

Accelerated Price Adjustment for Fisher Markets with Exact Recovery of Competitive Equilibrium*

He Chen¹, Chonghe Jiang², and Anthony Man-Cho So^(✉)³

¹ The Chinese University of Hong Kong, Hong Kong SAR, China
hechen@link.cuhk.edu.hk

² Massachusetts Institute of Technology, Cambridge, MA, US
chonghej@mit.edu

³ The Chinese University of Hong Kong, Hong Kong SAR, China
mancho@se.cuhk.edu.hk

Abstract. The canonical price-adjustment process, tâtonnement, typically fails to converge to the exact competitive equilibrium (CE) and requires a high iteration complexity of $\tilde{O}(1/\epsilon)$ to compute ϵ -CE prices in widely studied linear and quasi-linear Fisher markets. This paper proposes refined price-adjustment processes to overcome these limitations. By formulating the task of finding CE of a (quasi-)linear Fisher market as a strongly convex nonsmooth minimization problem, we develop a novel accelerated price-adjustment method (APM) that finds an ϵ -CE price in $\tilde{O}(1/\sqrt{\epsilon})$ lightweight iterations, which significantly improves upon the iteration complexities of tâtonnement methods. Furthermore, through our new formulation, we construct a recovery oracle that maps approximate CE prices to exact CE prices at a low computational cost. By coupling this recovery oracle with APM, we obtain an adaptive price-adjustment method whose iterates converge to CE prices in finite steps. To the best of our knowledge, this is the first convergence guarantee to exact CE for price-adjustment methods in linear and quasi-linear Fisher markets. Our developments pave the way for efficient lightweight computation of CE prices. We also present numerical results to demonstrate the fast convergence of the proposed methods and the efficient recovery of CE prices.

1 Introduction

Competitive equilibrium (CE) is one of the most fundamental concepts in economic theory and has many modern applications, such as the digital ad auction, online resource allocation, and fair division; see, e.g., [23, 3, 36, 46], and the references therein. This concept was initially proposed by Walras [47] in the nineteenth century to characterize the ideal status of a general market, in which every agent sells his initial endowment to buy goods. As a special case of Walras’s model, Fisher considered a market that sells m goods to n buyers, where every buyer has a fixed budget [9]. In such a market, a group of allocations and prices is called a CE if (i) for each buyer, the total price of collected goods does not exceed his budget; (ii) every buyer gets his optimal bundle; (iii) all goods are cleared out.

The existence of CE was unknown until the work of Arrow and Debreu [1] in the 1950s. Later, Eisenberg and Gale [29] showed that the CE of a Fisher market are the Karush-Kuhn-Tucker (KKT) points of a certain convex program. Since then, the computation of CE has received significant attention. Based on the Eisenberg-Gale (EG) program, many polynomial-time algorithms, such as the ellipsoid method [34], interior point method [49], and combinatorial methods [12, 26, 28], were proposed to compute CE. However, these methods adopt a centralized model that precludes parallel computation and requires solving expensive subproblems in each iteration. As markets grow in scale and automated markets prevail in modern applications, there is an increasing need for *lightweight* and *decentralized*⁴ algorithms. Typical examples of such algorithms include proportional response dynamics [5, 48], first-order methods [30, 37], and tâtonnement [31, 18].

* Authors are in α - β order.

⁴ As opposed to centralized methods, we refer to decentralized methods as algorithms that split the update into subtasks, executed in parallel across multiple nodes.

Tâtonnement, proposed by Walras [47], is one of the earliest lightweight and decentralized algorithms for Fisher markets. It refers to a price-adjustment process that increases the price of a good when the demand exceeds the supply and decreases the price when the demand is smaller [18]. Thanks to its purely price-based update rule, tâtonnement requires only $\mathcal{O}(m)$ memory to store the iterates, significantly less than the $\mathcal{O}(mn)$ memory required by other lightweight algorithms, making it appealing in large-scale markets. More importantly, as tâtonnement mimics the price behavior in real-world markets, it has garnered much interest in its theoretical properties, and a long line of work has attempted to establish its convergence. Walras [47] conjectured that tâtonnement converges to CE. Nearly half a century later, Arrow and Hurwicz [2] provided a proof for the convergence of continuous-time tâtonnement under the weak gross substitutes (WGS) assumption. As for the discrete-time tâtonnement, there are different variants and their convergence properties depend on the specific utility functions [16]. Cheung et al. [17] showed that for *constant-elasticity-of-substitution* (CES) utilities⁵ with $0 \leq \rho_i < 1$, the multiplicative tâtonnement achieves a linear convergence rate to CE. Cheung et al. [18] considered entropic tâtonnement and proved that its iterates converge to CE at a linear rate for complementary CES utilities ($-\infty < \rho_i < 0$) and at a sublinear rate for Leontief utilities. Further, in homothetic Fisher markets, Goktas et al. [31] established a convergence rate of $\mathcal{O}((1 + E^2)/T)$ for entropic tâtonnement, where E represents an upper bound on the elasticity of demand and T denotes the iteration number. However, the above convergence results are not applicable to the most commonly used linear and quasi-linear utilities (characterized by parameters $\rho_i = 1$ and $E = +\infty$). Indeed, it has been shown that tâtonnement does not converge to the exact CE in (quasi-)linear Fisher markets (see [19, Example 1]), and only weaker convergence guarantees have been established. Cole and Tao [19] proved that the entropic tâtonnement converges to an approximate CE in linear Fisher markets. Nan et al. [39] showed that for both linear and quasi-linear utilities, the additive tâtonnement converges to an ϵ -CE at a linear rate of $1 - \Theta(\epsilon)$. Consequently, the additive tâtonnement finds an ϵ -CE in $\mathcal{O}(\log(\frac{1}{\epsilon})\frac{1}{\epsilon})$ iterations. In comparison, other lightweight algorithms, such as the mirror descent methods, which iteratively adjust bid vectors [30, 5], converge to the exact CE at a rate of $\mathcal{O}(1/T)$.

From the above discussion, we see that with linear and quasi-linear utilities, the dominant price-adjustment processes—tâtonnement methods—not only fail to converge to a CE but also require a high iteration complexity of $\tilde{\mathcal{O}}(1/\epsilon)$ for finding an ϵ -CE. These limitations motivate the following questions for (quasi-)linear Fisher markets:

- (Q1) *Can we develop a faster price-adjustment process for computing approximate CE prices?*
- (Q2) *Can we refine price-adjustment processes to ensure iterate convergence to exact CE prices?*

1.1 Technical Contributions

In this paper, we develop novel price-adjustment processes to provide affirmative answers to (Q1) and (Q2). Starting from the dual of the EG program, we treat the linear and quasi-linear Fisher markets in a unified manner and formulate the task of computing CE into a box-constrained strongly convex nonsmooth minimization problem. In the new formulation, the objective function merely involves the prices and consists of a sum of exponential and piecewise linear functions. Such a simple structure facilitates the algorithm design. Specifically, by smoothing those piecewise linear functions, we obtain a surrogate problem whose objective function is strongly convex and smooth with a computable modulus. Therefore, acceleration methods can be applied, leading to the so-called *accelerated price-adjustment method* (APM), a novel price-adjustment process distinct from tâtonnement: In each iteration, instead of reacting to the present excess supply, APM predicts the future excess supply to adjust the prices. As one of our main contributions, we prove that APM finds ϵ -CE prices in $\tilde{\mathcal{O}}(1/\sqrt{\epsilon})$ iterations. Such a rate significantly improves upon the iteration complexities of tâtonnement methods, thereby providing an affirmative answer to (Q1). Furthermore, we show that the prices produced by APM can be used to compute an approximate CE allocation, offering an additional advantage over tâtonnement methods.

⁵ CES utilities are of the form $u_i(y) = (\sum_{j \in [m]} a_{ij} y_j^{\rho_i})^{1/\rho_i}$ for $y \in \mathbb{R}_+^m$ with $\rho_i \leq 1$ and $a_{ij} \geq 0$; see, e.g., [18, Definition 2.5].

To address (Q2), as our second main contribution, we construct a *recovery oracle* that maps an ϵ -CE price (for sufficiently small ϵ) to an exact CE price and show that it only requires $\mathcal{O}((m+n)^2)$ arithmetic operations, comparable to the iteration cost of tâtonnement. This result is significant, as it establishes a direct connection between approximate and exact CE prices in (quasi-)linear Fisher markets and provides a tool for price-adjustment processes to compute exact CE. Subsequently, by coupling the recovery oracle with APM (resp. tâtonnement methods), we develop the *adaptive* APM (resp. adaptive tâtonnement), which is guaranteed to find CE prices in finite steps under a practical stopping criterion. To the best of our knowledge, this is the first price-adjustment process with an iterate convergence guarantee to CE prices, providing a definitive answer to (Q2). We note that the techniques developed in this paper, especially the recovery oracle and the acceleration scheme, are novel to the study of computing CE and may also extend to the computation of other market equilibria.

