Columbia University
IN THE CITY OF NEW YORK

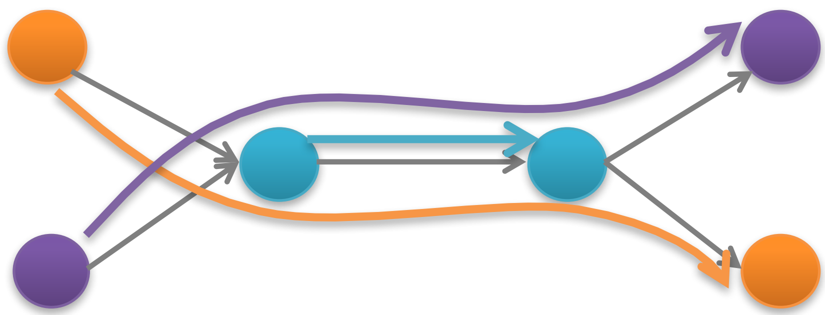
A Fast Distributed Stateless Algorithm for α-Fair Packing Problems

Jelena Marašević, Cliff Stein, and Gil Zussman
Columbia University, New York

ICALP'16, July 2016

Fair Resource Allocation: Applications

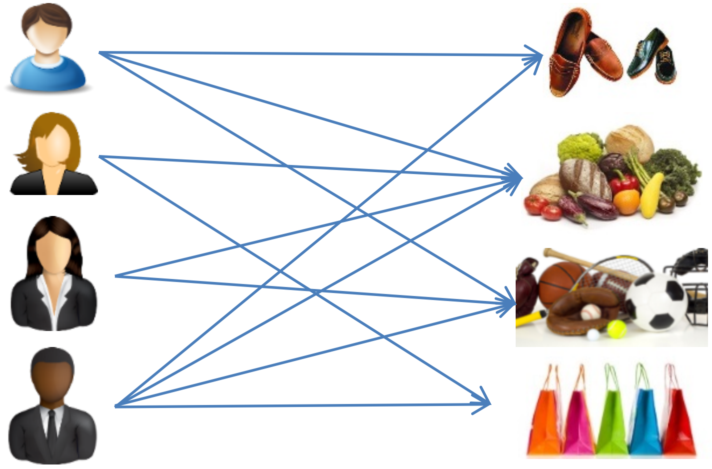
Network congestion control



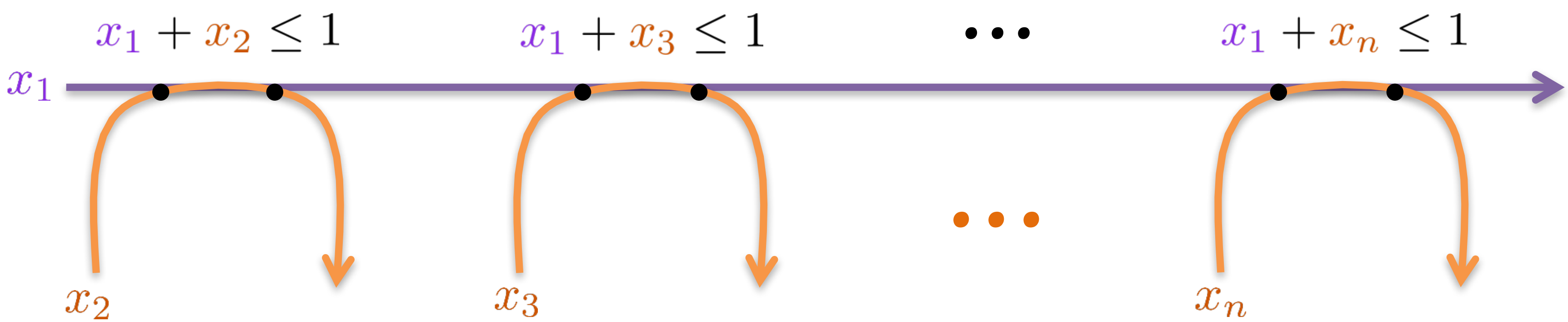
Resource management in Datacenters

Healthcare scheduling

Market equilibria problems



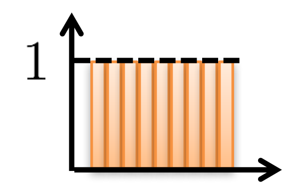
Fair Resource Allocation: Motivation



How to allocate nonnegative $x_1, x_2, \ldots, x_n$? $(x_2 = x_3 = \ldots = x_n \equiv y)$

