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

Tianlong Nan, Yuan Gao, Christian Kroer

Columbia University, New York, USA
{tianlong.nan, gao.yuan, christian.kroer}@columbia.edu

Abstract

We consider the problem of large-scale Fisher market equilibrium computation through scalable first-order optimization methods. It is well-known that market equilibria can be captured using structured convex programs such as the Eisenberg-Gale and Shmyrev convex programs. Highly performant deterministic full-gradient first-order methods have been developed for these programs. In this paper, we develop new block-coordinate first-order methods for computing Fisher market equilibria, and show that these methods have interpretations as tâtonnement-style or proportional response-style dynamics where either buyers or items show up one at a time. We reformulate these convex programs and solve them using proximal block coordinate descent methods, a class of methods that update only a small number of coordinates of the decision variable in each iteration. Leveraging recent advances in the convergence analysis of these methods and structures of the equilibrium-capturing convex programs, we establish fast convergence rates of these methods.

Introduction

In a market equilibrium (ME) a set of items is allocated to a set of buyers via a set of prices for the items and an allocation of items to buyers such that each buyer spends their budget optimally, and all items are fully allocated. Due to its rich structural properties and strong fairness and efficiency guarantees, ME has long been used to develop fair division and online resource allocation mechanisms (Gao and Kroer 2021; Aziz and Ye 2014; Barman, Krishnamurthy, and Vaish 2018; Arnsperger 1994). Market model and corresponding equilibrium computation algorithms have been central research topics in market design and related areas in economics, computer science and operations research with practical impacts (Scarf et al. 1967; Kantorovich 1975; Othman, Sandholm, and Budish 2010; Daskalakis, Goldberg, and Papadimitriou 2009; Cole et al. 2017; Kroer et al. 2019). More specifically, many works in market design rely on the assumption that ME can be computed efficiently for large-scale market instances. For example, the well-known fair division mechanism without money *competitive equilibrium from equal incomes* (CEEI) requires computing a ME of a Fisher market under uniform buyer budgets (Varian et al. 1974). Recent work has also established close connections between market equilibria and important solution concepts in the context of large-scale Internet markets, such as *pacing equilibria* in repeated auctions (Conitzer et al. 2018, 2019; Kroer and Stier-Moses 2022). Motivated by the classical and emerging applications described above, we are interested in developing efficient equilibrium computation algorithms for large-scale market instances. In general, computing a ME is a hard problem (Chen and Teng 2009; Vazirani and Yannakakis 2011; Othman, Papadimitriou, and Rubinstein 2016). However, for the case of Fisher markets and certain classes of utility functions, efficient algorithms are known (Devanur et al. 2008; Zhang 2011; Gao and Kroer 2020), often based on solving a specific convex program—whose solutions are ME and vice versa—using an optimization algorithm. In this paper, we focus on two well-known such convex programs, namely, the Eisenberg-Gale (EG) (Eisenberg and Gale 1959; Eisenberg 1961) and Shmyrev convex programs (Shmyrev 2009; Cole et al. 2017).

Most existing equilibrium computation literature studies the case of a static market where all buyers and items are present in every time step of the equilibrium computation process. In contrast, we study a setting where only a random subset of buyer-item pairs show up at each time step. Such a setting is well-motivated from a computational perspective, since stochastic methods are typically more efficient for extremely large problems. Secondly, our model allows us to model new types of market dynamics. We make use of recent advances in *stochastic first-order optimization*, more specifically, block-coordinate-type methods, to design new equilibrium computation algorithms for this setting. The resulting equilibrium computation algorithms have strong convergence guarantees and consistently outperform deterministic full-information algorithms in numerical experiments. In addition, many of the optimization steps not only give efficient update formulas for market iterates, but also translate to interpretable market dynamics.

Summary of contribution. We propose two stochastic algorithms for computing large-scale ME: *(proximal) block-coordinate descent on EG* (`BCDEG`) and *block-coordinate proportional response* (`BCPR`). These algorithms are derived by applying stochastic block-coordinate-type algorithms on reformulated equilibrium-capturing convex programs. More

specifically, BCDEG is based on (proximal) stochastic block coordinate descent (BCD) and BCPR is based on a non-Euclidean (Bregman) version of BCD. We show that these algorithms enjoy attractive theoretical convergence guarantees, and discuss important details for efficient implementation in practice. Furthermore, we show that the Euclidean projection onto the simplex in BCDEG (and other preexisting projected-gradient-type methods) has a tâtonnement-style interpretation. We then demonstrate the practical efficiency of our algorithms via extensive numerical experiments on synthetic and real market instances, where we find that our algorithms are substantially faster than existing state-of-the-art methods such as proportional response dynamics.

Preliminaries and notation. Unless otherwise stated, we consider a *linear Fisher market* with n *buyers* and m *items*. We use $i \in [n]$ to denote a buyer and $j \in [m]$ to denote an item. Each buyer i has a *budget* $B_i > 0$ and each item has supply one. An *allocation* (or *bundle*) for buyer i is a vector $x_i \in \mathbb{R}^m_+$ specifying how much buyer i gets of each item j. Given x_i, buyer i gets utility $\langle v_i, x_i \rangle = \sum_j v_{ij} x_{ij}$, where $v_i \in \mathbb{R}^m_+$ is their valuation vector. Given *prices* $p \in \mathbb{R}^m_+$ (i.e., price of item j is p_j) on all items, buyer i pays $\langle p, x_i \rangle = \sum_j p_j x_{ij}$ for x_i. Given prices p and budget B_i, a bundle x_i is *budget feasible* for buyer i if $\langle p, x_i \rangle \leq B_i$. We also use $x_{\cdot j} \in \mathbb{R}^n_+$ to denote a vector of amounts of item j allocated to all buyers. The *demand set* of buyer i is the set of budget-feasible utility-maximizing allocations:

$$\mathcal{D}_i(p) = \arg\max_{x_i} \{u_i = \langle v_i, x_i \rangle : \langle p, x_i \rangle \leq B_i\}. \qquad \text{(D)}$$

A market equilibrium (ME) is an allocation-price pair (x^*, p^*) such that $x^*_i \in \mathcal{D}_i(p^*)$ for all i and $\sum_i x^*_{ij} \leq 1$ for all j, with equality if $p^*_j > 0$.