Finally, we demonstrate the practicality and efficiency of the proposed price-adjustment methods via numerical experiments. We compare APM with tâtonnement, mirror descent, and primal-dual hybrid gradient method (PDHG) [37, 11] by evaluating their iteration number for finding an approximate CE price. Our numerical results show that, for both synthetic and real-world datasets, APM requires only about a quarter of the iterations of other algorithms. Furthermore, we compare the CPU time of the adaptive APM with the off-the-shelf solver to compute exact CE prices. It turns out that the adaptive APM is 10-100 times faster than the solver. With these competitive numerical performance and strong theoretical guarantees, the (adaptive) APM provides a substantial refinement over existing price-adjustment processes in (quasi-)linear Fisher markets.

1.2 Further Related Work

Many different variants of the Fisher market have been considered in the literature, leading to a host of new equilibrium notions. Examples include CE for chores [10, 6, 13], CE for public goods allocation [36, 35], and pacing equilibrium in auction markets [24, 21, 22]. Computing these equilibria amounts to finding solutions to different optimization problems, for which techniques from optimization, e.g., error bound [14, 39, 37], duality [20, 30], and penalty method [7], play a central role in obtaining solutions efficiently.

The optimization tools used in this paper basically lie in the field of first-order methods. Let us review them in order. We start with Nesterov’s acceleration, the key tool to the development of our APM. In each iteration, Nesterov’s acceleration method performs a gradient descent step with a carefully chosen stepsize and constructs a linear combination of two consecutive iterates. As the optimal first-order method for convex L -smooth optimization (see, e.g., [44, Chapter 2]), Nesterov’s acceleration method improves the rate of the gradient descent method (GDM) from $\mathcal{O}(1/k)$ to $\mathcal{O}(1/k^2)$, where the convergence measure is the function value gap and k denotes the iteration number. If the objective function is σ -strongly convex, then it accelerates the linear rate of the GDM by a factor of $\sqrt{\sigma/L}$. Some other acceleration methods, e.g., the heavy-ball method [40] and FISTA [4], achieve the same results; see [25] for a survey.

For all these acceleration methods, the smoothness of the objective function is a necessary condition. However, many real applications induce nonsmoothness in objective functions of the corresponding optimization problems. To deal with nonsmooth functions, a popular approach is to apply the smoothing technique, which involves deriving a smooth approximation of a nonsmooth function through appropriate regularization [41, 43]. Such an approximation preserves the convexity of the original problem and hence facilitates the application of acceleration methods. It is noteworthy that Chen et al. [14] has recently used the smoothing technique to develop an unconstrained smooth approximation problem for computing CE for chores. Nevertheless, they failed to accelerate their algorithm due to the nonconvex nature of the chores setting. In comparison, our work shows that for (quasi-)linear Fisher markets, acceleration can be achieved for price-adjustment processes.

On another front, when it comes to the computation of exact CE, the above tools offer little help. A classical technique to compute an exact CE is the interior point algorithm rounding procedure [49, 38]. Assuming that the inputs are rational with bit-length bounded by \mathcal{L} , this procedure rounds an ϵ -KKT point ($\epsilon \leq 2^{-\mathcal{O}(\mathcal{L})}$) into an exact CE through a system of linear equalities and inequalities. Then, together with the modified primal-dual path-following algorithm, it outputs an exact CE in at most $\mathcal{O}(\sqrt{mn}(m+n)^3\mathcal{L})$ arithmetic operations; see [49, Theorem 3]. However, such a procedure needs both ϵ -CE prices and allocations

to solve the required linear system, making it inapplicable to price-adjustment methods that only produce ϵ -CE prices. In contrast, our recovery oracle, based on the local solvability of the nonlinear optimality conditions of the new formulation, only requires approximate CE prices as input. Consequently, it can be coupled with price-adjustment methods to yield finite-step convergence to CE prices. Furthermore, the oracle avoids large linear systems and enjoys a significantly lower complexity of $\mathcal{O}((m+n)^2)$, consistent with the lightweight nature of price-adjustment methods.

Organization. This paper is organized as follows. Sec. 2 introduces the formal definition of CE. Sec. 3 proposes a unified box-constrained strongly convex formulation to compute CE for linear and quasi-linear utilities. In Sec. 4, we develop APM and prove that it finds an ϵ -CE in $\tilde{\mathcal{O}}(1/\sqrt{\epsilon})$ iterations. In Sec. 5, we present the recovery oracle and show how it can be used to recover an exact CE. By coupling the recovery oracle with APM, we further develop the adaptive APM that is guaranteed to find an exact CE in finite steps. In Sec. 6, we report numerical results to show the superior performance of our algorithms. Finally, we give some closing remarks in Sec. 7.

Notation. The notation used in this paper is mostly standard. We use $[m]$ to denote the set $\{1, \dots, m\}$ for any positive integer m . For an index set $J \subseteq [m]$, we use $|J|$ to denote its cardinality. We use e_j , $j \in [m]$ to denote the standard basis vectors in \mathbb{R}^m and let e_0 be the zero vector of \mathbb{R}^m for the sake of consistency. Let \mathcal{D}_m denote the m -dimensional simplex, i.e., $\mathcal{D}_m = \{y \in \mathbb{R}_+^m : \sum_{j \in [m]} y_j = 1\}$. For a vector $y \in \mathbb{R}^m$ and a closed convex set $\mathcal{C} \in \mathbb{R}^m$, let $\Pi_{\mathcal{C}}(y)$ to denote the projection of y onto \mathcal{C} . We adopt the convention that $\log 0 = -\infty$, $\exp(-\infty) = 0$, $[0] = \emptyset$, and $\max_{j \in \emptyset} \{y_j\} = -\infty$.

2 Preliminary: Definition of CE

Consider a Fisher market with m divisible items and n buyers. Each buyer i spends his budget B_i ($B_i > 0$) purchasing a bundle of goods to maximize his utility u_i , which is a function of his allocation vector $x_i \in \mathbb{R}_+^m$. By setting the price vector $p \in \mathbb{R}_+^m$ appropriately, the market would strike a balance at the following equilibrium [30, 1].

Definition 2.1 (Competitive Equilibrium in Fisher Market). *We say that a price vector $p^* \in \mathbb{R}_+^m$ and an allocation $x^* \in \mathbb{R}_+^{n \times m}$ satisfy competitive equilibrium (CE) if and only if*

- (E1). $\sum_{j \in [m]} p_j x_{ij}^* \leq B_i$ for all $i \in [n]$;
- (E2). $u_i(x_i^*) = \max_{x_i \in \mathbb{R}_+^m} \{u_i(x_i) : \sum_{j \in [m]} p_j x_{ij} \leq B_i\}$ for all $i \in [n]$;
- (E3). $\sum_{i \in [n]} x_{ij}^* \leq 1$ for all $j \in [m]$ and the equality must hold if $p_j^* > 0$.

Utility functions u_i , $i \in [n]$ play a crucial role in determining CE as they define the objective functions in the buyer's individual optimization problem (E2). The most simple example of u_i is arguably linear utility, i.e., $u_i : x_i \mapsto \sum_{j \in [m]} v_{ij} x_{ij}$, only parameterized by a utility vector $v_i \in \mathbb{R}_+^m$. Another prevalent example is the quasi-linear utility $u_i : x_i \mapsto \sum_{j \in [m]} (v_{ij} - p_j) x_{ij}$, which represents the utility of the goods purchased minus the total payment. With this utility, a buyer i would prefer not to purchase any goods if the prices p_j exceed his valuations v_{ij} for all $j \in [m]$.

Condition (E3) refers to market clearance. To ensure that all goods are sold out at CE, throughout the paper, we make the following assumption that is commonly adopted in Fisher markets; see, e.g., [49, Assumption 1] and [30, Section 4].

Assumption 1 *Without loss of generality, we assume that the utility matrix $v \in \mathbb{R}^{n \times m}$ is nondegenerate, i.e., it does not allow a zero row or column.*

Assumption 1 ensures that the prices at CE, i.e., p_j^* , $j \in [m]$, are strictly positive (see, e.g., [49, Sec. 2] and [30, Lemma 2]), and hence $\sum_{i \in [n]} x_{ij}^* = 1$ for all $j \in [m]$ by (E3). Further, we have upper and lower bounds on p_j^* , $j \in [m]$ by [30, Lemma 2 and 12].

Lemma 2.1. For linear and quasi-linear utilities, the prices $p_j^*, j \in [m]$ at CE have upper and lower bounds $\bar{p} := (1 - \alpha^{-1})\|B\|_1 + \max_{i \in [n], j \in [m]} \alpha^{-1} v_{ij}$ and $\underline{p} := \min_{j \in [m]} \max_{i \in [n]} \frac{v_{ij} B_i}{\|v_i\|_1 + \alpha^{-1} B_i}$, i.e.,

$$\underline{p} \leq p_j^* \leq \bar{p}, \quad \forall j \in [m],$$

where $\alpha = 1$ (resp. $\alpha = +\infty$) corresponds to quasi-linear (resp. linear) utilities.