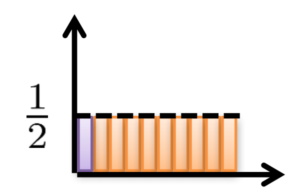
- Maximize efficiency:

$$x_1 = 0,\ y = 1, \qquad \sum_{j=1}^{n} x_j = n - 1$$



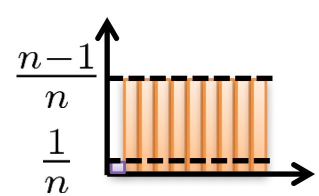
- Maximize fairness:

$$x_1 = y = \tfrac{1}{2}, \qquad \sum_{j=1}^{n} x_j = \tfrac{n}{2}$$



- Trade-off efficiency and fairness:

$$x_1 = \tfrac{1}{n},\ y = \tfrac{n-1}{n}, \quad \sum_{j=1}^{n} x_j = \frac{n^2 - 2n + 2}{n}$$



α-Fairness

Definition (weighted α −fairness) [MW'00]. Given a convex and compact feasible region $\mathcal{R} \subseteq \mathbb{R}^n_+$, $\alpha \geq 0$, and a positive vector of weights w, a vector $x^* \in \mathcal{R}$ is weighted α −fair if for any other $x \in \mathcal{R}$:

$$\sum_{j=1}^n w_j \frac{x_j - x_j^*}{(x_j^*)^\alpha} \leq 0.$$

Lemma [MW'00]. A vector $x^* \in \mathcal{R}$ is weighted α −fair if and only if it solves the following optimization problem:

$$\begin{array}{ll} \mathbf{max} & \sum_{j=1}^{n} w_j f_\alpha(x_j) \\ \mathbf{s.t.} & x \in \mathcal{R} \end{array}$$

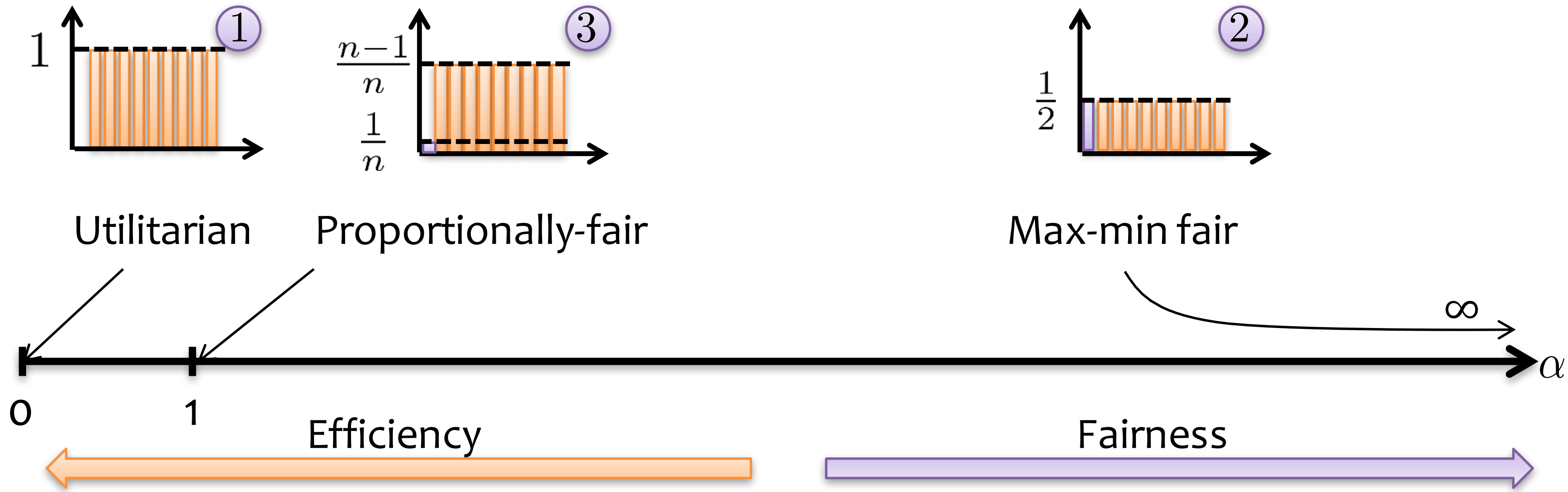
, where

$$f_\alpha(x_j) = \begin{cases} \ln(x_j), & \text{if } \alpha = 1 \\ \frac{x_j^{1-\alpha}}{1-\alpha}, & \text{if } \alpha \neq 1 \end{cases}.$$