Related Work

Since the seminal works by Eisenberg and Gale (Eisenberg and Gale 1959; Eisenberg 1961), there has been an extensive literature on equilibrium computation for Fisher markets, often based on convex optimization characterizations of ME (Devanur et al. 2008; Zhang 2011; Birnbaum, Devanur, and Xiao 2011; Cole et al. 2017; Gao and Kroer 2020; Garg and Kapoor 2006). Equilibrium computation algorithms for more general market models with additional constraints—such as indivisible items and restrictions on the set of permissible bundles for each buyer—have also been extensively studied (Othman, Sandholm, and Budish 2010; Budish et al. 2016); these algorithms are often based on approximation algorithms, mixed-integer programming formulations, and local search heuristics.

In Gao and Kroer (2020), the authors considered three (deterministic) first-order optimization methods, namely projected gradient (PG), Frank-Wolfe (FW) and mirror descent (MD) for solving convex programs capturing ME. To the best of our knowledge there are no existing results on block-coordinate methods for ME. In the optimization literature there is an extensive and ongoing literature on new block-coordinate-type algorithms and their analysis (see, e.g., Tseng (2001); Wright (2015); Beck and Tetruashvili (2013); Hanzely and Richtárik (2019, 2021); Liu and Wright (2015); Nesterov (2012); Richtárik and Takáč (2014); Gao et al. (2020); Attouch, Bolte, and Svaiter (2013); Zhang (2020); Reddi et al. (2016)). As mentioned previously, our BCDEG algorithm is based on *proximal block-coordinate descent* (PBCD) applied to EG. Linear convergence of the mean-square error of the last iterate of PBCD for nonsmooth, finite-sum, composite optimization (with an objective function of the form $F(x) = \sum_i f_i(x) + \psi(x)$) has been established under different error bound conditions (Richtárik and Takáč 2014; Karimi, Nutini, and Schmidt 2016; Reddi et al. 2016). For BCPR, we adopt the analysis of a recently proposed *non-Euclidean* (Bregman) PBCD (Gao et al. 2020), which in turn made use of the convergence theory developed in Bauschke, Bolte, and Teboulle (2017).

Block Coordinate Descent Algorithm for the EG Program

(Proximal) Block coordinate descent methods (BCD) are often used to solve problems whose objective function consists of a smooth part and a (potentially) nonsmooth part. The second part is typically block-coordinate-wise separable. BCD algorithms update only a small block of coordinates at each iteration, which makes each iteration much cheaper than for deterministic methods, and this enables scaling to very large instances. The EG convex program, which captures market equilibria, can be written in a form amenable to proximal block coordinate descent method. The EG convex program for buyers with linear utility functions is

$$\begin{aligned} \max_{x} \quad & \sum_i B_i \log u_i \\ \text{s.t.} \quad & u_i \leq \langle v_i, x_i \rangle \quad \forall\, i \\ & \sum_i x_{ij} \leq 1 \quad \forall\, j \\ & x \geq 0. \end{aligned} \qquad \text{(EG)}$$

Any optimal solution x^* to (EG) and the (unique) optimal Lagrange multipliers $p^* \in \mathbb{R}^m_+$ associated with the constraints $\sum_i x_{ij} \leq 1$, $j \in [m]$ forms a market equilibrium. In fact, this holds more generally if the utility functions are concave, continuous, nonnegative, and homogeneous with degree 1.

To apply BCD, note that (EG) is of the following form:

$$\min_{x \in \mathbb{R}^{m \times n}} F(x) := f(x) + \psi(x) = f(x) + \sum\nolimits_j \psi_j(x_{\cdot j}) \quad (1)$$

where $f(x) = -\sum_i B_i \log \langle v_i, x_i \rangle$ and $\psi_j(x_{\cdot j}) = 0$ if $\sum_i x_{ij} \leq 1, x_{\cdot j} \geq 0$ and $+\infty$ otherwise.

Thus, the nonsmooth term $\psi(x)$ decomposes along the items j, and we can therefore treat x as a set of blocks: each item j has a block of allocation variables corresponding to how much of item j is given to each buyer. We use $x_{\cdot j} = (x_{1j}, \ldots, x_{nj})$ to denote the j'th block. Given the full gradient of f at x as $\nabla f(x)$, we will also need the partial gradient w.r.t. the jth block $x_{\cdot j}$, which we denote as $\nabla_{\cdot j} f(x)$, that is, $\nabla_{\cdot j} f(x) = (\nabla_{1j} f(x), \ldots, \nabla_{nj} f(x))$.

In each iteration t of the BCD method, we first choose an index $j' \in [m]$ at random, with a corresponding stepsize $\eta_{j'}$.

Then, the next iterate x^+ is generated from x via $x^+_{\cdot j} = T_j(x)$ if $j = j'$ and $x_{\cdot j}$ otherwise, where $T_j(x)$ equals

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

The above proximal mapping is equivalent to

$$T_j(x) = \text{Proj}_{\Delta_n}\Big(x_{\cdot j} - \eta_{j'} \nabla_{\cdot j} f(x)\Big), \quad (3)$$

so we can generate $T_j(x) \in \mathbb{R}^n$ via a single projection onto the n-dimensional simplex. Since the projection is the most computationally expensive part, our method ends up being cheaper than full projected gradient descent by a factor of m.

To make sure $\nabla_{\cdot j} f(x)$ exists, we need to bound buyer utilities away from zero at every iteration. To that end, let $\underline{u}_i = \langle v_i, (B_i / \sum_{l=1}^n B_l)\vec{1} \rangle$ be the utility of the proportional allocation. Then we perform "quadratic extrapolation" where we replace the objective function $f(x)$ with the function $\tilde{f}(x) = \sum_i \tilde{f}_i(x_i) = \sum_i \tilde{g}_i(u_i)$ where $\tilde{g}_i(u_i) = -B_i \log u_i$ if $u_i \geq \underline{u}_i$ and $\log \underline{u}_i - \frac{B_i}{\underline{u}_i}(u_i - \underline{u}_i) + \frac{B_i}{2\underline{u}_i^2}(u_i - \underline{u}_i)^2$ otherwise. For (EG), replacing f with $\tilde{f}$ does not affect any optimal solution (Gao and Kroer 2020, Lemma 1).

We will also need the following Lipschitz bound which ensures that the iterates are descent steps, meaning that the expected objective value is non-increasing as long as the stepsizes are not too large. The upper bound on the allowed stepsizes (that ensure descent iterates) for a specific block is governed by the Lipschitz constant w.r.t. that block of coordinates. More details and proofs can be found in Appendix A.