3 Unified Strongly Convex Formulation

In this section, we formulate the task of computing CE into a structured strongly convex minimization problem. To begin, we consider the dual EG programs with widely studied linear and quasi-linear utilities [20, 23, 30], whose optimal solution p^* is nothing but CE price vector.

$$\left. \begin{array}{l} \text{Linear:} \\ \min_{p \in \mathbb{R}_+^m} \sum_{j \in [m]} p_j - \sum_{i \in [n]} B_i \log(\beta_i) \\ \text{subject to } p_j \geq v_{ij} \beta_i, \forall i \in [n], j \in [m]; \end{array} \right| \begin{array}{l} \text{Quasi-linear:} \\ \min_{p \in \mathbb{R}_+^m} \sum_{j \in [m]} p_j - \sum_{i \in [n]} B_i \log(\beta_i) \\ \text{subject to } p_j \geq v_{ij} \beta_i, \forall i \in [n], j \in [m] \\ \beta_i \leq 1, \quad \forall i \in [n]. \end{array}$$

To investigate the above two optimization problems in a unified manner, we introduce a parameter $\alpha \in \{1, +\infty\}$ and consider the following formulation:

$$\begin{array}{l} \min_{p \in \mathbb{R}_+^m} \sum_{j \in [m]} p_j - \sum_{i \in [n]} B_i \log(\beta_i) \\ \text{subject to } p_j \geq v_{ij} \beta_i, \quad \forall i \in [n], j \in [m] \\ \beta_i \leq \alpha, \quad \forall i \in [n], \end{array} \quad (3.1)$$

where $\alpha = +\infty$ (resp. $\alpha = 1$) corresponds to linear (resp. quasi-linear) Fisher market setting.

Observe that the objective function of Problem (3.1) is monotonically decreasing with respect to $\beta_i, i \in [n]$. We can omit all the constraints and obtain a problem that merely concerns p .

$$\min_{p \in \mathbb{R}_+^m} f(p) := \sum_{j \in [m]} p_j - \sum_{i \in [n]} B_i \log \left(\min \left\{ \alpha, \frac{p_j}{v_{ij}}, j \in [m] \right\} \right). \quad (3.2)$$

By substituting p_j with $\exp(\mu_j)$ for $j \in [m]$ and introducing the notation $v_{i0} = 1$ and $\mu_0 = \log(\alpha)$, we further simplify the above problem as

$$\min_{\mu \in \mathbb{R}^m} F(\mu) := \sum_{j \in [m]} \exp(\mu_j) + \sum_{i \in [n]} B_i \max_{j \in \{0\} \cup [m]} \{\log(v_{ij}) - \mu_j\}. \quad (\mathcal{P})$$

With the exponential functions $\exp(\mu_j), j \in [m]$ incorporated, the objective function F is strongly convex on \mathbb{R}^m . However, it lacks a global strong convexity modulus. To address this issue, we impose box constraints on $\mu_j, j \in [m]$. Specifically, let μ^* denote the optimal solution of (\mathcal{P}) and use the relations $p_j = \exp(\mu_j), j \in [m]$. By Lemma 2.1, we see that

$$\underline{\mu} := \log(\underline{p}) \leq \mu_j^* \leq \bar{\mu} := \log(\bar{p}), \quad \forall j \in [m]. \quad (3.3)$$

Therefore, we can constrain the variables $\mu_j, j \in [m]$ in the box $[\underline{\mu}, \bar{\mu}]$ without changing the optimal solution of (\mathcal{P}) , leading to the following box-constrained formulation:

$$\begin{array}{l} \min_{\mu \in \mathbb{R}^m} F(\mu) = \sum_{j \in [m]} \exp(\mu_j) + \sum_{i \in [n]} B_i \max_{j \in \{0\} \cup [m]} \{\log(v_{ij}) - \mu_j\} \\ \text{subject to } \underline{\mu} \leq \mu_j \leq \bar{\mu}, \quad \forall j \in [m]. \end{array} \quad (\mathcal{P}^*)$$

Clearly, in the feasible region of (\mathcal{P}^*) , the objective function F is strongly convex with modulus $\sigma_0 = \exp(\underline{\mu})$, and its smooth part, i.e., the exponential sum $\sum_{j \in [m]} \exp(\mu_j)$, has a Lipschitz continuous gradient with modulus $L_0 = \exp(\bar{\mu})$. Moreover, the box constraints allow a simple closed-form projection, where the j -th element of the projection of $\mu \in \mathbb{R}^m$ is given by $\Pi_{[\underline{\mu}, \bar{\mu}]}(\mu_j) = \max\{\underline{\mu}, \min\{\mu_j, \bar{\mu}\}\}$. These properties will prove crucial for accelerating price adjustment.

4 Accelerated Price Adjustment

To tackle the formulation (\mathcal{P}^*) , we first smooth the max terms in the objective function F via the entropy regularization. Specifically, we consider an alternative form of F , i.e.,

$$F(\mu) = \sum_{j \in [m]} \exp(\mu_j) + \sum_{i \in [n]} B_i \max_{\lambda_i \in \mathcal{D}_{m+1}} \left\{ \sum_{j \in [m] \cup \{0\}} \lambda_{ij} (\log(v_{ij}) - \mu_j) \right\}, \quad (4.1)$$

and approximate it by

$$F_\delta(\mu) = \sum_{j \in [m]} \exp(\mu_j) + \sum_{i \in [n]} B_i \max_{\lambda_i \in \mathcal{D}_{m+1}} \left\{ \sum_{j \in \{0\} \cup [m]} \lambda_{ij} (\log(v_{ij}) - \mu_j) - \delta \lambda_{ij} \log(\lambda_{ij}) \right\}. \quad (4.2)$$

Here, $\delta > 0$ is the approximation parameter and the optimal value of the inner maximization problems can be computed by plugging in the optimal solution $\lambda_{ij}^* = \exp(\frac{\log(v_{ij}) - \mu_j}{\delta}) / \sum_{j \in \{0\} \cup [m]} \exp(\frac{\log(v_{ij}) - \mu_j}{\delta})$. Then, by replacing F with F_δ in (\mathcal{P}^*) and relaxing the box constraints $\mu_j \in [\underline{\mu}, \bar{\mu}]$ to $\mu_j \in [\underline{\mu} - \eta, \bar{\mu} + \eta]$, $j \in [m]$ for some $\eta \geq 0$, we obtain the following approximation problem for (\mathcal{P}) :

$$\begin{aligned} \min_{\mu \in \mathbb{R}^m} \quad & F_\delta(\mu) = \sum_{j \in [m]} \exp(\mu_j) + \delta \sum_{i \in [n]} B_i \log \left(\sum_{j \in \{0\} \cup [m]} \exp \left(\frac{\log(v_{ij}) - \mu_j}{\delta} \right) \right) \\ \text{subject to} \quad & \underline{\mu} - \eta \leq \mu_j \leq \bar{\mu} + \eta, \quad \forall j \in [m]. \end{aligned} \quad (\mathcal{P}_\delta)$$

We have the following properties for the approximation problem (\mathcal{P}_δ) .

Fact 4.1 (Properties of Problem (\mathcal{P}_δ)). *The following holds:*

- (i) *Smoothness:* F_δ is L -smooth in the box $[(\underline{\mu} - \eta)\mathbf{1}_m, (\bar{\mu} + \eta)\mathbf{1}_m]$, where $L = \exp(\bar{\mu} + \eta) + \|B\|_1/\delta$.
- (ii) *Strong convexity:* F_δ is σ -strongly convex on the box constrain set, where $\sigma = \exp(\underline{\mu} - \eta)$.
- (iii) *Approximation error:* $F \leq F_\delta \leq F + \delta \log(m+1)\|B\|_1$.
- (iv) *Subgradient approximation:* $\lim_{\delta \rightarrow 0} \nabla F_\delta(\mu) \in \partial F(\mu)$ for all $\mu \in \mathbb{R}^m$.

Now, we are ready to present our accelerated price-adjustment method (APM), which consists of a gradient step and an acceleration step. The gradient step computes the gradient of the approximation function F_δ and performs the projection onto the box constraints, where the computation cost is $\mathcal{O}(mn)$. The acceleration step computes a simple linear combination at a cost of $\mathcal{O}(m)$. Therefore, the total iteration cost of APM is $\mathcal{O}(mn)$, identical to that of tâtonnement methods. Furthermore, by selecting an appropriate relaxation radius η , APM includes a practical stopping criterion to identify an ϵ -CE price.