$$\left(\frac{df_\alpha(x_j)}{dx_j} = \frac{1}{{x_j}^\alpha} \right)$$

[MW'00] Mo, Jeonghoon, and Walrand, Jean, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on Networking (ToN)* 8.5 (2000): 556-567.

Measuring α-Fairness



Quantification of tradeoffs between efficiency and fairness:

- Axiomatic theory of fairness [Lan et al. 2010]
- Relative loss [Bertsimas et al. 2012]:
  - In efficiency (sum of allocated values)
  - In fairness (minimum allocated value)

α-Fair Packing

$$\begin{aligned} \textbf{max} \quad & \sum_{j=1}^{n} w_j f_\alpha(x_j) \\ \textbf{s.t.} \quad & \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

, where

$$f_\alpha = \begin{cases} \ln(x_j), & \text{if } \alpha = 1 \\ \frac{x_j^{1-\alpha}}{1-\alpha}, & \text{if } \alpha \neq 1 \end{cases}$$

$$\mathbf{A} \geq \mathbf{0}, \quad \mathbf{b} > \mathbf{0}$$

- The focus is on distributed algorithms with asynchronous updates

Main Result

An ε-approximation algorithm that is:

- Distributed;
- Stateless:
  - Asynchronous;
  - Self-stabilizing;
  - Dynamic – supports constant # of variable/constraint insertions/deletions

Convergence time:

- Poly-log in the input size and polynomial in ε^{-1}

Related Work

- (Sequential) convex programming can give $\mathrm{poly}(N, \log(\varepsilon^{-1}))$
- **Max-min fairness** [Megiddo 1974], [Bertsekas and Gallager 1992], [Kleinberg et al. 1999], [Radunovic and Le-Boudec 2007]
- **Packing LPs** [Plotkin, Shmoys, Tardos 1991], [Luby and Nisan 1993], [Awerbuch and Khandekar 2008], [Allen-Zhu and Orecchia 2015, 2016], [Wang et al. 2016]
  - Only linear objectives
- **Network congestion control** [Kelly et al. 1998], [Mo and Walrand 2000], [Low et al. 2002], [Sarkar 2004], [Yi and Chiang 2008]
  - No guaranteed convergence time
- **Network utility maximization** [Mosk-Aoyama et al. 2007], [Beck et al. 2014]
- Discrete tatonnement for Eisenberg-Gale markets [Cheung et al. 2013]

Outline

- Introduction
- Model, Scaling, and Preliminaries
- Algorithm
- Convergence
- Asymptotic behavior of α-fair allocations

Scaled Problem

$$\begin{aligned} \mathbf{max} \quad & \sum_{j=1}^{n} w_j f_\alpha(x_j) \\ \mathbf{s.t.} \quad & \mathbf{A} \cdot \mathbf{x} \leq \mathbf{1}, \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

, where

$$f_\alpha = \begin{cases} \ln(x_j), & \text{if } \alpha = 1 \\ \frac{x_j^{1-\alpha}}{1-\alpha}, & \text{if } \alpha \neq 1 \end{cases}$$

$$\mathbf{A} \geq \mathbf{0}, \quad A_{ij} \neq 0 \Rightarrow A_{ij} \geq 1$$

- Any α-fair packing problem can be scaled to this form without affecting the approximation guarantee

- Notation:

$$A_{\max} = \max_{i,j} A_{ij}, \quad w_{\max} = \max_j w_j, \quad w_{\min} = \min_j w_j$$