Lemma 1. *For any $j \in [m]$, let*

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

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

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

Let L be the "global" Lipschitz constant which determines the stepsize of (full) gradient descent. Let $L_{\max} = \max_j L_j$. Generally, it is easy to see that $1 \leq L/L_{\max} \leq n$, but we can get a stronger bound than this using the gradient $(\nabla f)_{ij} = B_i v_{ij}/u_i$. When only the variables $x_{\cdot j}$ in block j change, we have that L_j is bounded by the maximal diagonal value of the j'th-block (sub) Hessian matrix. In contrast, for L it depends on the maximal trace of the i'th-block (sub) Hessian matrix over all buyers i. From a market perspective this can be interpreted as follows: when we only adjust one block, each buyer's utility fluctuates based on that one item. On the contrary, for full gradient methods, every item contributes to the change of u_i. This yields that the ratio of L_j/L is roughly $\max_i v_{ij}^2 / \max_i \|v_i\|^2$.

Algorithm 1 states the (proximal) BCD algorithm for the EG convex program. Each iteration only requires a single projection onto an n-dimensional simplex, as opposed to m projections for the full projected gradient method. Moreover, we show in Appendix A that for linear utilities we can further reduce the computational cost per iteration of Algorithm 1 when the valuation matrix v_{ij} is sparse.

Algorithm 1: `BCDEG` (Proximal) Block Coordinate Descent for the EG Program

Input: Initial x^0, stepsizes $\eta_1^k, \eta_2^k, \ldots, \eta_m^k, \ \forall\, k \in \mathbb{N}$
for $k \leftarrow 1, 2, \ldots$ **do**
  pick $j_k \in [m]$ with probability $1/m$;
  $g^{k-1} \leftarrow \nabla_{\cdot j_k} \tilde{f}(x^{k-1})$;
  $x^k_{\cdot j_k} \leftarrow \text{Prox}_{\Delta_n}\left(x^{k-1}_{\cdot j_k} - \eta^k_{j_k} g^{k-1}\right)$;
  $x^k_{\cdot j} \leftarrow x^{k-1}_{\cdot j}, \ \forall\, j \neq j_k$;

Line search strategy. As mentioned in previous work such as Richtárik and Takáč (2014), line search is often very helpful for BCD. If it can be performed cheaply, then it can greatly speed up numerical convergence. We show later that this occurs for our setting.

We incorporate line search in Algorithm 1 with a (coordinate-wise) l_2 smoothness condition. The line search modifies Algorithm 1 as follows: after computing $x^k_{\cdot j_k}$, we check whether

$$\eta_{j_k} \|\nabla_{\cdot j_k} \tilde{f}(x^k) - \nabla_{\cdot j_k} \tilde{f}(x^{k-1})\| \leq \|x^k_{\cdot j_k} - x^{k-1}_{\cdot j_k}\|.$$

If this check succeeds then we increase the stepsize by a small multiplicative factor and go to the next iteration. If it fails then we decrease by a small multiplicative factor and redo the calculation of $x^k_{\cdot j_k}$. This line search algorithm can be implemented in $O(n)$ cost per iteration, whereas a full gradient method requires $O(nm)$ time. A full specification of `BCDEG` with line search (`BCDEG-LS`) is given in Appendix A.

Convergence analysis. Next we establish the linear convergence of Algorithm 1 under reasonably-large (fixed) stepsizes, as well as for the line search variant.

Following prior literature on the linear convergence of first-order methods for structured convex optimization problems under "relaxed strong convexity" conditions, we show that `BCDEG` generates iterates that have linear convergence of the expected objective value. For more details on these relaxed sufficient conditions that ensure linear convergence of first-order methods, see Karimi, Nutini, and Schmidt (2016) and references therein.

Gao and Kroer (2020) showed that (EG) and other equilibrium-capturing convex programs can be reformulated to satisfy these conditions. Hence, running first-order methods on these convex programs yields linearly-convergent equilibrium computation algorithms. Similar to the proof of Gao and Kroer (2020, Theorem 2) and Karimi, Nutini, and Schmidt (2016, (39)), we first establish a *Proximal-PŁ* inequality.

Lemma 2. *For any feasible x and any $L > 0$, define $\mathcal{D}_\psi(x, L)$ to be*

$$-2L \min_y \langle \nabla \tilde{f}(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 + \psi(y) - \psi(x)$$

then we have the inequality

$$\frac{1}{2}\mathcal{D}_\psi(x, L) \geq \min\left\{\frac{\mu}{\theta^2(A, C)}, L\right\}(F(x) - F^*) \quad (6)$$

where $\theta(A, C)$ is the Hoffman constant of the polyhedral set of optimal solutions of (EG)*, which is characterized by matrices A and C where C is a matrix capturing optimality conditions, and F^* is the optimal value.*

In previous inequalities, they essentially showed that for any $L \geq \lambda = \mu/\theta^2(A, C) > 0$, $\frac{1}{2}\mathcal{D}_\psi(x, L) \geq \lambda(F(x) - F^*)$. However, here we generate a Proximal-PŁ inequality giving a lower bound on $\frac{1}{2}\mathcal{D}_\psi(x, L)$ for any $L > 0$.

Then, combining Lemma 2 with Richtárik and Takáč (2014, Lemmas 2 & 3), we can establish the main convergence theorem for BCDEG.

Theorem 1. *Given an initial iterate x^0 and stepsizes $\eta_j^0 = 1/L_j$, $\forall\ j$ satisfying* (4)*, let x^k be the random iterates generated by Algorithm 1. Then,*

$$\mathbf{E}\big[F(x^{k+1})\big] - F^* \leq (1-\rho)^k \left(F(x^0) - F^*\right), \quad (7)$$

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

Karimi, Nutini, and Schmidt (2016) also develop a block-coordinate method that applies to EG. Unlike their result, our result admits larger stepsizes that can vary per block, as well as a line search strategy, which is helpful for practical performance as we show in numerical experiment section.

Economic Interpretation of Projected Gradient Steps

As Goktas, Viqueira, and Greenwald (2021) argue, one drawback of computing market equilibrium via projected gradient methods such as BCDEG is that these methods do not give a natural interpretation as market dynamics. To address this deficiency, in this section we show that Algorithm 1, and projected gradient descent more generally, can be interpreted as distributed pricing dynamics that balance supply and demand.