Accelerated Price-adjustment Method (APM)	
Parameters:	$\epsilon \in (0, \exp(\underline{\mu} - 1)]$; $\delta = \epsilon / (2 \log(m+1) \ B\ _1)$; $\eta = 1$; $L = \exp(\bar{\mu} + \eta) + \ B\ _1/\delta$; $\sigma = \exp(\underline{\mu} - \eta)$; $q = \sigma/L$.
Initialization:	$\mu^0 \in [\underline{\mu}\mathbf{1}_m, \bar{\mu}\mathbf{1}_m]$; $y^0 = \mu^0$;
Stopping Criterion:	$\ \nabla F_\delta(\mu^t)\ _2 \leq \min\{\sigma\epsilon, \sqrt{\sigma\epsilon}\}$;
Gradient Step:	$\mu_j^{t+1} = \max\left\{ \underline{\mu} - \eta, \min\left\{ y_j^t - \frac{1}{L} \nabla_j F_\delta(y^t), \bar{\mu} + \eta \right\} \right\}$, $\forall j \in [m]$;
Acceleration Step:	$y^{t+1} = \mu^{t+1} + \frac{1 - \sqrt{q}}{1 + \sqrt{q}} (\mu^{t+1} - \mu^t)$.

APM has an economic interpretation: It mimics a market that predicts future prices and excess supply to adjust prices. Specifically, future prices $\exp(y_j^t)$ are estimated as current prices $\exp(\mu_j^t)$ multiplied by a price momentum $\exp(\frac{1-\sqrt{q}}{1+\sqrt{q}}(\mu_j^t - \mu_j^{t-1}))$. Subsequently, with the smooth function ∇F_δ approximating the excess supply, the future excess supply is predicted by $\nabla F_\delta(y^t)$. Then, instead of reacting to the current excess supply $v \in \partial F(\mu^t)$, the market uses the predicted future excess supply $\nabla F_\delta(y^t)$ to adjust the prices. As we shall show, through such a price-adjustment process, the prices will converge to approximate CE prices at a faster rate.

To derive the convergence rate for APM, we first define the approximate CE measure by the function value gap $F(\mu) - \inf F$, which equals zero if and only if $\mu_j = \mu_j^* = \log(p_j^*), j \in [m]$. We note that this approximate CE measure is stronger than the distance square measure in [13, Theorem 1 and Corollary 1], as $\|p - p^*\|_2^2 \leq \mathcal{O}(\epsilon)$ can be implied by $f(p) - f(p^*) = F(\mu) - F(\mu^*) \leq \mathcal{O}(\epsilon)$ due to the quadratic growth of f (see Nan et al. [39, Lemma 4]).

Definition 4.1 (Approximate CE Prices). *We say that $p \in \mathbb{R}^m$ is an ϵ -CE price vector if for $\mu_j = \log(p_j), j \in [m]$, one has $F(\mu) - \inf F \leq \epsilon$.*

Now, we are prepared to establish the convergence rate for APM.

Theorem 4.1. *APM finds an ϵ -CE price vector in at most $\tilde{\mathcal{O}}\left(\frac{1}{\sqrt{\epsilon}}\right)$ iterations.*

To the best of our knowledge, APM is the first price-adjustment process to achieve an iteration complexity of $\tilde{\mathcal{O}}(1/\sqrt{\epsilon})$ in (quasi-)linear Fisher markets, representing a significant improvement over tâtonnement methods. We thereby address (Q1). Furthermore, we show that the prices produced by APM can be utilized to compute an approximate CE allocation.

Proposition 4.1. *Consider the output μ of APM that satisfies the stopping criterion with $\epsilon \leq \log(m+1)\|B\|_1$. The corresponding prices $p_j = \exp(\mu_j), j \in [m]$ and allocation $x \in \mathbb{R}_+^{n \times m}$ given by $x_{ij} = \frac{B_i}{p_j} \cdot \frac{\exp(\frac{\log(v_{ij}) - \mu_j}{\delta})}{\sum_{j \in \{0\} \cup [m]} \exp(\frac{\log(v_{ij}) - \mu_j}{\delta})}, i \in [n], j \in [m]$, satisfy the following:*

- (B1). $\sum_{j \in [m]} p_j x_{ij} \leq B_i$ for all $i \in [n]$;
- (B2). $u_i(x_i) + \alpha^{-1} B_i \geq \left(1 - \frac{2\epsilon}{\|B\|_1}\right) \max_{x'_i \in \mathbb{R}_+^n} \left\{ u_i(x'_i) + \alpha^{-1} B_i : \sum_{j \in [m]} p_j x'_{ij} \leq B_i \right\}$ for all $i \in [n]$;
- (B3). $|\sum_{i \in [n]} x_{ij} - 1| \leq \epsilon$ for all $j \in [m]$.

Note that (B1) coincides with (A1), while (B2)⁶ and (B3) are approximate versions of (A2) and (A3), respectively. We see that the computed allocation x and price p satisfy the CE conditions up to an error of $\mathcal{O}(\epsilon)$. Therefore, APM yields not only approximate CE prices but also an approximate CE allocation. This additional advantage stems from our smoothing strategy and stopping criterion based on $\|\nabla F_\delta(\mu^t)\|$. In comparison, traditional tâtonnement methods, due to their nonsmooth formulations (typically (3.2)), lack such a stopping criterion, making it difficult for them to compute an approximate CE allocation.

5 Recovery of Exact CE

Though APM finds an approximate CE price at a fast rate, it shares the same drawback with existing price-adjustment methods, i.e., the lack of a convergence guarantee to exact CE prices. To address this gap, we explore the relationship between approximate and exact CE prices. It turns out that through the optimality conditions of our formulation (\mathcal{P}) at the optimal solution μ^* , one can develop a *recovery oracle* \mathcal{R} to bridge them, as specified in the following theorem.

⁶ (B2) includes $\alpha^{-1} B_i$ to ensure that both sides remain nonnegative under quasi-linear utilities, where $\alpha = 1$.

Theorem 5.1 (Recovery of Exact CE). *There exists a constant $\Delta^* > 0$ and an oracle \mathcal{R} that maps $\mu \in \mathbb{R}^m$ and $r \in \mathbb{R}$ to μ^* (denoted by $\mathcal{R}(\mu, r) = \mu^*$) whenever $\mu \in \mathbb{B}(\mu^*, r)$ and $0 < r < \Delta^*/4$. Furthermore, the oracle \mathcal{R} completes this mapping in at most $\mathcal{O}((m+n)^2)$ arithmetic operations.*

Theorem 5.1 is significant as it not only reveals that the exact CE prices can be recovered from its nearby points, i.e., ϵ -CE prices, but also demonstrates the low computational cost of the recovery oracle \mathcal{R} , which is comparable to a single tâtonnement iteration. With the recovery oracle in hand, one can refine APM or other price-adjustment methods by coupling them with \mathcal{R} , thereby achieving an iterate convergence guarantee to CE, as demonstrated in Sec. 5.2.

The construction of the recovery oracle \mathcal{R} is rooted in the local property of our formulation (\mathcal{P}) and the so-called *connection class* (see Definition 5.1) of the active index sets

$$\mathcal{J}_i(\mu) := \arg \max_{j \in \{0\} \cup [m]} \{\log(v_{ij}) - \mu_j\}, \quad i \in [n].$$

The key observations are: (i) given all the connection classes of $\mathcal{J}_i(\mu^*)$, $i \in [n]$, the optimal solution μ^* can be exactly calculated via some basic mathematical operations; (ii) there exists a radius $\Delta^* > 0$ such that $\mathcal{J}_i(\mu^*)$, $i \in [n]$, along with their connection classes, can be computed through an arbitrary point $\mu \in \mathbb{R}^m$ with $\|\mu - \mu^*\|_2 < \Delta^*/4$. See Appendix 5.1 for construction details.

For the radius Δ^* , we remark that it only depends on the optimal solution μ^* and the utility parameters v_{ij} , which ultimately reduces to dependence only on the parameters v_{ij} , B_i . Indeed, Δ^* is given by $\Delta^* := \Delta(\mu^*)$, where the function $\Delta : \mathbb{R}^m \rightarrow \mathbb{R}$ is defined as follows:

$$\Delta_i(\mu) := \max_{j \in \{0\} \cup [m]} \{\log(v_{ij}) - \mu_j\} - \max_{j \in (\{0\} \cup [m]) \setminus \mathcal{J}_i(\mu)} \{\log(v_{ij}) - \mu_j\}; \quad \Delta(\mu) := \min_{i \in [n]} \Delta_i(\mu). \quad (5.1)$$

The above definition yields a natural economic interpretation. Specifically, write $\mu_j^* = \log(p_j^*)$ and $\Delta_i(\mu^*) = \log(\frac{v_{ij_1}}{p_{j_1}^*} / \frac{v_{ij_2}}{p_{j_2}^*})$, where $j_1 \in \mathcal{J}_i(\mu^*)$ and $j_2 \in \arg \max_{j \in (\{0\} \cup [m]) \setminus \mathcal{J}_i(\mu^*)} \{\log(v_{ij}/p_j^*)\}$. We see that $\Delta_i(\mu^*)$ represents the logarithm of the ratio between the highest and the second highest bang-per-buck value for buyer i . Hence, $\Delta^* = \min_{i \in [n]} \Delta_i(\mu^*)$ indicates the smallest bang-per-buck gap between buyers' top two choice at CE. Intuitively, a larger gap makes it easier to rank demands and determine CE prices across goods, which aligns with the result of Theorem 5.1.