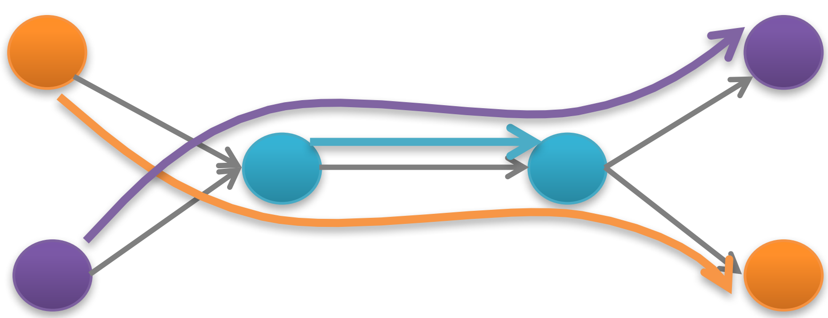
$$N \equiv nm A_{\max} \frac{w_{\max}}{w_{\min}}$$

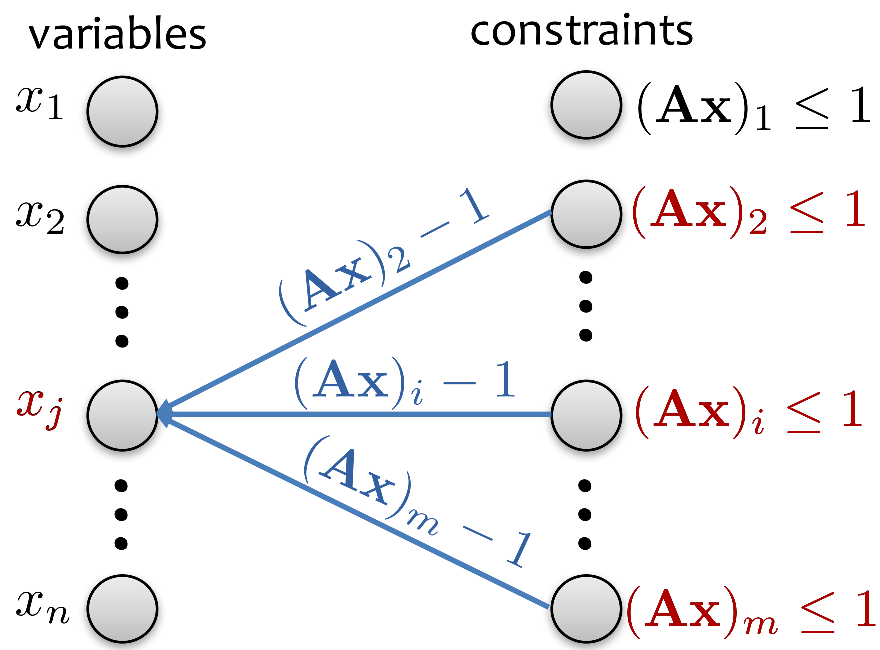
Model of Distributed Computation

- Each distributed agent j knows:
  - weight w_j
  - j^{th} column of $\mathbf{A}$
  - global problem parameters:

    $A_{\max}, w_{\max}, n, m$
- Agent j collects in each round:
  - $(\mathbf{Ax})_i - 1$, for all i with $A_{ij} \neq 0$

$$\begin{aligned} \mathbf{max} \quad & \sum_{j=1}^{n} w_j f_\alpha(x_j) \\ \mathbf{s.t.} \quad & \mathbf{A} \cdot \mathbf{x} \leq \mathbf{1}, \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$





KKT conditions

$$\max \quad \sum_{j=1}^{n} w_j f_\alpha(x_j)$$

$$\text{s.t.} \quad \sum_{j=1}^{n} A_{ij} x_j \le 1, i \in \{1, ..., m\}$$

$$x_j \ge 0, \forall j$$

y_i : dual variable (Lagrange multiplier)

1. $\mathbf{A} \cdot \mathbf{x} \le \mathbf{1}, \mathbf{x} \ge \mathbf{0}$ (primal feasibility)
2. $\mathbf{y} \ge \mathbf{0}$ (dual feasibility)
3. $y_i = y_i \sum_j A_{ij} x_j, \forall i$ (complementary slackness)
4. $\boxed{{x_j}^{\alpha} \sum_i y_i A_{ij} = w_j, \forall j}$ (gradient conditions)

$$f_\alpha(x_j) = \begin{cases} \ln(x_j), & \text{if } \alpha = 1 \\ \frac{x_j^{1-\alpha}}{1-\alpha}, & \text{if } \alpha \ne 1 \end{cases}$$

Outline

- Introduction
- Model, Scaling, and Preliminaries
- Algorithm
- Convergence
- Asymptotic behavior of α-fair allocations

