Fast and Interpretable Dynamics for Fisher Markets via Block-Coordinate Updates

Tianlong Nan, Yuan Gao, Christian Kroer

Columbia University, IEOR

AAAI-23 · February 7-14, 2023 · Washington, DC, USA

In this paper. . .

- Two stochastic algorithms for computing large-scale market equilibria:
  - *(Proximal) block-coordinate descent on EG* (`BCDEG`)
  - *Block-coordinate proportional response* (`BCPR`)

In this paper...

- Two stochastic algorithms for computing large-scale market equilibria:
  - *(Proximal) block-coordinate descent on EG* (`BCDEG`)
  - *Block-coordinate proportional response* (`BCPR`)
- Strong theoretical convergence guarantees and details for efficient implementation

In this paper...

- Two stochastic algorithms for computing large-scale market equilibria:
  - *(Proximal) block-coordinate descent on EG* (BCDEG)
  - *Block-coordinate proportional response* (BCPR)

- Strong theoretical convergence guarantees and details for efficient implementation

- A tâtonnement-style interpretation for projected-gradient-type methods in market equilibrium computation (e.g. BCDEG)

In this paper. . .

- Two stochastic algorithms for computing large-scale market equilibria:
  - *(Proximal) block-coordinate descent on EG* (`BCDEG`)
  - *Block-coordinate proportional response* (`BCPR`)
- Strong theoretical convergence guarantees and details for efficient implementation
- A tâtonnement-style interpretation for projected-gradient-type methods in market equilibrium computation (e.g. `BCDEG`)
- Extensive numerical experiments demonstrating the numerical efficiency of the algorithms

Introduction

- Fisher market:
  - A set of m divisible items (each denoted as j) with unit supplies;
  - A set of n buyers (each denoted as i) with budget $B_i > 0$;
  - Valuation $(v_{ij})_{i\in[n],j\in[m]}$, price $(p_j)_{j\in[m]}$, and allocation $(x_{ij})_{i\in[n],j\in[m]}$.
- A market equilibrium (ME) is an allocation-price pair satisfying
  - Buyer optimality;
  - Market clearance.

Block Coordinate Descent Algorithm for the EG Program (BCDEG)

- Eisenberg-Gale (EG) convex program:

$$\begin{aligned} \max_x \quad & \sum\nolimits_i B_i \log u_i \\ \text{s.t.} \quad & u_i \le \langle v_i, x_i \rangle \quad \forall\, i \\ & \sum\nolimits_i x_{ij} \le 1 \quad \forall\, j \\ & x \ge 0. \end{aligned} \quad \rightarrow \quad \begin{aligned} \min_x \quad & F(x) := f(x) + \sum_j \psi_j(x_{\cdot j}) \\ & f(x) = -\sum_i B_i \log \langle v_i, x_i \rangle \\ & \psi_j(x_{\cdot j}) = \begin{cases} 0 & \text{if } x_{\cdot j} \in \Delta_n \\ +\infty & \text{otherwise} \end{cases} \quad \forall\, j. \end{aligned}$$

Block Coordinate Descent Algorithm for the EG Program (BCDEG)

- Eisenberg-Gale (EG) convex program:

$$\begin{aligned} \max_x \quad & \sum_i B_i \log u_i \\ \text{s.t.} \quad & u_i \le \langle v_i, x_i \rangle \quad \forall\, i \\ & \sum_i x_{ij} \le 1 \quad \forall\, j \\ & x \ge 0. \end{aligned} \quad \rightarrow \quad \begin{aligned} \min_x \quad & F(x) := f(x) + \sum_j \psi_j(x_{\cdot j}) \\ & f(x) = -\sum_i B_i \log \langle v_i, x_i \rangle \\ & \psi_j(x_{\cdot j}) = \begin{cases} 0 & \text{if } x_{\cdot j} \in \Delta_n \\ +\infty & \text{otherwise} \end{cases} \quad \forall\, j. \end{aligned}$$

- At each iteration, randomly choose $j \in [m]$ and only update coordinate j via

$$x_{\cdot j}^{+} = T_j(x) = \arg\min_{y_{\cdot j}} \langle \nabla_{\cdot j} f(x), y_{\cdot j} - x_{\cdot j} \rangle + \frac{1}{\eta_j} \|y_{\cdot j} - x_{\cdot j}\|^2 + \psi_j(y_{\cdot j}), \tag{1}$$

or equivalently

$$T_j(x) = \mathrm{Proj}_{\Delta_n}\Big(x_{\cdot j} - \eta_{j'} \nabla_{\cdot j} f(x) \Big).$$

where η_j is a stepsize given by a simple formula.