Remark 5.1. With μ^* obtained, the CE prices can be recovered by letting $p_j^* = \exp(\mu_j^*)$, $j \in [m]$. Furthermore, one can recover the CE allocation x^* by computing a feasible solution $\lambda^* \in \mathbb{R}^{n \times (m+1)}$ for the following linear system and letting $x_{ij}^* = \lambda_{ij}^*/p_j^*$, $i \in [n], j \in [m]$.

$$\begin{aligned} \sum_{j \in \{0\} \cup [m]} \lambda_{ij} &= B_i, & \forall i \in [n]; & \quad \sum_{i \in [n]} \lambda_{ij} = \exp(\mu_j^*), \forall j \in [m]; \\ \lambda_{ij} &\geq 0, \quad \forall j \in \{0\} \cup [m], i \in [n]; & \quad \lambda_{ij} &= 0, \quad \forall j \notin \mathcal{J}_i(\mu^*), i \in [n]. \end{aligned} \quad (5.2)$$

5.1 Construction of Recovery Oracle

The foundation of our recovery oracle lies in the following observation: In Problem (\mathcal{P}), if the active index sets at the optimal solution μ^* are obtained, then μ^* can be computed directly from the optimality conditions. To illustrate this, we recall the maximum functions $h_i(\mu) = \max_{j \in \{0\} \cup [m]} \{\log(v_{ij}) - \mu_j\}$ and the index functions $\mathcal{J}_i(\mu) = \arg \max_{j \in \{0\} \cup [m]} \{\log(v_{ij}) - \mu_j\}$, $i \in [n]$. We have the following optimality condition of (\mathcal{P}):

$$(\exp(\mu_1), \exp(\mu_2), \dots, \exp(\mu_m)) - \sum_{i \in [n]} \lambda_i = 0; \quad (5.3)$$

$$\text{where } \lambda_i \in -B_i \partial h_i(\mu) = \text{conv} \{B_i e_j : j \in \mathcal{J}_i(\mu)\}.$$

The key to solving this equation is thoroughly examining the intersection properties of active index sets $\mathcal{J}_i(\mu)$, $i \in [n]$. For this purpose, we define the so-called connection class for $\mathcal{J}_i(\mu)$, $i \in [n]$.

Definition 5.1 (Connection Class of Index Sets). Given nonempty index sets $J_i \subseteq [n], i \in [n]$, we say that J_i connects $J_{i'}$, denoted by $J_i \sim J_{i'}$, if $J_i \cap J_{i'} \neq \emptyset$ or there exist indices $i_1, i_2, \dots, i_s, s \in [n]$ such that

$$J_i \cap J_{i_1} \neq \emptyset; \quad J_{i_1} \cap J_{i_2} \neq \emptyset; \quad \dots; \quad J_{i_{s-1}} \cap J_{i_s} \neq \emptyset; \quad J_{i_s} \cap J_{i'} \neq \emptyset.$$

Otherwise, we say that J_i disconnects $J_{i'}$, denoted by $J_i \not\sim J_{i'}$. Further, we say that $\{J_i : i \in I \subseteq [n]\}$ form a connection class if $J_i \sim J_{i'}$ for all $i, i' \in I$ and $J_i \not\sim J_{i'}$ for $i \in I, i' \in [n] \setminus I$.

Clearly, the connection class is an equivalence class and satisfies (i) $J_i \sim J_i$; (ii) if $J_i \sim J_{i'}$, then $J_{i'} \sim J_i$; (iii) if $J_{i_1} \sim J_{i_2}$ and $J_{i_2} \sim J_{i_3}$, then $J_{i_1} \sim J_{i_3}$. Moreover, each index set J_i is contained in a unique connection class. Based on the connection class, we have the following lemma that serves as the foundation of the recovery oracle.

Lemma 5.1. Let μ^* be the optimal solution of Problem (\mathcal{P}). Suppose that the index sets $\mathcal{J}_i(\mu^*) = J_i^*, i \in [n]$ are given. Let $\{J_l^* : l \in [s]\}$ be all connection classes of $J_i^*, i \in [n]$ and $\tilde{J}_l^* = \bigcup_{i \in I_l^*} J_i^*, l \in [s]$. Then, $[m] \subseteq \bigcup_{l \in [s]} \tilde{J}_l^* = \bigcup_{i \in [n]} J_i^*$ and the optimality condition (5.3) reduces to the following equations over $l \in [s]$, which admits an exact solution:

$$\log(v_{ij}) - \mu_j = \log(v_{ij'}) - \mu_{j'}, \quad \forall j, j' \in J_i^*, i \in I_l^*; \quad (5.4)$$

$$\sum_{j \in \tilde{J}_l^*} \exp(\mu_j) = \sum_{i \in I_l^*} B_i \quad \text{if } 0 \notin \tilde{J}_l^*. \quad (5.5)$$

Given Lemma 5.1, two natural questions arise: (i) how to derive the active index sets $\mathcal{J}_i(\mu^*), i \in [n]$; (ii) how to classify the connection classes of the active index sets efficiently. For question (i), we show that $\mathcal{J}_i(\mu^*), i \in [n]$ can be deduced from the relaxed active index sets at nearby points of μ^* .

Lemma 5.2. Let μ^* be the optimal solution of Problem (\mathcal{P}) and $\mu \in \mathbb{B}(\mu^*, r)$ with $0 < r < \Delta^*/4$. Then, we have $\mathcal{J}_i(\mu^*) = \{j \in \{0\} \cup [m] : \log(v_{ij}) - \mu_j \geq h_i(\mu) - 2r\}, i \in [n]$.

Algorithm 1 Classification Procedure \mathcal{E}

Input: Index sets $J_i \subseteq \{0\} \cup [m], i \in [n]$

```

1:  $s = 0, I^c = [n]$  ▷  $I^c$  represents the set of unclassified indices
2: while  $I^c \neq \emptyset$  do
3:    $s = s + 1,$ 
4:   Select an arbitrary index  $i^s \in I^c$  ▷ Find an unclassified index  $i^s \in [n]$ 
5:    $I_s = \{i^s\}, I^c = I^c \setminus \{i^s\}$ 
6:    $\tilde{J}_s = \bigcup_{i \in I_s} J_i = J_{i^s}, J_{check} = \emptyset, t = 1$ 
7:   while  $J_{check} \subsetneq \tilde{J}_s$  do
8:     Select an arbitrary index  $j_t^s \in \tilde{J}_s \setminus J_{check}$  ▷ Use indices of  $\tilde{J}_s$  to find connected sets
9:      $\mathcal{I}_{new}(j_t^s) = \{i \in I^c : j_t^s \in J_i\}$ 
10:     $I_s = I_s \cup \mathcal{I}_{new}(j_t^s)$ 
11:     $I^c = I^c \setminus \mathcal{I}_{new}(j_t^s)$ 
12:     $\mathcal{J}_{new}(j_t^s) = \bigcup_{i \in \mathcal{I}_{new}(j_t^s)} J_i;$ 
13:     $\tilde{J}_s = \tilde{J}_s \cup \mathcal{J}_{new}(j_t^s)$ 
14:     $J_{check} = J_{check} \cup \{j_t^s\}, t = t + 1$  ▷ We see that  $j_t^s \in J_{i^s} \cup (\bigcup_{\tau \in [t-1]} \bigcup_{i \in \mathcal{I}_{new}(j_\tau^s)} J_i)$ 
15:   end while
16: end while
Output:  $I_l, \tilde{J}_l$  for  $l \in [s]$ ;  $i^l, j_t^l, \mathcal{I}_{new}(j_t^l)$  for  $t \in [|\tilde{J}_l|], l \in [s]$ 

```

For question (ii), we develop an algorithm, i.e., Algorithm 1, that classifies the connection classes of the input index sets $J_i \subseteq \{0\} \cup [m], i \in [n]$. The main idea is to iteratively incorporate sets that share the elements

of the current connected sets. Specifically, we consider the connected sets $J_i, i \in I$ and select an index j from their combination, i.e., $j \in \tilde{J} := \bigcup_{i \in I} J_i$. By identifying the sets $J_i, i \in \mathcal{I}_{new}(j)$ that also contain the element j , we enlarge the connected sets to $J_i, i \in I \cup \mathcal{I}_{new}(j)$, and then update $\tilde{J} = \bigcup_{i \in I \cup \mathcal{I}_{new}} J_i, I = I \cup \mathcal{I}_{new}$. Repeat this enlarging process until no new connected sets can be found. Then, the connected sets obtained form a connection class. To identify other connection classes and complete the classification, we simply start with unclassified sets and repeat the enlarging process. The validity of Algorithm 1 is ensured by the following lemma.