Intuition

$$\boxed{{x_j}^{\alpha} \sum_i y_i A_{ij} = w_j}$$

Primal algorithm ($\alpha = 1 \rightarrow f_\alpha = \ln(x)$)
from [Kelly et al. 1998]

$$\frac{dx_j}{dt} = k(w_j - {x_j}^{\alpha} \sum_i y_i A_{ij})$$

$$y_i = F\left(\sum_j A_{ij} x_j\right)$$

$$= C \cdot \exp(\kappa(\sum_j A_{ij} x_j - 1))$$

Packing algorithm ($\alpha = 0 \rightarrow f_\alpha = x$)
from [Awerbuch and Khandekar 2008]

Initialization: $x_j \leftarrow 0$
In each round:

$$\boxed{y_i \leftarrow \exp(\kappa(\sum_j A_{ij} x_j - 1))}$$

If ${x_j}^{\alpha} \sum_i y_i A_{ij} \leq (1-\gamma) w_j$

$$x_j \leftarrow \max\{\delta, (1+\beta) x_j\}$$

If ${x_j}^{\alpha} \sum_i y_i A_{ij} \geq (1+\gamma) w_j$

$$x_j \leftarrow (1-\beta) x_j$$

$$\Phi_{PF}(x) = \sum_{j=1}^{n} w_j f_1(x_j) - \sum_{i=1}^{m} \int_{z=0}^{\sum_k A_{ik} x_k} F(z) dz$$

$$= \sum_{j=1}^{n} w_j f_1(x_j) - \frac{1}{\kappa} \sum_{i=1}^{m} y_i + \text{const.}$$

$$\Phi_{LP}(x) = \sum_{j=1}^{n} w_j f_0(x_j) - \frac{1}{\kappa} \sum_{i=1}^{m} y_i$$

$$f_\alpha(x_j) = \begin{cases} \ln(x_j), & \text{if } \alpha = 1 \\ \frac{x_j^{1-\alpha}}{1-\alpha}, & \text{if } \alpha \neq 1 \end{cases}$$

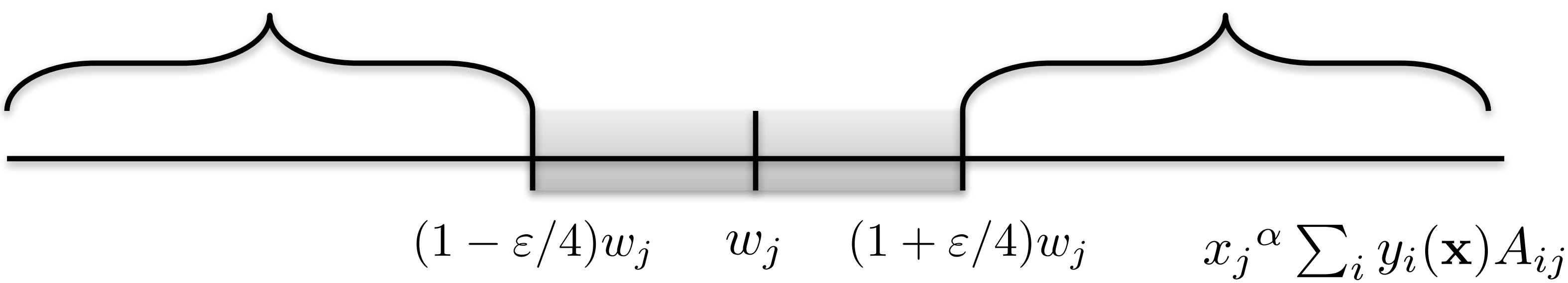
Algorithm

$$\boxed{{x_j}^{\alpha} \sum_i y_i A_{ij} = w_j}$$

$$y_i = y_i(\mathbf{x}) = C \cdot e^{\kappa\left(\sum_j A_{ij} x_j - 1\right)}$$

$$x_j \leftarrow x_j(1+\beta_1)$$

$$x_j \leftarrow \begin{cases} x_j(1-\beta_2), & \text{if } x_j(1-\beta_2) \geq \delta_j \\ \delta_j, & \text{otherwise} \end{cases}$$



Outline

- Introduction
- Model, Scaling, and Preliminaries
- Algorithm
- Convergence
- Asymptotic behavior of α-fair allocations

A High-Level Analysis Overview

KKT Conditions:

1. $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{1}, \mathbf{x} \geq \mathbf{0}$ (primal feasibility)
2. $\mathbf{y} \geq \mathbf{0}$ (dual feasibility)
3. $y_i = y_i \sum_j A_{ij} x_j, \forall i$ (complementary slackness)
4. $\underbrace{{x_j}^{\alpha} \sum_i y_i A_{ij} = w_j, \forall j}_{\text{The main part}}$ (gradient conditions)

Preliminaries

Choose:

- A bounded, non-decreasing potential function;
- A suitable definition of stationary rounds, so that:
  - In non-stationary rounds, potential increases significantly
  - In stationary rounds, the solution provides an ε- approximation

Potential Function

$$\Phi(\mathbf{x}) = \sum_j w_j f_\alpha(x_j) - \frac{1}{\kappa}\sum_i y_i(\mathbf{x})$$

What happens when algorithm performs updates?

$$\frac{\partial \Phi(\mathbf{x})}{\partial x_j} = \frac{w_j}{{x_j}^\alpha} - \sum_i y_i(\mathbf{x}) A_{ij} = \frac{1}{{x_j}^\alpha}\left(w_j - {x_j}^\alpha \sum_i y_i(\mathbf{x}) A_{ij}\right)$$

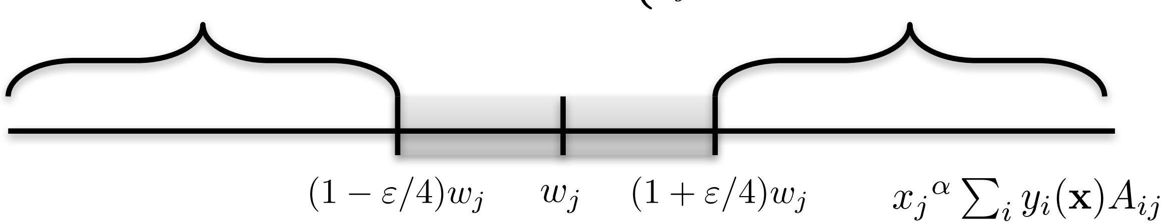
$$x_j \uparrow \quad w_j > {x_j}^\alpha \sum_i y_i(\mathbf{x}) A_{ij} \quad \Rightarrow \Phi(\mathbf{x}) \uparrow$$

$$x_j \downarrow \quad w_j < {x_j}^\alpha \sum_i y_i(\mathbf{x}) A_{ij} \quad \Rightarrow \Phi(\mathbf{x}) \uparrow$$

$$\boxed{y_i = y_i(\mathbf{x}) = C \cdot e^{\kappa(\sum_j A_{ij} x_j - 1)}}$$

$$\boxed{{x_j}^\alpha \textstyle\sum_i y_i A_{ij} = w_j}$$

$$x_j \leftarrow x_j(1+\beta_1) \qquad x_j \leftarrow \begin{cases} x_j(1-\beta_2), & \text{if } x_j(1-\beta_2) \geq \delta_j \\ \delta_j, & \text{otherwise} \end{cases}$$



The General Idea

x_j is bounded $\Phi(\mathbf{x})$ is bounded

- The algorithm makes updates as long as:

$$\exists j : {x_j}^{\alpha} \sum_i A_{ij} y_i(\mathbf{x}) \notin ((1-\varepsilon/4)w_j, (1+\varepsilon/4)w_j)$$

  - It may take a long time before the algorithm stops making updates…
- The idea is to use the notion of stationary rounds:
  - In a stationary round, bound the duality gap (use Lagrange duality)
  - In non-stationary round, show a large (multiplicative or additive) progress in the potential function

Convergence Results

	Approximation	Convergence Time	Notes
$\alpha < 1$	$(1+\varepsilon)$-multiplicative	$O\left(\frac{\ln^4(N/\varepsilon)}{\alpha^2\varepsilon^5}\right)$	$\varepsilon \leq \frac{1-\alpha}{\alpha}$
$\alpha = 1$	$^*W\varepsilon$ -additive	$O\left(\frac{\ln^4(N/\varepsilon)}{\varepsilon^5}\right)$	$\varepsilon \leq 1$
$\alpha > 1$	$(1-\varepsilon\alpha)$-multiplicative	$O\left(\frac{\ln^2(N/\varepsilon)}{\varepsilon^4}\right)$	$\varepsilon \leq \frac{9}{10} \cdot \frac{1}{\alpha}$

$^*W = \sum_j w_j$

Outline

- Introduction
- Model, Scaling, and Preliminaries
- Algorithm
- Convergence
- Asymptotic behavior of α-fair allocations

Asymptotic Cases and Behavior

Lemma 1. If $\alpha \leq O\left(\frac{\varepsilon}{\ln(N/\varepsilon)}\right)$, then α-fair packing can be ε-approximated by any ε-approximate packing LP algorithm.