Block Coordinate Descent Algorithm for the EG Program (BCDEG)

- Eisenberg-Gale (EG) convex program:

$$\begin{aligned} \max_x \quad & \sum_i B_i \log u_i \\ \text{s.t.} \quad & u_i \le \langle v_i, x_i \rangle \quad \forall\, i \\ & \sum_i x_{ij} \le 1 \quad \forall\, j \\ & x \ge 0. \end{aligned} \quad \rightarrow \quad \begin{aligned} \min_x \quad & F(x) := f(x) + \sum_j \psi_j(x_{\cdot j}) \\ & f(x) = -\sum_i B_i \log \langle v_i, x_i \rangle \\ & \psi_j(x_{\cdot j}) = \begin{cases} 0 & \text{if } x_{\cdot j} \in \Delta_n \\ +\infty & \text{otherwise} \end{cases} \quad \forall\, j. \end{aligned}$$

- At each iteration, randomly choose $j \in [m]$ and only update coordinate j via

$$x_{\cdot j}^{+} = T_j(x) = \arg\min_{y_{\cdot j}} \langle \nabla_{\cdot j} f(x), y_{\cdot j} - x_{\cdot j} \rangle + \frac{1}{\eta_j} \left\| y_{\cdot j} - x_{\cdot j} \right\|^2 + \psi_j(y_{\cdot j}), \qquad (1)$$

or equivalently

$$T_j(x) = \mathrm{Proj}_{\Delta_n}\Big(x_{\cdot j} - \eta_{j'} \nabla_{\cdot j} f(x) \Big).$$

where η_j is a stepsize given by a simple formula.

- In practice, use adaptive stepsizes (economical line search).

Lemma (Coordinate-wise Lipschitz Continuity)

For any $j \in [m]$, let

$$L_j = \max_{i \in [n]} \frac{B_i v_{ij}^2}{\underline{u}_i^2}. \tag{2}$$

Then, for all $x, y \in \mathcal{X}$ such that x, y differ only in the jth block, we have

$$\tilde{f}(y) \leq \tilde{f}(x) + \langle \nabla_{\cdot j} \tilde{f}(x), y - x \rangle + \frac{L_j}{2} \|y - x\|^2. \tag{3}$$

- $\underline{u}_i = \langle v_i, (B_i / \sum_{l=1}^n B_l)\mathbf{1} \rangle$, the utility of the proportional allocation;
- $\tilde{f}$ is a "quadratic extrapolation" function of f to handle gradients on $u_i = 0$;
- "Global" Lipschitz constant for the stepsize of (full) gradient descent:

$$L = \max_{i \in [n]} \frac{B_i \|v_i\|^2}{\underline{u}_i^2},$$

the ratio of L_j / L is roughly $\max_i v_{ij}^2 / \max_i \|v_i\|^2$.

Convergence Analysis for BCDEG

Lemma (Coordinate-wise Lipschitz Continuity)

For any $j \in [m]$, let

$$L_j = \max_{i \in [n]} \frac{B_i v_{ij}^2}{\underline{u}_i^2}. \tag{4}$$

Then, for all $x, y \in \mathcal{X}$ such that x, y differ only in the jth block, we have

$$\tilde{f}(y) \le \tilde{f}(x) + \langle \nabla_{\cdot j} \tilde{f}(x), y - x \rangle + \frac{L_j}{2} \|y - x\|^2. \tag{5}$$

Theorem (Linear Convergence of BCDEG)

*Given an initial iterate x^0 and stepsizes $\eta_j^0 = 1/L_j, \forall\, j$ satisfying (**??**), let x^k be the random iterates generated by BCDEG. Then,*

$$\mathbf{E}\left[F(x^{k+1})\right] - F^* \le (1-\rho)^k \left(F(x^0) - F^*\right), \tag{6}$$

where $\rho = \min\left\{\frac{\mu}{mL_{\max}\theta^2(A,C)}, \frac{1}{m}\right\}$ and $L_{\max} = \max_j L_j$.