Lemma 5.3. *The classification procedure \mathcal{E} (i.e., Algorithm 1) satisfies the followings:*

- (i) *The output $\{J_i : i \in I_l\}, l \in [s]$ form connection classes of $J_i, i \in [n]$ with $\bigcup_{i \in I_l} J_i = \tilde{J}_l$.*
- (ii) *The computational cost of \mathcal{E} is at most $\mathcal{O}((m+n)^2)$.*

In each iteration, Algorithm 1 records the indices of the newly incorporated sets that contain the index j_t^l . These indices facilitate an explicit solution procedure \mathcal{P} (Algorithm 2) for the equations (5.4) and (5.5), allowing parallel computation across different l . The underlying idea of \mathcal{P} is to utilize the index sets $\mathcal{I}_{new}(j_t^l), t \in [|\tilde{J}_l|]$ and (5.4) to compute the constants $a_j, j \in \tilde{J}_l$ such that $\mu_j = \mu_{j_t^l} + a_j$, and then substitute $\mu_j = \mu_{j_t^l} + a_j$ into (5.5) to derive a solution.

Algorithm 2 Solution Procedure \mathcal{P}

Input: I_l, \tilde{J}_l for $l \in [s]$; $i^l, j_t^l, \mathcal{I}_{new}(j_t^l)$ for $t \in [|\tilde{J}_l|], l \in [s]$

- 1: **for** $l = 1, 2, \dots, s$ **do**
- 2: $a_{j_1^l} = 0, J_{done} = J_{i^l}$
- 3: $a_j = \log(v_{i^l j}) - \log(v_{i^l j_1^l}), \quad \forall j \in J_i$ $\triangleright a_{j_t^l}$ is computed before the t -th inner iteration
- 4: **for** $t = 1, \dots, |\tilde{J}_l|$ **do**
- 5: $a_j = a_{j_t^l} + \log(v_{ij}) - \log(v_{ij_t^l}), \quad \forall j \in J_i \setminus J_{done}, i \in \mathcal{I}_{new}(j_t^l)$
- 6: $J_{done} = J_{done} \cup (\bigcup_{i \in \mathcal{I}_{new}(j_t^l)} J_i)$
- 7: **end for**
- 8: **if** $0 \in \tilde{J}_l$ **then**
- 9: $\mu_{j_1^l} = -a_0$ $\triangleright a_j, j \in \tilde{J}_l$ are computed in the "for" loop w.r.t. t
- 10: $\mu_j = \mu_{j_1^l} + a_j, \quad \forall j \in \tilde{J}_l$
- 11: **else**
- 12: $\mu_{j_1^l} = \log\left(\sum_{i \in I_l} B_i\right) - \log\left(\sum_{j \in \tilde{J}_l} \exp(a_j)\right)$
- 13: $\mu_j = \mu_{j_1^l} + a_j, \quad \forall j \in \tilde{J}_l$
- 14: **end if**
- 15: **end for**

Output: μ

Lemma 5.4 (Property of \mathcal{P}). *Let μ^* be the optimal solution of Problem (\mathcal{P}) and $J_i^* = \mathcal{J}_i(\mu^*), i \in [n]$. We have $\mathcal{P}(\mathcal{E}(\{J_i^*, i \in [n]\})) = \mu^*$, where the computational cost of \mathcal{P} is at most $\mathcal{O}(mn)$.*

Now, we are ready to give our recovery oracle \mathcal{R} in Algorithm 3. With the input $\mu \in \mathbb{R}^m$ and $r \in \mathbb{R}$, we first define $J_i := \{j \in \{0\} \cup [m] : \log(v_{ij}) - \mu_j \geq h_i(\mu) - 2r\}$. By Lemma 5.2, we know that $J_i = \mathcal{J}_i(\mu^*)$ when $\mu \in \mathbb{B}(\mu^*, r)$ for $0 < r < \Delta^*/4$. Then, we apply the classification procedure \mathcal{E} and obtain the connection classes of $J_i, i \in [n]$. Subsequently, the solution procedure \mathcal{P} is implemented to output a vector $\hat{\mu}$. By Lemma 5.4, the recovery $\hat{\mu} = \mu^*$ is guaranteed when $J_i = \mathcal{J}_i(\mu^*)$. Hence, we have $\mathcal{R}(\mu, r) = \mu^*$ for $\mu \in \mathbb{B}(\mu^*, r)$ with $0 < r < \Delta^*/4$. Moreover, the computational cost of \mathcal{R} is dominated by the classification procedure \mathcal{E} and solution procedure \mathcal{P} , leading to a total cost of $\mathcal{O}((m+n)^2)$ by Lemma 5.3 and 5.4. We summarize these results in the following proposition, which directly proves Theorem 5.1.

Algorithm 3 Recovery oracle \mathcal{R}

Input: Parameter $r > 0$ and vector $\mu \in \mathbb{R}^m$

- 1: Let $h_i = \max_{j \in \{0\} \cup [m]} \{\log(v_{ij}) - \mu_j\}$, $i \in [n]$
- 2: Compute the index sets $J_i = \{j \in \{0\} \cup [m] : \log(v_{ij}) - \mu_j \geq h_i - 2r\}$, $i \in [n]$
- 3: **if** $[m] \subseteq \bigcup_{i \in [n]} J_i$ **then**
- 4: $\hat{\mu} = \mathcal{P}(\mathcal{E}(\{J_i, i \in [n]\})) \in \mathbb{R}^m$
- 5: **else**
- 6: $\hat{\mu} = \text{NaN} \notin \mathbb{R}^m$
- 7: **end if**

Output: $\hat{\mu}$

Proposition 5.1. *Let μ^* be the optimal solution of Problem (\mathcal{P}) and Δ^* be defined by (5.1). Given the input $\mu \in \mathbb{R}^m$, $r > 0$, let $\hat{\mu} \in \mathbb{R}^m$ be the output of the recovery oracle \mathcal{R} , i.e., $\hat{\mu} = \mathcal{R}(\mu, r)$. The following holds:*

- (i) *If the inputs (μ, r) satisfy $0 < r < \Delta^*/4$ and $\mu \in \mathbb{B}(\mu^*, r)$, then we have $\hat{\mu} = \mu^*$.*
- (ii) *The computational cost of \mathcal{R} is at most $\mathcal{O}((m+n)^2)$.*

Proof. (i) Given $\mu \in \mathbb{B}(\mu^*, r)$ and $0 < r < \Delta^*/4$, Lemma 5.2 ensures $J_i = \mathcal{J}_i(\mu^*)$, $i \in [n]$. Then, we have $[m] \subseteq \bigcup_{i \in [n]} J_i$ by Lemma 5.1, and further $\hat{\mu} = \mathcal{P}(\mathcal{E}(\{J_i(\mu^*), i \in [n]\}))$ by Line 4 of Algorithm 3. The desired $\hat{\mu} = \mu^*$ follows from Lemma 5.4.

(ii) Clearly, in Algorithm 3, the total computational cost of evaluating $h_i, J_i, i \in [n]$, and $\bigcup_{i \in [n]} J_i$ is of the order $\mathcal{O}(mn)$. Hence, it suffices to show that the procedures \mathcal{E}, \mathcal{P} have a complexity of $\mathcal{O}((m+n)^2)$, which is ensured by Lemma 5.3 and 5.4.

5.2 Adaptive APM

Given Theorem 5.1, a naive approach to compute the exact CE prices is to apply the oracle \mathcal{R} to the last iteration of APM: By the σ -strong convexity of F on the box $[(\underline{\mu} - \eta)\mathbf{1}, (\underline{\mu} + \eta)\mathbf{1}]$, the output of APM with $\epsilon < \sigma \cdot (\Delta^*)^2/32$, denoted by μ , satisfies $\|\mu - \mu^*\|_2 < \Delta^*/4$. Hence, one may expect to recover μ^* via $\mathcal{R}(\mu, \sqrt{2\epsilon/\sigma})$. However, such a scheme is impractical, as the radius Δ^* is unknown, making it impossible to set an appropriate accuracy ϵ for APM.

An improved approach is to find a sequence of APM outputs $\{\mu^k\}_{k \geq 0}$ with decaying accuracy $\{\epsilon_k\}_{k \geq 0}$, i.e., $F(\mu^k) - F(\mu^*) \leq \epsilon_k$ with $\epsilon_k \rightarrow 0$, and subsequently compute $\hat{\mu}^k = \mathcal{R}(\mu^k, \sqrt{2\epsilon_k/\sigma})$, $k \geq 0$. We call this algorithm *adaptive APM*. Clearly, there exists an index $K > 0$ such that $\sqrt{2\epsilon_k/\sigma} < \Delta^*/4$ for all $k \geq K$. It follows that $\hat{\mu}^k = \mu^*$ for all $k \geq K$ by Theorem 5.1. Therefore, the adaptive APM finds CE in finite steps.