The projection step in Algorithm 1, for an individual buyer i and a chosen item j, is as follows (where we use j for j_k and drop the time index for brevity):

$$x_{\cdot j}^k \leftarrow \text{Proj}_{\Delta^n}\left(x_{\cdot j}^{k-1} - \eta_j g^{k-1}\right). \quad (8)$$

As is well-known, the projection of a vector $y \in \mathbb{R}^n$ onto the simplex $\Delta^n = \left\{x \in \mathbb{R}_+^n : \sum_i x_i = 1\right\}$ can be found using an $O(n \log n)$ algorithm (the earliest discovery that we know of is Held, Wolfe, and Crowder (1974); see Appendix B for a discussion of more recent work on simplex projection). The key step is to find the (unique) number t such that

$$\sum\nolimits_i (y_i - t)^+ = 1 \quad (9)$$

and compute the solution as $x = (y - t \cdot \mathbf{1})^+$ (componentwise). This can be done with a simple one-pass algorithm if the y_i are sorted. In fact, the number t corresponds to the (unique) optimal Lagrange multiplier of the constraint $\sum_i x_i = 1$ in the KKT conditions.

In the projection step in Algorithm 1, (9) has the form

$$\sum\nolimits_i (x_{ij}^{k-1} - \eta_j g_i^{k-1} - t)^+ = 1. \quad (10)$$

Recall that we have $g^{k-1} < 0$. Note that the left-hand side of (10) is non-increasing in t and strictly decreasing around the solution (since some terms on the left must be positive for the sum to be 1). Furthermore, setting $t = 0$ gives a lower bound of $\sum_i (x_{ij}^{k-1} - \eta_j g_i^{k-1} - t)^+ > \sum_i x_{ij}^{k-1} = 1$. Hence, the unique solution t^* must be positive. Now we rewrite t as $\eta_j p_j$ for some "price" p_j. Then, (10) can be written as

$$\sum\nolimits_i D_i^k(p_j) = 1,\ D_i^k(p_j) = (x_{ij}^{k-1} - \eta_j g_i^{k-1} - \eta_j p_j)^+. \quad (11)$$

In other words, the projection step is equivalent to finding p_j^k that solves (11). Here, D_i^k can be viewed as the *linear demand function* of buyer i at time k given a prior allocation x_{ij}^{k-1}. The solution p_j^k can be seen as a *market-clearing price*, since $\sum_i D_i^k(p_j^k)$ exactly equals the unit supply of item j. After setting the price, the updated allocations can be computed easily just as in the simplex projection algorithm, that is, $x_{ij}^k = \left(x_{ij}^{k-1} - \eta_j g^{k-1} - \eta_j p_j^k\right)^+$ for all i. Equivalently, the new allocations are given by the current linear demand function: $x_{ij}^k = D_i^k(p_j^k)$.

Summarizing the above, we can recast Algorithm 1 into the following dynamic pricing steps. At each time k, the following events occur.

- An item j is sampled uniformly at random.
- For each buyer i, her demand function becomes $D_i^k(p_j) \leftarrow (x_{ij}^{k-1} - \eta_j g^{k-1} - \eta_j p_j)^+$.
- Find the unique price p_j^k such that $\sum_i D_i^k(p_j^k) = 1$.
- Each buyer chooses their new allocation of item j via $x_{ij}^k = D_i^k(p_j^k)$.

To get some intuition for the linear demand function, note that when $u_i \geq \underline{u}_i$, we have $g_i^{k-1} = -B_i v_{ij}/u_i^{k-1}$, and therefore it holds that $D_i^k\left(B_i v_{ij}/u_i^{k-1}\right) = x_{ij}^{k-1}$. In other words, the current demand of buyer i is exactly the previous-round allocation x_{ij}^{k-1} if the price of item j is $(B_i/u_i^{k-1})v_{ij}$. This can be interpreted in terms familiar from the solution of EG: let $\beta_i^{k-1} = B_i/u_i^{k-1}$ be the *utility price* at time $k-1$ for buyer i, then we get that after seeing prices p_j^k, buyer i increases their allocation of goods that beat their current utility price, and decreases their allocation on goods that are worse than their current utility price. The stepsize η_j denotes buyer i's responsiveness to price changes.

The fact that the prices are set in a way that equates the supply and demand is reminiscent of tâtonnement-style price setting. The difference here is that the buyers are the ones who slowly adapt to the changing environment, while the prices are set in order to achieve exact market clearing under the current buyer demand functions.

Relative Block Coordinate Descent Algorithm for PR Dynamics

Birnbaum, Devanur, and Xiao (2011) showed that the *Proportional Response dynamics* (PR) for linear buyer utilities can be derived by applying mirror descent (MD) with the KL divergence on the Shmyrev convex program formulated

in buyers' *bids* (Shmyrev 2009; Cole et al. 2017). The authors derived an $O(1/k)$ last-iterate convergence guarantee of PR (MD) by exploiting a "relative smoothness" condition of the Shmyrev convex program. This has later been generalized and led to faster MD-type algorithms for more general relatively smooth problems (Hanzely and Richtárik 2021; Lu, Freund, and Nesterov 2018; Gao et al. 2020). In this section, we propose a randomized extension of PR dynamics, which we call *block coordinate proportional response* (BCPR). BCPR is based on a recent stochastic mirror descent algorithm (Gao et al. 2020). We provide stepsize rules and show that each iteration involves only a single buyer and can be performed in $O(m)$ time.

Let $b \in \mathbb{R}^{n\times m}$ denote the matrix of all buyers' bids on all items and $p_j(b) := \sum_i b_{ij}$ denote the price of item j given bids b. Denote $a_{ij} = \log v_{ij}$ if $v_{ij} > 0$ and 0 otherwise. The Shmyrev convex program is

$$\begin{aligned} \max_{b} \quad & \sum\nolimits_{i,j} a_{ij} b_{ij} - \sum\nolimits_j p_j(b) \log p_j(b) \\ \text{s.t.} \quad & \sum\nolimits_j b_{ij} = B_i \quad \forall\, i \\ & b \geq 0. \end{aligned} \tag{S}$$

It is known that an optimal solution (b^*, p^*) of (S) gives equilibrium prices p_j^*. Corresponding equilibrium allocations can be constructed via $x_{ij}^* = b_{ij}^*/p_j^*$ for all i, j. (S) can be rewritten as minimization of a smooth finite-sum convex function with a potentially nonsmooth convex separable regularizer:

$$\min_{b\in\mathbb{R}^{m\times n}} \Phi(b) := \varphi(b) + r(b) = \varphi(b) + \sum\nolimits_i r_i(b_i) \tag{12}$$

where $\varphi(b) = -\sum_{i,j} b_{ij} \log\left(\frac{v_{ij}}{p_j(b)}\right)$ and $r_i(b_i) = 0$ if $\sum_j b_{ij} = B_i$, $b_i \geq 0$ and $+\infty$ otherwise.