Economic Interpretation of Projected Gradient Steps

- Projected gradient step for buyer i and item j:

$$y = x_{\cdot j} - \eta_j g \quad \text{and} \quad x_{\cdot j}^{+} \leftarrow \mathrm{Proj}_{\Delta^n}(y). \tag{7}$$

Economic Interpretation of Projected Gradient Steps

- Projected gradient step for buyer i and item j:

$$y = x_{\cdot j} - \eta_j g \quad \text{and} \quad x_{\cdot j}^{+} \leftarrow \mathrm{Proj}_{\Delta^n}(y). \tag{7}$$

- The key step in projection computation: find the unique multiplier t such that

$$\sum_i (y_i - t)^+ = 1, \tag{8}$$

which gives the solution $x_{\cdot j}^{+} = (y - t \cdot \mathbf{1})^+$.

Economic Interpretation of Projected Gradient Steps

- Projected gradient step for buyer i and item j:

$$y = x_{\cdot j} - \eta_j g \quad \text{and} \quad x_{\cdot j}^{+} \leftarrow \mathrm{Proj}_{\Delta^n}(y). \tag{7}$$

- The key step in projection computation: find the unique multiplier t such that

$$\sum_i (y_i - t)^{+} = 1, \tag{8}$$

which gives the solution $x_{\cdot j}^{+} = (y - t \cdot \mathbf{1})^{+}$.

- Replacing t with $\eta_j p_j$ for a "price" p_j:

$$x_{ij}^{+} = D_i(p_j) = (x_{ij} - \eta_j g_i - \eta_j p_j)^{+} \quad \text{and} \quad \sum_i D_i(p_j) = 1. \tag{9}$$

Economic Interpretation of Projected Gradient Steps

- Projected gradient step for buyer i and item j:

$$y = x_{\cdot j} - \eta_j g \quad \text{and} \quad x_{\cdot j}^{+} \leftarrow \mathrm{Proj}_{\Delta^n}(y). \tag{7}$$

- The key step in projection computation: find the unique multiplier t such that

$$\sum_i (y_i - t)^{+} = 1, \tag{8}$$

which gives the solution $x_{\cdot j}^{+} = (y - t \cdot \mathbf{1})^{+}$.

- Replacing t with $\eta_j p_j$ for a "price" p_j:

$$x_{ij}^{+} = D_i(p_j) = (x_{ij} - \eta_j g_i - \eta_j p_j)^{+} \quad \text{and} \quad \sum_i D_i(p_j) = 1. \tag{9}$$

- D_i: buyer i's *linear demand function* given prior allocation x_i

Economic Interpretation of Projected Gradient Steps

- Projected gradient step for buyer i and item j:

$$y = x_{\cdot j} - \eta_j g \quad \text{and} \quad x_{\cdot j}^{+} \leftarrow \mathrm{Proj}_{\Delta^n}(y). \tag{7}$$

- The key step in projection computation: find the unique multiplier t such that

$$\sum_i (y_i - t)^{+} = 1, \tag{8}$$

which gives the solution $x_{\cdot j}^{+} = (y - t \cdot \mathbf{1})^{+}$.

- Replacing t with $\eta_j p_j$ for a "price" p_j:

$$x_{ij}^{+} = D_i(p_j) = (x_{ij} - \eta_j g_i - \eta_j p_j)^{+} \quad \text{and} \quad \sum_i D_i(p_j) = 1. \tag{9}$$

- D_i: buyer i's *linear demand function* given prior allocation x_i
- p_j: *market-clearing* price for item j