Lemma 2. ε-approximate solution to α-fair packing for $\alpha = 1$ is also an ε-approximate solution to α-fair packing for $|\alpha - 1| \leq O\left(\frac{\varepsilon^2}{\ln^2(N/\varepsilon)}\right)$.

Lemma 3. The optimal solution to α-fair packing for $\alpha \geq \frac{\ln(N/\varepsilon)}{\varepsilon}$ is also an entry-wise ε-approximation of the max-min fair vector. Furthermore, in this case the max-min fair vector is an $O(\varepsilon\alpha)$-approximation to the α-fair packing.

Max-Min Fair Packing

Definition (max-min fairness). A vector $\mathbf{x} \geq \mathbf{0}$ is max-min fair if $\mathbf{Ax} \leq \mathbf{1}$ and any other vector $\mathbf{z} \geq \mathbf{0}$ such that $\mathbf{Az} \leq \mathbf{1}$ satisfies:
if $z_j > x_j$ for some j then there exists k such that $z_k < x_k \leq x_j$.

- Finding a max-min fair vector subject to packing constraints is not a convex problem, but rather a multi-objective problem
- The best (distributed) convergence time is $O(n)$, total work: $O(mn^2)$

α-Fair vs Max-Min Fair Packing

Lemma 3. The optimal solution to α-fair packing for $\alpha \geq \frac{\ln(N/\varepsilon)}{\varepsilon}$ is also an entry-wise ε-approximation of the max-min fair vector. Furthermore, in this case the max-min fair vector is an $O(\varepsilon\alpha)$-approximation to the α-fair packing.

- It was known from [Mo, Walrand '00] that when $\alpha \to \infty$, α-fair vector approaches the max-min fair vector
- Lemma 3 tells us how fast this happens
- As a side result, we also get the first convex relaxation of the max-min fair packing problem with the ε-multiplicative gap

Summary & Future Directions

- A fast, distributed, and stateless algorithm for α-fair packing problems
- Characterization of asymptotic cases of α-fair allocations
- The problem arises in many different application areas
- Future directions:
  – Improving the convergence time by relaxing the "statelessness"
  – Extension of the techniques to other (non-smooth) convex problems

Thanks!

jelena@ee.columbia.edu

www.ee.columbia.edu/~jelena

Why is Poly-Log Convergence for α-Fair Packing Surprising?

- α-fair objectives are neither $\underbrace{\text{Lipschitz continuous}}$ nor $\underbrace{\text{smooth}}$

$$\boxed{\|f(x) - f(y)\| \leq M\|x - y\|} \qquad \boxed{\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|}$$

$$\frac{df_\alpha(x_j)}{dx_j} = \frac{1}{{x_j}^\alpha} \xrightarrow{x \downarrow 0} \infty \qquad \frac{d^2 f_\alpha(x_j)}{d{x_j}^2} = -\alpha \frac{1}{{x_j}^{\alpha+1}} \xrightarrow{x \downarrow 0} -\infty$$

- α- fair objectives are strongly concave for $\mathbf{x} \leq \mathbf{1}$

  $\Rightarrow$ The dual objective is smooth

  But, the smoothness parameter is at least linear in some of the input parameters (# of variables, width)

- Nesterov's "smooth minimization of non-smooth functions"

$$\min_{\mathbf{x} \in P} \hat{f}(\mathbf{x}) + \boxed{\max_{\mathbf{y} \in Q} \{\langle \mathbf{A}\mathbf{x}, \mathbf{y} \rangle - \hat{\phi}(\mathbf{y})\}}$$

$$\min_{\mathbf{x} \geq \mathbf{0}} - \sum_j w_j f_\alpha(x_j)(\mathbf{x}) + \boxed{\max_{\mathbf{y} \geq \mathbf{0}} \{\langle \mathbf{A}\mathbf{x} - \mathbf{1}, \mathbf{y} \rangle\}}$$