Now we introduce the relative randomized block coordinate descent (RBCD) method for (12). We use the KL divergence as the Bregman distance in the proximal update of b. Let $D_{\mathrm{KL}}(q^1, q^2) = \sum_i q_i^1 \log(q_i^1/q_i^2)$ denote the KL divergence between q^1 and q^2 (assuming $\sum_i q_i^1 = \sum_i q_i^2$ and $q_i^1, q_i^2 > 0, \forall\, i$). In each iteration, given a current b, we select $i \in [n]$ uniformly at random and only the update i-th block of coordinates b_i. The next iterate b^+ is $b_i^+ = T_i(b_i)$ for i and $b^+ = b$ for the remaining blocks. Here, $T_i(b_i)$ equals

$$\arg\min_a \langle \nabla_i \varphi(b), a - b_i\rangle + \frac{1}{\alpha_i} D_{\mathrm{KL}}(a, b_i) + r_i(a) \tag{13}$$

where $\alpha_i > 0$ is the stepsize. It is well-known that (13) is equivalent to the following simple, explicit update formula:

$$b_{ij}^+ = \frac{1}{Z_i} b_{ij} \left(\frac{v_{ij}}{p_j}\right)^{\alpha_i} \qquad \forall j = 1, 2, \ldots, m \tag{14}$$

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

Algorithm 2 states the full block coordinate proportional response dynamics.

Convergence Analysis. The objective function of (S) is relatively smooth with $L = 1$ (Birnbaum, Devanur, and Xiao 2011, Lemma 7). This means that $\alpha_i = 1$ is a safe lower bound for stepsizes. For Algorithm 2, a last-iterate sublinear convergence rate is given by Gao et al. (2020) for $0 < \alpha_i < \frac{1+\theta_i}{L_i}$, where θ_i is the Bregman symmetry measure introduced by Bauschke, Bolte, and Teboulle (2017). For the KL divergence $\theta_i = 0$. Their proof still goes through for $\alpha_i = 1/L_i$, which yields the following

Algorithm 2: Block Coordinate Proportional Response (BCPR)

Input: Initial b^0, p^0, stepsizes $\alpha_1^k, \ldots, \alpha_n^k,\ \forall\, k \in \mathbb{N}$
for $k \leftarrow 1, 2, \ldots$ **do**
  pick $i_k \in [n]$ with probability $1/n$;
  compute $b_{i_k}^+$ based on (14) with stepsize $\alpha_{i_k}^k$;
  $b_{i_k}^k \leftarrow b_{i_k}^+$ and $b_{i'}^k \leftarrow b_{i'}^{k-1},\ \forall\, i' \neq i_k$;
  $p_j^k \leftarrow \sum_i b_{ij}^k,\ \forall\, j$;

Theorem 2. *Let b^k be random iterates generated by Algorithm 2 with $\alpha_i^k = 1/L_i$ for all k, then*

$$\mathbf{E}\big[\Phi(b^k)\big] - \Phi^* \leq \frac{n}{n+k}\Big(\Phi^* - \Phi(b^0) + D(b^*, b^0)\Big) \tag{15}$$

where Φ^* is the optimal objective value.

Line search. To speed up BCPR, we introduce a line search strategy and an adaptive stepsize strategy. BCPR with line search can be implemented by comparing $D_{\mathrm{KL}}(p^+, p)$ and $D_{\mathrm{KL}}(b_i^+, b_i)$, which takes $O(m)$ time, and is much cheaper than computing the whole objective function value of (S). Beyond that, by storing p, we also avoid touching all variables in each iteration. Therefore, the amount of data accessed and computation needed is $O(m)$ per iteration (vs. $O(nm)$ for full gradient methods). In the adaptive strategy we compute larger stepsize based on Lipschitz estimates using a closed-form formula. The BCPR with Line Search (BCPR-LS) and Adaptive BCPR (A-BCPR) are formally stated in Appendix C.

As with BCDEG, our experiments demonstrate that larger stepsizes can accelerate Algorithm 2. When we consider a series of stepsizes $\{\alpha_i^k\}_{k\in\mathbb{N}}$ generated by line search or an adaptive strategy, we can show the inequality

$$\begin{aligned} &\mathbf{E}\big[\Phi(b^k) - \Phi(b^*)\big] \\ &\leq \frac{n}{n+k}\left(\Phi(b^*) - \Phi(b^0) + \sum_i \frac{1}{\alpha_i^0} D(b_i^*, b^0)\right) \\ &+ \frac{1}{k}\sum_{l=1}^{k} \mathbf{E}\left[\sum_i \frac{1}{\alpha_i^l} D(b_i^*, b^l) - \sum_i \frac{1}{\alpha_i^{l-1}} D(b_i^*, b^l)\right]. \end{aligned} \tag{16}$$

However, we cannot guarantee convergence, as we are unable to ensure the convergence of the last term above.

Proportional Response with Line Search

In this section we extend vanilla PR dynamics to PR dynamics with line search (PRLS), by developing a Mirror Descent with Line Search (MDLS) algorithm. Intuitively, the LS strategy is based on repeatedly incrementing the stepsize and

checking the relative-smoothness condition, with decrements made when the condition fails. This is similar to the projected gradient method with line search (PGLS) in Gao and Kroer (2020, A.5), but replaces the ℓ_2 norm with the Bregman divergence. This allows larger stepsizes while guaranteeing the same sublinear last-iterate convergence. The general MDLS algorithm is stated in Appendix D.

Convergence rate. Birnbaum, Devanur, and Xiao (2011, Theorem 3) showed that a constant stepsize of $\alpha^k = 1/L$ ensures sublinear convergence at a $1/k$ rate. One of the key steps in establishing the rate is the descent lemma, which also holds in the line search case:

Lemma 3. *Let b^k be MDLS iterates. Then, $D(b^*, b^{k+1}) \leq D(b^*, b^k)$ for all k.*

We have the following theorem.

Theorem 3. *Let b^k be iterates generated by PRLS starting from any initial solution feasible b^0, then we have*

$$\varphi(b^k) - \varphi^* \leq \frac{1}{\rho^-} \cdot \frac{D(b^*, b^0)}{k} \tag{17}$$

where ρ^- is the shrinking factor of stepsize.