Economic Interpretation of Projected Gradient Steps

- Seller-side "dynamic pricing" view:
  - Randomly sample an item j
  - Each buyer i forms a demand function of item j:

$$D_i(p_j) \leftarrow (x_{ij} - \eta_j g - \eta_j p_j)^+$$

  - Seller sets the unique (market-clearing) price p_j such that

$$\sum_i D_i(p_j) = 1$$

  - Each buyer i chooses her new allocation of item j via her demand

$$x_{ij}^+ = D_i(p_j)$$

Economic Interpretation of Projected Gradient Steps

- Seller-side "dynamic pricing" view:
  - Randomly sample an item j
  - Each buyer i forms a demand function of item j:

$$D_i(p_j) \leftarrow (x_{ij} - \eta_j g - \eta_j p_j)^+$$

  - Seller sets the unique (market-clearing) price p_j such that

$$\sum_i D_i(p_j) = 1$$

  - Each buyer i chooses her new allocation of item j via her demand

$$x_{ij}^+ = D_i(p_j)$$

- Let $\beta_i = B_i/u_i$ be the current *utility price* for buyer i. Then:
  - $x_{ij}^+ = x_{ij}$ (stays the same) if $p_j = \beta_i v_{ij}$;
  - $x_{ij}^+ > x_{ij}$ (increases) if $p_j < \beta_i v_{ij}$;
  - $x_{ij}^+ < x_{ij}$ (decreases) if $p_j > \beta_i v_{ij}$.

Remark.

Here, buyers slowly adapt to the changing environment, while the prices are set to balance supply and demand, similar to a tâtonnement process.

Block Coordinate Proportional Response (BCPR)

- The Shmyrev convex program:

$$\begin{aligned} \max_{b} \quad & \sum_{i,j:v_{ij}>0} b_{ij} \log\left(\frac{v_{ij}}{p_j(b)}\right) \\ \text{s.t.} \quad & \sum_j b_{ij} = B_i \quad \forall\, i \\ & b \geq 0. \end{aligned} \quad \rightarrow \quad \begin{aligned} \min_{b} \quad & \Phi(b) := \varphi(b) + \sum_i r_i(b_i) \\ & \varphi(b) = -\sum_{i,j:v_{ij}>0} b_{ij} \log\left(\frac{v_{ij}}{p_j(b)}\right) \\ & r_i(b_i) = \begin{cases} 0 & \text{if } \sum_j b_{ij} = B_i,\, b_i \geq 0 \\ +\infty & \text{otherwise} \end{cases} \quad \forall\, i, \end{aligned}$$

where $p_j(b) = \sum_i b_i$ for all j.

- The Shmyrev convex program:

$$\begin{aligned} \max_b \quad & \sum_{i,j:v_{ij}>0} b_{ij} \log\left(\frac{v_{ij}}{p_j(b)}\right) \\ \text{s.t.} \quad & \sum_j b_{ij} = B_i \quad \forall\, i \\ & b \geq 0. \end{aligned} \quad \rightarrow \quad \begin{aligned} \min_b \quad & \Phi(b) := \varphi(b) + \sum_i r_i(b_i) \\ & \varphi(b) = -\sum_{i,j:v_{ij}>0} b_{ij} \log\left(\frac{v_{ij}}{p_j(b)}\right) \\ & r_i(b_i) = \begin{cases} 0 & \text{if } \sum_j b_{ij} = B_i, b_i \geq 0 \\ +\infty & \text{otherwise} \end{cases} \quad \forall\, i, \end{aligned}$$

where $p_j(b) = \sum_i b_i$ for all j.

- Applying *mirror descent* (MD) on Shmyrev gives the classical *Proportional Response* dynamics, a scalable, efficient and interpretable equilibrium computation algorithm.

- BCPR runs MD update on each block: at each iteration, randomly choose $i \in [n]$, only update coordinate i via

$$b_i^+ = T_i(b) := \arg\min_{a_i} \left\{ \langle \nabla_i \varphi(b), a_i - b_i \rangle + \frac{1}{\alpha_i} D_{\mathrm{KL}}(a_i, b_i) + r_i(a_i) \right\}. \tag{10}$$

Block Coordinate Proportional Response (BCPR)

- BCPR runs MD update on each block: at each iteration, randomly choose $i \in [n]$, only update coordinate i via

$$b_i^+ = T_i(b) := \arg\min_{a_i} \left\{ \langle \nabla_i \varphi(b), a_i - b_i \rangle + \frac{1}{\alpha_i} D_{\mathrm{KL}}(a_i, b_i) + r_i(a_i) \right\}. \tag{10}$$

- The above update has a simple formula:

$$b_{ij}^+ = \frac{1}{Z_i} b_{ij} \left(\frac{v_{ij}}{p_j} \right)^{\alpha_i}, \tag{11}$$

where Z_i is a normalization constant s.t. $\sum_j b_{ij}^+ = B_i$.

