current framework
1. Only covering or packing LP
2. Variables can only increase.

Cannot impose: $a \geq b$ or $a \geq b_1 - b_2$

Problem with monotonicity:

Predicting with Experts: Do as well as best expert in hindsight
n experts: Each day, predict rain or shine.

Online $\leq$ Best expert $(1+\varepsilon) + O(\log n)/\varepsilon$ (low regret)
In any LP, $x_{i,t}$ = Prob. of expert i at time t.

Recent Extensions

Handle somewhat more general settings

[B., Buchbinder, Naor 10] (can capture expert learning)

(Algorithms based on similar insight)

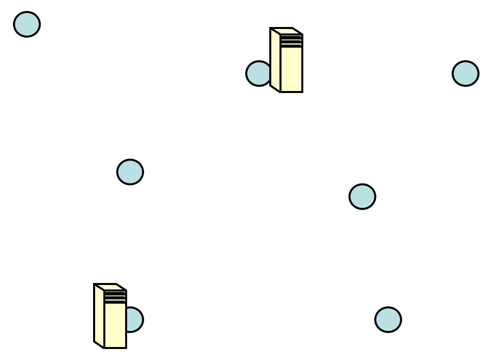
Potential function interpretation [B., Buchbinder, Naor 11]

(Useful when primal-dual view is messy)

K-Server Problem

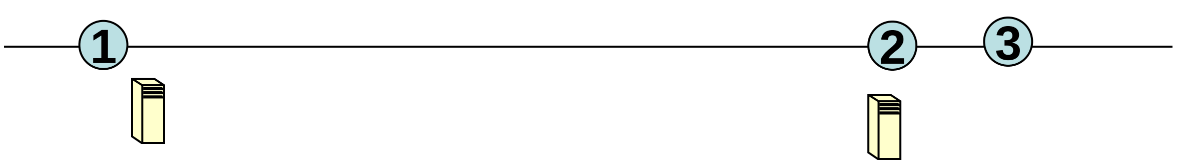
The k-server Problem

- k servers lie in an n-point metric space.
- Requests arrive at metric points.
- To serve request: Need to move some server there.



Goal: Minimize total distance traveled.

Objective: Competitive ratio.

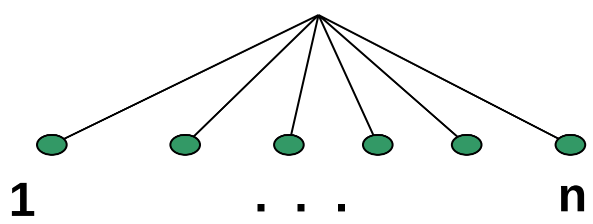


Move Nearest Algorithm

The Paging/Caching Problem

K-server on the uniform metric.

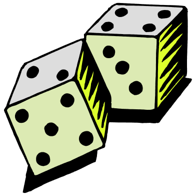
Server on location p = page p in cache



Previous Results: Paging

Paging (Deterministic) [Sleator Tarjan 85]:

- Any deterministic algorithm >= k-competitive.
- LRU is k-competitive (also other algorithms)



Paging (Randomized):

- Rand. Marking O(log k) [Fiat, Karp, Luby, McGeoch, Sleator, Young 91].
- Lower bound H_k [Fiat et al. 91], tight results known.

K-server conjecture

[Manasse-McGeoch-Sleator ’88]:
There exists k competitive algorithm on any metric space.

Initially no f(k) guarantee.

Fiat-Rababi-Ravid’90: exp(k log k)

…

Koutsoupias-Papadimitriou’95: 2k-1

Chrobak-Larmore’91: k for trees.

Randomized k-server Conjecture

There is an O(log k) competitive algorithm for any metric.

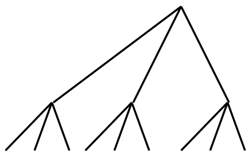
Uniform Metric: log k

Polylog for very special cases (uniform-like)

Line: $n^{2/3}$ [Csaba-Lodha'06]

$\exp(O(\log n)^{1/2})$ [Bansal-Buchbinder-Naor'10]

Depth 2-tree: No o(k) guarantee



Result

Thm [B.,Buchbinder,Madry,Naor 11]: There is an $O(\log^2 k \log^3 n)$ competitive* algorithm for k-server on any metric with n points.

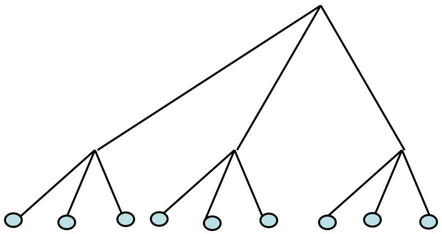
*** Hiding some log log n terms**

Our Approach

Hierarchically Separated Trees (HSTs) [Bartal 96].