Unlike the result in BCPR-LS, the deterministic algorithm maintains its convergence guarantee with line search. The main issue in the block coordinate case is the lack of monotonicity of $\mathbf{E}[D_i(b^*, b^k)]$, which is avoided by the deterministic algorithm. In the proof, we also give a tighter, path-dependent bound.

Block Coordinate Descent Algorithm for CES Utility Function

In this section, we show that block coordinate descent algorithms also work for the case where buyers have *constant elasticity of substitution* (CES) utility functions (for $\rho \in (0, 1)$).

Formally, a CES utility function for buyer i with parameters $v_i \in \mathbb{R}^m_+$ and $\rho \in (0, 1)$ is $u_i(x_i) = (\sum_j v_{ij} x_{ij}^\rho)^{\frac{1}{\rho}}$ where v_{ij} denotes the valuation per unit of item j for buyer i. CES utility functions are convex, continuous, nonnegative and homogeneous and hence the resulting ME can still be captured by the EG convex program.

BCDEG for CES utility function. The resulting EG program is of similar form as (1) with $f(x) = -\sum_i \frac{B_i}{\rho} \log \langle v_i, x_i^\rho \rangle$, where $x_i^\rho = (x_{i1}^\rho, \ldots, x_{im}^\rho)$ and the same separable $\psi(x)$ as (1). Hence, as for linear Fisher markets, we can apply block-coordinate descent.

Since u and x may reach 0 at some iterates, we need smooth extrapolation techniques to ensure the existence of gradients. First, similar to Zhang (2011, Lemma 8), we lower bound x^* for all $i, j : v_{ij} > 0$, which ensures that extrapolation will not affect the equilibrium when $\min_{i,j} v_{ij} > 0$. Our bounds are tighter than Zhang (2011).

Lemma 4. *For a market with CES utility functions with $\rho \in (0, 1)$ and market equilibrium allocation x^*, for any i, j such that $v_{ij} > 0$, we have $x^*_{ij} \geq \underline{x}^*_{ij} := \omega_1(i)^{\frac{1}{1-\rho}} \omega_2(i)^{\frac{1+\rho}{1-\rho}}$, where $\omega_1(i) = \frac{B_i}{m \sum_l B_l}$, $\omega_2(i) = \min_{j:v_{ij}>0} v_{ij} / \max_j v_{ij}$.*

In Appendix E, we show how to use this bound to perform safe extrapolation. This yields the following theorem for applying BCDEG to CES utilities. Due to its similarity to the linear case, we give the full algorithm in the appendix. The theorem is a direct consequence of Lemma 4 and Richtárik and Takáč (2014, Theorem 7).

Theorem 4. *Let x^k be the random iterates generated by BCDEG for CES utility function ($\rho \in (0, 1)$) with stepsizes $\eta_j^k = 1/L_j \; \forall \; k$, where*

$$L_j = \max_{i \in [n]} \frac{B_i v_{ij} \underline{x}_{ij}^{\rho-2}}{\underline{u}_i(\rho)} \quad \text{for all } j \in [m] \tag{18}$$

and $\underline{u}_i(\rho) = \sum_j v_{ij} \underline{x}^{*\rho}_{ij}$ (defined in Lemma 4). Then,

$$\mathbf{E}\big[F(x^k)\big] - F^* \leq \Big(1 - \frac{\mu(L)}{m}\Big)^k \big(F(x^0) - F^*\big) \tag{19}$$

where $\mu(L)$ is the strong-convexity modulus w.r.t. the weighted norm $\sum_j L_j \|\cdot\|_j^2$.

BCPR for CES utility function. Unlike for linear utilities, the EG program for CES utility cannot be converted to a simple dual problem. Hence, we cannot view PR for $\rho \in (0, 1)$ as a mirror-descent algorithm and analyze it with typical relative smoothness techniques. However, Zhang (2011) nonetheless showed convergence of PR for $\rho \in (0, 1)$. We show that we can still extend their proof to show convergence of block coordinate PR for CES utility.

Theorem 5. *Let b^k be the random iterates generated by BCPR for CES utility function ($\rho \in (0, 1)$). For any $\epsilon > 0$, when*

$$k \geq \frac{2 \log \frac{\sqrt{8 D(b^*, b^0)} W^{\frac{1}{1-\rho}}}{\epsilon}}{\log \frac{n}{n-1+\rho}}, W = \frac{n}{\min_{v_{ij}>0} v_{ij} \cdot \min_i B_i}, \tag{20}$$

we have $\mathbf{E}\left[\frac{|b_{ij} - b^*_{ij}|}{b^*_{ij}}\right] \leq \epsilon$ for all i, j such that $v_{ij} > 0$.

Numerical Experiments

We performed numerical experiments based on both simulated (for linear and CES ($\rho \in (0, 1)$) utilities) and real data to test the scalability of our algorithms.

To measure the amount of work performed, we measure the number of accesses to cells in the valuation matrix. For the deterministic algorithms, each iteration costs $n \times m$, whereas for our BCDEG algorithms each iteration costs n, and for our BCPR algorithms each iteration costs m. For algorithms that employ line search, we count the valuation accesses required in order to perform the line search as well.

To measure the accuracy of a solution, we use the duality gap and average relative difference between u and u^*. The instances are small enough that we can compute the equilibrium utilities u^* using Mosek (2010).

Simulated low-rank instances. To simulate market instances, we generate a set of valuations that mimic *approximately low rank valuations*, which are prevalent in real markets, and were previously studied in the market equilibrium context by Kroer et al. (2019). The valuation for item j and

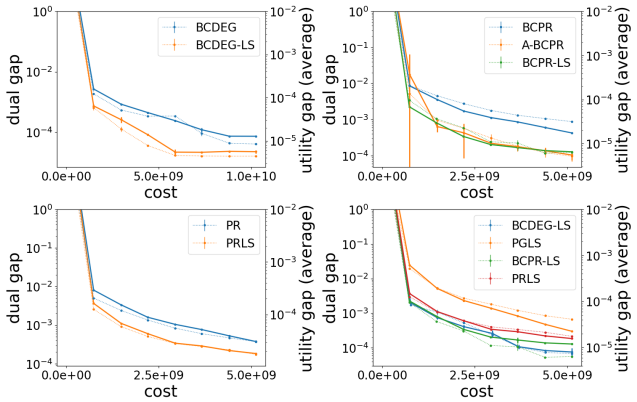

Figure 1: Performance on simulated low-rank instances. Random algorithms were implemented with seeds $0-9$. We also plotted vertical bars representing standard deviations across different seeds. The left y-axis shows performance in terms of the duality gap (solid lines for each algorithm) while the right y-axis shows performance in terms of utilities (dotted line for each algorithm). The x-axis shows units of work performed.