Theorem ($O(1/k)$ Convergence of BCPR)

Let b^k be random iterates generated by BCPR with $\alpha_i^k = 1/L_i$ for all k, then

$$\mathbf{E}\left[\Phi(b^k)\right] - \Phi^* \leq \frac{n}{n+k}\left(\Phi^* - \Phi(b^0) + D(b^*, b^0)\right), \tag{12}$$

where Φ^* is the optimal objective value.

Theorem ($O(1/k)$ Convergence of BCPR)

Let b^k be random iterates generated by BCPR with $\alpha_i^k = 1/L_i$ for all k, then

$$\mathbf{E}\left[\Phi(b^k)\right] - \Phi^* \leq \frac{n}{n+k}\Big(\Phi^* - \Phi(b^0) + D(b^*, b^0)\Big), \tag{12}$$

where Φ^* is the optimal objective value.

- Again, we use line search strategy to update stepsizes adaptively. However, theoretically we only have

$$\mathbf{E}\left[\Phi(b^k) - \Phi(b^*)\right] \leq \frac{n}{n+k}\left(\Phi(b^*) - \Phi(b^0) + \sum_i \frac{1}{\alpha_i^0} D(b_i^*, b^0)\right) + \frac{1}{k}\sum_{l=1}^{k} \mathbf{E}\left[\sum_i \frac{1}{\alpha_i^l} D(b_i^*, b^l) - \sum_i \frac{1}{\alpha_i^{l-1}} D(b_i^*, b^l)\right]. \tag{13}$$

This cannot guarantee convergence due to the last term.

Proportional Response with Line Search (PRLS)

- We also give a Mirror Descent with Line Search (MDLS) algorithm, leveraging recent advances in *relatively smooth* convex optimization.

Theorem ($O(1/k)$ Convergence of PRLS)

Let b^k be iterates generated by PRLS starting from any initial solution feasible b^0, then we have

$$\varphi(b^k) - \varphi^* \leq \frac{1}{\rho^-} \cdot \frac{D(b^*, b^0)}{k}, \tag{14}$$

where ρ^- is the shrinking factor of stepsize.

- We also give a Mirror Descent with Line Search (MDLS) algorithm, leveraging recent advances in *relatively smooth* convex optimization.

Theorem ($O(1/k)$ Convergence of PRLS)

Let b^k be iterates generated by PRLS starting from any initial solution feasible b^0, then we have

$$\varphi(b^k) - \varphi^* \leq \frac{1}{\rho^-} \cdot \frac{D(b^*, b^0)}{k}, \tag{14}$$

where ρ^- is the shrinking factor of stepsize.

- In the deterministic case, $D(b^*, b^k)$ decreases monotonely.

Block Coordinate Descent Algorithm for CES utility function ($0 < \rho < 1$)

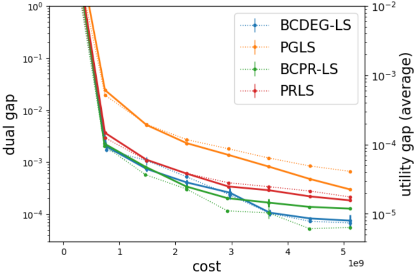
- BCDEG for CES utility function.

Block Coordinate Descent Algorithm for CES utility function ($0 < \rho < 1$)

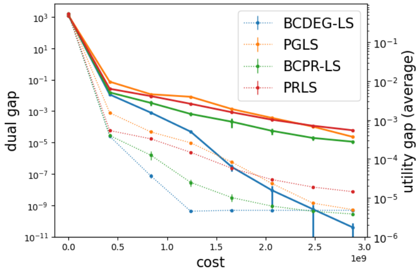
- BCDEG for CES utility function.

- BCPR for CES utility function.

Numerical Experiments



(a) Simulated data



(b) Real data

Figure: Partial experimental results on error bar plots.
(the x-axis shows units of work performed)

Thank you for listening!