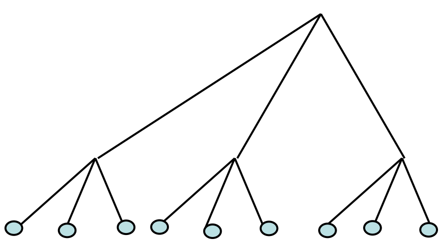
Any Metric

O(log n)



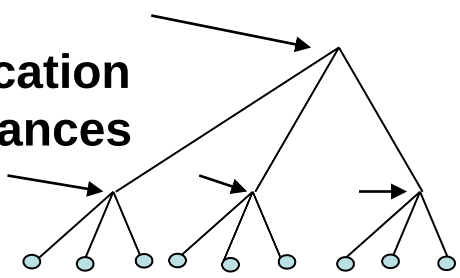
Allocation Problem (uniform metrics): [Cote-Meyerson-Poplawski'08]

(decides how to distribute servers among children)



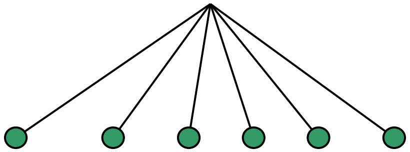
K-server on HST

Allocation instances



Allocation Problem

Uniform Metric



At each time t, request arrives at some location i

Request = ($h_t(0),\ldots,h_t(k)$) [monotone: $h(0) \geq h(1) \ldots \geq h(k)$]

Upon seeing request, can reallocate servers

Hit cost = $h_t(k_i)$ [k_i : number of servers at i]

Total cost = Hit cost + Move cost

Eg: Paging = cost vectors (∞,0,0,…,0)

*Total servers k(t) can also change (let's ignore this)

Allocation to k-server

Thm [Cote-Poplawski-Meyerson]: An online algorithm for allocation

s.t. for any $\varepsilon > 0$,

i) Hit Cost (Alg) $\leq$ $(1+\varepsilon)$ OPT

ii) Move Cost (Alg) $\leq$ $\beta(\varepsilon)$ OPT

gives $\approx$ $O(d\ \beta(1/d))$ competitive k-server alg. on depth d HSTs

d = log (aspect ratio)   So, β = poly($1/\varepsilon$) polylog(k,n) suffices

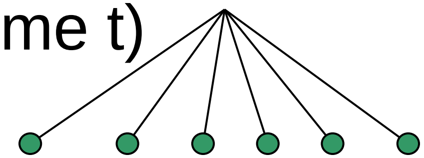
*HSTs need some well-separatedness

*Later, we do tricks to remove dependence on aspect ratio

We do not know how to obtain such an algorithm.

Fractional Allocation Problem

$x_{i,j}$: prob. of having j servers at location i (at time t)



$\sum_j x_{i,j} = 1$ (prob. distribution)

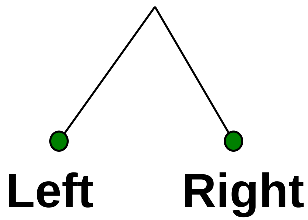
$\sum_i \sum_j j\, x_{i,j} \leq k$ (global server bound)

Cost: Hit cost with $h(0),\ldots,h(k) = \sum_j x_{i,j} h(j)$

Moving ε mass from (i,j) to (i,j') costs $\varepsilon\, |j'-j|$

Surprisingly, fractional allocation does not give good randomized alg. for allocation problem.

A gap example



Allocation Problem on 2 points

Requests alternate on locations.

Left: (1,1,…,1,0) Right: (1,0,…,0,0)

Any integral solution must pay $\Omega(T)$ in T steps.

Claim: Fractional Algorithm pays only T/(2k) .

$X_{L,0} = 1/k$ $x_{L,k} = 1\text{-}1/k$

$X_{R,1} = 1$

No move cost. Hit cost of 1/k on left requests.

Fractional Algorithm Suffices

Thm (Analog of Cote et al): Suffices to have fractional allocation algorithm with $(1+\varepsilon,\beta(\varepsilon))$ guarantee.

Gives a fractional k-server algorithm on HST

Thm (Rounding): Fractional k-server alg. on HSTs -> Randomized Alg. with $O(1)$ loss.

Thm (Frac. Allocation): Design a fractional allocation algorithm with $\beta(\varepsilon) = O(\log (k/\varepsilon))$.

Concluding Remarks

Primal Dual: Unifying idea in many online algorithms.

Close connections to multiplicative updates method.

Related work for regret minimization [Shwartz-Singer 07]

Towards a unified framework for problems dealing with uncertainty ?

Thank you