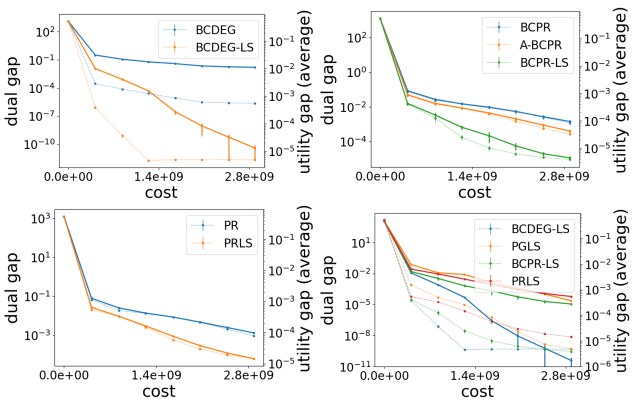

Figure 2: Performance on movierating instances. The plot setup is the same as in Fig. 1.

buyer i is generated as: $v_{ij} = v_i v_j + \epsilon_{ij}$, where $v_i \sim \mathcal{N}(1,1)$, $v_j \sim \mathcal{N}(1,1)$, and $\epsilon_{ij} \sim \textit{uniform}(0,1)$. Here, buyer i's valuation for item j consists of three parts: a value of item j itself (v_j), buyers i's average valuation (v_i), and a random term ϵ_{ij}. We consider markets with $n = m = 400$. All budgets and supplies are equal to one.

Movierating instances. We generate a market instance using a movie rating dataset collected from twitter called *Movietweetings* (Dooms, De Pessemier, and Martens 2013). Here, users are viewed as the buyers, movies as items, and ratings as valuations. Each buyer is assigned a unit budget and each item unit supply. We use the "snapshots 200K" data set and remove users and movies with too few entries. Using the matrix completion software *fancyimpute* (Rubinsteyn and Feldman 2016), we estimate missing valuations. The resulting instance has $n = 691$ buyers and $m = 632$ items.

First we compare each of our new algorithms in terms of the different stepsize strategies: `BCDEG` vs. `BCDEG-LS`, `BCPR` vs. `A-BCPR` vs. `BCPR-LS`, and `PR` vs `PRLS`. The results are shown in Fig. 1 and Fig. 2 in the upper left, upper right, and lower left corners. In all cases we see that our new line search variants perform the best.

Second, we then compare our block-coordinate algorithms to the best deterministic state-of-the-art market equilibrium algorithms: `PGLS` (Gao and Kroer 2020) and `PRLS`. The results are shown in Fig. 1 and Fig. 2 in the lower right corner. For all markets either `BCDEG` or `BCPR-LS` is best on all metrics, followed by `PRLS`, and `PGLS` in order. In general, we see that the stochastic block-coordinate algorithms converge faster than their deterministic counterparts across the board, even after thousands of iterations of the deterministic methods. Thus, block-coordinate methods seem to be better even at high precision, while simultaneously achieving better early performance due to the many more updates performed per unit of work.

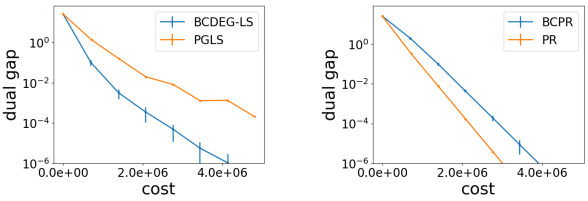

Figure 3: Performance on simulated instances with CES utility with $\rho \in (0,1)$. The setup is the same as in Fig. 1.

CES utilities. Similar to linear utilities, we generate a $n = m = 200$ scale market instance with the CES-utility parameters generated as $v_{ij} = v_i v_j + \epsilon_{ij}$, where $v_i \sim \mathcal{N}(1, 0.2^2)$, $v_j \sim \mathcal{N}(1, 0.2^2)$, and $\epsilon_{ij} \sim \textit{uniform}(0, 0.2)$. For the CES instances we were not able to obtain high-accuracy solutions from existing conic solvers, and thus we only measure performance in terms of the dual gap. Somewhat surprisingly, we find that for CES utilities vanilla `PR` converges very fast; faster than `BCPR` and all versions of `BCDEG`. Moreover, `BCDEG` suffers from extremely small stepsizes, so we have to use `BCDEG-LS`. Here, we used specific small parameters for the distributions in the simulated utilities. More discussion and experiments on CES utilities are given in Appendix E.

Conclusion and Future Work

We proposed two stochastic block-coordinate algorithms for computing large-scale ME: (proximal) block-coordinate descent on EG (`BCDEG`) and block-coordinate proportional response (`BCPR`). For each algorithm we provided theoretical convergence guarantees and showed numerically that they outperform existing state-of-the-art algorithms. We also provided a new economic interpretation of the projected gradient update used in `BCDEG`. For future work, we are interested in deriving a sublinear convergence rate for `BCPR` with adaptive stepsizes, extending it to leverage distributed and parallel computing capabilities, and allowing more general dynamic settings and other buyer utility models.

References

Arnsperger, C. 1994. Envy-freeness and distributive justice. *Journal of Economic Surveys*, 8(2): 155–186.

Attouch, H.; Bolte, J.; and Svaiter, B. F. 2013. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming*, 137(1): 91–129.

Aziz, H.; and Ye, C. 2014. Cake cutting algorithms for piecewise constant and piecewise uniform valuations. In *International Conference on Web and Internet Economics*, 1–14. Springer.

Barman, S.; Krishnamurthy, S. K.; and Vaish, R. 2018. Finding fair and efficient allocations. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, 557–574.

Bauschke, H. H.; Bolte, J.; and Teboulle, M. 2017. A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42(2): 330–348.

Beck, A.; and Tetruashvili, L. 2013. On the convergence of block coordinate descent type methods. *SIAM journal on Optimization*, 23(4): 2037–2060.

Birnbaum, B.; Devanur, N. R.; and Xiao, L. 2011. Distributed algorithms via gradient descent for fisher markets. In *Proceedings of the 12th ACM conference on Electronic commerce*, 127–136. ACM.