To stop the adaptive APM, it suffices to test the optimality of $\hat{\mu}^k$, $k \geq 0$ for (\mathcal{P}) . The test is equivalent to checking the feasibility of the linear system (5.2) with μ^* replaced by $\hat{\mu}^k$, which can be further formulated as a max-flow problem and efficiently solved, as detailed in Appendix C. We now formally present the adaptive APM as follows.

Adaptive APM	
Input:	$\{\epsilon_k\}_{k \geq 1}$ with $\epsilon_k = \theta^k$, $\theta \in (0, 1)$.
Inner Solver:	APM with strong convexity modulus σ .
Stopping Criterion:	$\hat{\mu}^k$ satisfies the optimality conditions of Problem (\mathcal{P}) .
Approximate Step:	Run the inner solver to find μ^k satisfying $F(\mu^k) - F(\mu^*) \leq \epsilon_k$.
Recovery Step:	$\hat{\mu}^k = \mathcal{R}(\mu^k, \sqrt{2\epsilon_k/\sigma})$.

Theorem 5.2. *The adaptive APM finds CE prices in at most K iterations with*

$$K = \max \left\{ \left\lceil 2 \log_{\theta} \left(\frac{\sqrt{\sigma} \Delta^*}{6} \right) \right\rceil, 1 \right\}.$$

To the best of our knowledge, Theorem 5.2 provides the first convergence guarantee to exact CE prices for price-adjustment processes in (quasi-)linear Fisher markets, thereby addressing (Q2). This demonstrates the power of our recovery oracle. Theorem 5.2 also has an economic interpretation: As the market adapts the prices with diminishing increments (i.e., $\epsilon_k \rightarrow 0$ and the stepsize of the inner solver goes to zero), the iterate prices would well approximate the CE prices, providing sufficient information for sellers and buyers to determine satisfactory prices (i.e., CE prices).

Remark 5.2. By combining Theorem 5.2 and 4.1, we see that the adaptive APM needs at most

$$\sum_{k=1}^K \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{\epsilon_k}}\right) \leq \tilde{\mathcal{O}}\left(\frac{K}{\sqrt{\epsilon_K}}\right) = \tilde{\mathcal{O}}\left(\frac{1}{\Delta^*}\right)$$

APM iterations to compute the exact CE.

Remark 5.3. In the adaptive APM framework, if we substitute the inner solver with the additive tâtonnement and replace the strong convexity modulus σ with the quadratic growth modulus α in [13, Lemma 4], the result of Theorem 5.2 remains valid, with σ replaced by α .

Proof (Proof of Theorem 5.2). We first show that the adaptive APM stops in K iterations. Notice that the outputs μ^k , $k \geq 0$ of APM are in the box $[(\underline{\mu} - \eta)\mathbf{1}, (\underline{\mu} + \eta)\mathbf{1}]$ and F is σ -strongly convex on this box. We have $F(\mu^k) - F(\mu^*) \geq \frac{\sigma}{2} \|\mu^k - \mu^*\|_2^2$ for all $k \geq 0$. On the other hand, the approximation step ensures $F(\mu^k) - F(\mu^*) \leq \epsilon_k$. We have

$$\|\mu^k - \mu^*\|_2 \leq \sqrt{\frac{2}{\sigma} \epsilon_k}, \quad \forall k \geq 0. \quad (5.6)$$

By the definition of K , for $k \geq K$, we have $\epsilon_k \leq \theta^K \leq \sigma(\Delta^*)^2/36$, or equivalently, $\sqrt{2\epsilon_k/\sigma} \leq \Delta^*/(3\sqrt{2})$. This, together with (5.6) and Theorem 5.1, implies $\hat{\mu}^k = \mu^*$ for $k \geq K$. We complete the proof.

6 Experiments

In this section, we demonstrate the competitive numerical performance of our APM in computing approximate CE prices. Moreover, we show that the adaptive APM can successfully recover exact CE prices, requiring much less time than the off-the-shelf solver. We run all experiments on a personal desktop that uses the Apple M3 Pro Chip and has 18GB RAM. The optimization solver implemented is Mosek version 10.2.3.

6.1 APM for Computing Approximate CE

We evaluate the number of iterations needed by different algorithms for finding approximate CE prices. We consider both linear and quasi-linear utility and collect data for utility parameters from synthetic and real-world datasets under different instance sizes. The budgets B_i , $i \in [n]$ are set as 1, following the commonly adopted equal income setting [30]. We use the additive tâtonnement method [39], the mirror descent method [30, 5], and PDHG [37] as benchmarks, as they are representative of tâtonnement, proportional response dynamics, and first-order methods respectively. Moreover, they share the same iteration computational cost of $\mathcal{O}(mn)$ with APM.

To implement the mirror descent method to compute approximate CE prices, we use its iterates, i.e., the bid vectors, to recover prices based on their relations (see, e.g., [5, Equation (2)]). Following Gao and Kroer [30], we set the stepsize of the mirror descent to 1. The stepsize of additive tâtonnement is set to 10^{-4} . Adaptive stepsize strategy is also tested for them. Following Huang et al. [33], we use adaptive stepsize for PDHG based on the primal and dual residual. We set the stepsize of APM to $1/L$ and iteratively decrease δ , where L is the smoothness modulus of F_δ . For synthetic data, the utility parameters v_{ij} , $i \in [n], j \in [m]$ are generated by distributions usually adopted in the literature. For real-world data, the utility matrix is constructed from a movie rating dataset [27], where each movie is treated as an item and each rater is treated

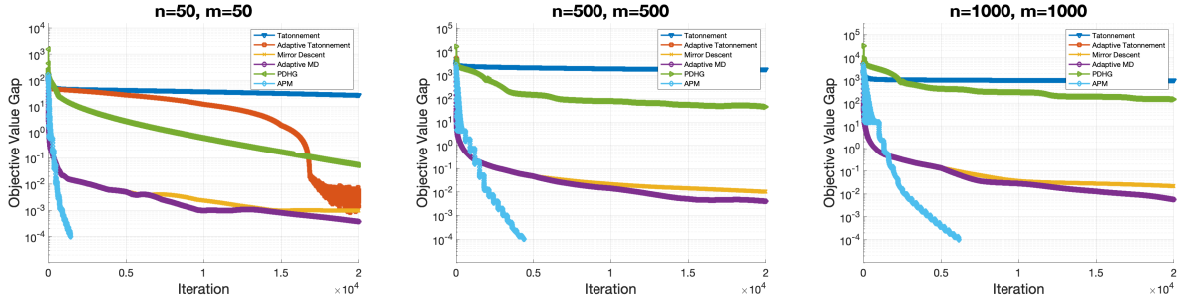


Fig. 1: Comparison of different algorithms for computing ϵ -CE prices in *linear* Fisher markets with uniform data, where the precision ϵ is 10^{-4} .

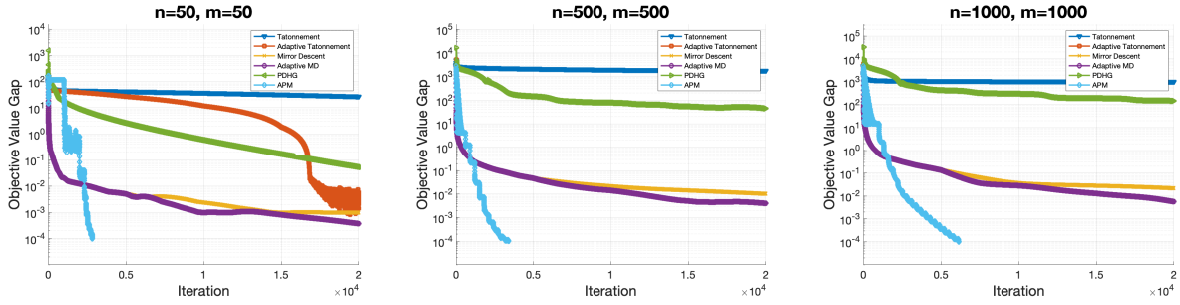


Fig. 2: Comparison of different algorithms for computing ϵ -CE prices in *linear* Fisher markets with exponential data, where the precision ϵ is 10^{-4} .