Budish, E.; Cachon, G. P.; Kessler, J. B.; and Othman, A. 2016. Course match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation. *Operations Research*, 65(2): 314–336.

Chen, X.; and Teng, S.-H. 2009. Spending is not easier than trading: on the computational equivalence of Fisher and Arrow-Debreu equilibria. In *International Symposium on Algorithms and Computation*, 647–656. Springer.

Cole, R.; Devanur, N. R.; Gkatzelis, V.; Jain, K.; Mai, T.; Vazirani, V. V.; and Yazdanbod, S. 2017. Convex program duality, fisher markets, and Nash social welfare. In *18th ACM Conference on Economics and Computation, EC 2017*. Association for Computing Machinery, Inc.

Conitzer, V.; Kroer, C.; Panigrahi, D.; Schrijvers, O.; Sodomka, E.; Stier-Moses, N. E.; and Wilkens, C. 2019. Pacing Equilibrium in First-Price Auction Markets. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. ACM.

Conitzer, V.; Kroer, C.; Sodomka, E.; and Stier-Moses, N. E. 2018. Multiplicative Pacing Equilibria in Auction Markets. In *International Conference on Web and Internet Economics*.

Daskalakis, C.; Goldberg, P. W.; and Papadimitriou, C. H. 2009. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1): 195–259.

Devanur, N. R.; Papadimitriou, C. H.; Saberi, A.; and Vazirani, V. V. 2008. Market equilibrium via a primal-dual algorithm for a convex program. *Journal of the ACM (JACM)*, 55(5): 1–18.

Dooms, S.; De Pessemier, T.; and Martens, L. 2013. Movietweetings: a movie rating dataset collected from twitter. In *Workshop on Crowdsourcing and human computation for recommender systems, CrowdRec at RecSys*, volume 2013, 43.

Eisenberg, E. 1961. Aggregation of utility functions. *Management Science*, 7(4): 337–350.

Eisenberg, E.; and Gale, D. 1959. Consensus of subjective probabilities: The pari-mutuel method. *The Annals of Mathematical Statistics*, 30(1): 165–168.

Gao, T.; Lu, S.; Liu, J.; and Chu, C. 2020. Randomized bregman coordinate descent methods for non-lipschitz optimization. *arXiv preprint arXiv:2001.05202*.

Gao, Y.; and Kroer, C. 2020. First-Order Methods for Large-Scale Market Equilibrium Computation. In *Neural Information Processing Systems 2020, NeurIPS 2020*.

Gao, Y.; and Kroer, C. 2021. Infinite-Dimensional Fisher Markets: Equilibrium, Duality and Optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Garg, R.; and Kapoor, S. 2006. Auction algorithms for market equilibrium. *Mathematics of Operations Research*, 31(4): 714–729.

Goktas, D.; Viqueira, E. A.; and Greenwald, A. 2021. A Consumer-Theoretic Characterization of Fisher Market Equilibria. In *International Conference on Web and Internet Economics*, 334–351. Springer.

Hanzely, F.; and Richtárik, P. 2019. Accelerated coordinate descent with arbitrary sampling and best rates for minibatches. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 304–312. PMLR.

Hanzely, F.; and Richtárik, P. 2021. Fastest rates for stochastic mirror descent methods. *Computational Optimization and Applications*, 1–50.

Held, M.; Wolfe, P.; and Crowder, H. P. 1974. Validation of subgradient optimization. *Mathematical programming*, 6(1): 62–88.

Kantorovich, L. 1975. Mathematics in economics: achievements, difficulties, perspectives. Technical report, Nobel Prize Committee.

Karimi, H.; Nutini, J.; and Schmidt, M. 2016. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 795–811. Springer.

Kroer, C.; Peysakhovich, A.; Sodomka, E.; and Stier-Moses, N. E. 2019. Computing large market equilibria using abstractions. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, 745–746.

Kroer, C.; and Stier-Moses, N. E. 2022. Market equilibrium models in large-scale internet markets. *Innovative technology at the interface of Finance and Operations. Springer Series in Supply Chain Management. Springer Natures*.

Liu, J.; and Wright, S. J. 2015. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, 25(1): 351–376.

Lu, H.; Freund, R. M.; and Nesterov, Y. 2018. Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, 28(1): 333–354.

Mosek, A. 2010. The MOSEK optimization software. *Online at http://www. mosek. com*, 54(2-1): 5.

Nesterov, Y. 2012. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2): 341–362.

Othman, A.; Papadimitriou, C.; and Rubinstein, A. 2016. The complexity of fairness through equilibrium. *ACM Transactions on Economics and Computation (TEAC)*, 4(4): 1–19.

Othman, A.; Sandholm, T.; and Budish, E. 2010. Finding approximate competitive equilibria: efficient and fair course allocation. In *AAMAS*, volume 10, 873–880.

Reddi, S. J.; Sra, S.; Póczos, B.; and Smola, A. 2016. Fast stochastic methods for nonsmooth nonconvex optimization. *arXiv preprint arXiv:1605.06900*.

Richtárik, P.; and Takáč, M. 2014. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1): 1–38.

Rubinsteyn, A.; and Feldman, S. 2016. fancyimpute: An Imputation Library for Python. https://github.com/iskandr/fancyimpute. Accessed: 2022-07-01.

Scarf, H.; et al. 1967. *On the computation of equilibrium prices*. Cowles Foundation for Research in Economics at Yale University New Haven, CT.

Shmyrev, V. I. 2009. An algorithm for finding equilibrium in the linear exchange model with fixed budgets. *Journal of Applied and Industrial Mathematics*, 3(4): 505.

Tseng, P. 2001. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3): 475–494.

Varian, H. R.; et al. 1974. Equity, envy, and efficiency. *Journal of Economic Theory*, 9(1): 63–91.

Vazirani, V. V.; and Yannakakis, M. 2011. Market equilibrium under separable, piecewise-linear, concave utilities. *Journal of the ACM (JACM)*, 58(3): 1–25.

Wright, S. J. 2015. Coordinate descent algorithms. *Mathematical Programming*, 151(1): 3–34.

Zhang, H. 2020. New analysis of linear convergence of gradient-type methods via unifying error bound conditions. *Mathematical Programming*, 180(1): 371–416.

Zhang, L. 2011. Proportional response dynamics in the Fisher market. *Theoretical Computer Science*, 412(24): 2691–2698.