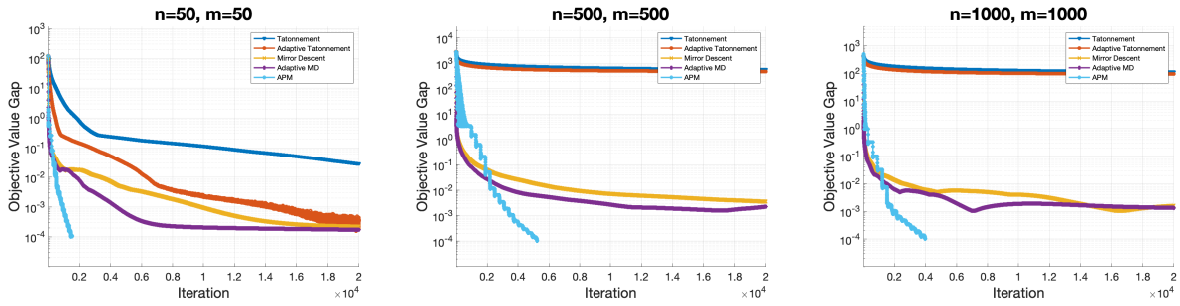


Fig. 3: Comparison of different algorithms for computing ϵ -CE prices in *quasi-linear* Fisher markets with uniform data, where the precision ϵ is 10^{-4} .

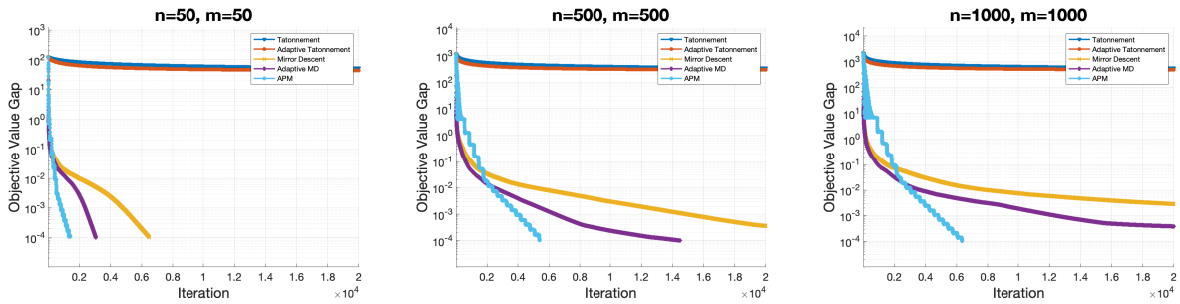


Fig. 4: Comparison of different algorithms for computing ϵ -CE prices in *quasi-linear* Fisher markets with exponential data, where the precision ϵ is 10^{-4} .

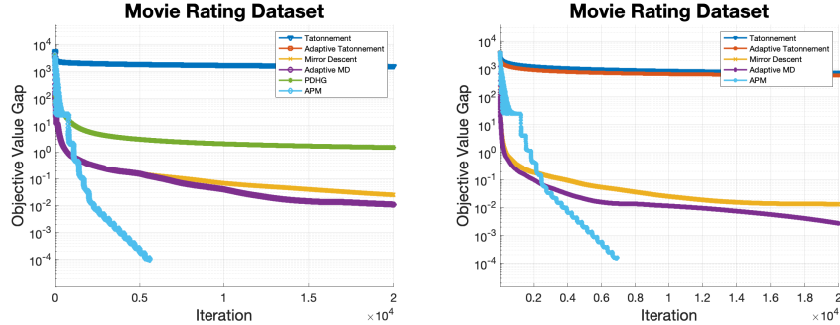


Fig. 5: Comparison of different algorithms for computing ϵ -CE prices in *linear* (left) and *quasi-linear* (right) Fisher markets with movie rating dataset, where the precision ϵ is 10^{-4} .

as a buyer. The valuation of a buyer for an item is equal to the rating he gives to the movie. The utility matrix is completed using the matrix completion software *fancyimpute*, which aligns with the data preprocessing approach of Nan et al. [39]. The resulting instance has $n = 691$ buyers and $m = 632$ items.

The numerical results are reported in Figure 1–5. These figures show that for both synthetic and real-world data, the proposed APM converges much faster than other algorithms: When APM finds approximate CE prices of desired precision, the mirror descent, additive tâtonnement, and PDHG only achieve a much lower precision. In most cases, APM only uses $\frac{1}{4}$ to $\frac{1}{5}$ of the iterations needed by other algorithms to find the desired approximate CE prices, demonstrating its efficiency. PDHG is not reported in the quasi-linear setting because the involved EG program requires CCNH conditions, which are not satisfied by quasi-linear utilities [37, Sec. 1]. We note that the objective function value gap of APM iterates oscillates in the early stage of the iterations. This can be attributed to the non-monotonicity of Nesterov’s acceleration (see, e.g., [25, Sec. 6.1.2]).

6.2 Adaptive APM for Computing Exact CE

We run the adaptive APM until the stopping criterion, i.e., the optimality conditions of Problem (\mathcal{P}), is satisfied within a tolerance of 10^{-8} . We evaluate the iteration number and CPU time used by the adaptive APM. As a comparison, we also record the CPU time of the Mosek solver to compute CE, which solves (3.1) to output a solution with a precision of 10^{-6} . Additionally, we compute the radius Δ^* based on the recovered CE prices. The experiments are conducted using both synthetic data (with 20 repeated trials) and real-world data collected from the movie rating dataset.

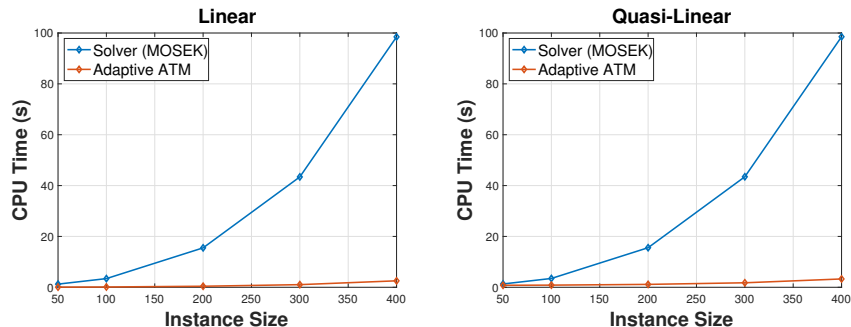


Fig. 6: Comparison of the CPU time of adaptive APM and Mosek solver for computing an exact CE with integer synthetic data.

Data	(n, m)	Utility	Iteration	Δ^*	CPU time (s)	Solver time (s)
Integer	(50,50)	Linear	2463	0.00131	0.0620	1.2455
		Quasi-linear	3788	0.00015	0.0703	1.1223
	(100,100)	Linear	2925	0.00242	0.1186	3.4148
		Quasi-linear	10219	0.00002	0.3302	3.3930
	(200,200)	Linear	10500	0.00012	0.4115	15.5087
		Quasi-linear	4670	0.00450	0.1993	16.0328
	(300,300)	Linear	3296	0.00051	1.0407	43.4084
		Quasi-linear	7600	0.00270	2.0301	41.0377
	(400,400)	Linear	9858	0.00032	2.5446	98.4671
		Quasi-linear	7239	0.00334	2.3128	102.9551
Exp	(50,50)	Linear	1238	0.00272	0.0223	1.2123
		Quasi-linear	7352	0.00005	0.1103	0.9431
	(100,100)	Linear	9937	0.00002	0.5103	3.4121
		Quasi-linear	12130	0.00003	0.2702	3.4235
	(200,200)	Linear	60523	0.00001	2.1323	15.1238
		Quasi-linear	2789	0.00023	0.1553	16.4562
	(300,300)	Linear	32391	0.00001	9.5784	43.1021
		Quasi-linear	23123	0.00003	6.0347	38.3302
	(400,400)	Linear	29353	0.00004	8.4263	102.3145
		Quasi-linear	47239	0.00003	12.5263	105.9257
Logrnd	(50,50)	Linear	2990	0.00211	0.0551	1.2353
		Quasi-linear	1278	0.00237	0.0302	1.1201
	(100,100)	Linear	3345	0.00312	0.1812	3.5642
		Quasi-linear	3648	0.00212	0.2912	3.4223
	(200,200)	Linear	21324	0.00002	0.9243	15.5890
		Quasi-linear	43679	0.00001	2.1345	16.3213
	(300,300)	Linear	44560	0.00002	14.1264	42.9084
		Quasi-linear	23905	0.00004	9.5782	41.0293
	(400,400)	Linear	67390	0.00002	16.2356	98.4131
		Quasi-linear	78924	0.00002	18.2317	102.9953
Movie Rating	(691,632)	Linear	17432	7.2×10^{-5}	2.5441	272.3047
		Quasi-linear	21328	2.4×10^{-5}	2.7332	267.1123

Table 1: Performance of the adaptive APM in computing an exact CE.

Figure 6 and Table 1 show that the adaptive APM uses approximately $\frac{1}{100}$ to $\frac{1}{10}$ running time of the Mosek solver to compute CE in most of cases. Furthermore, as the instance size grows, the CPU time of the adaptive APM increases slightly, whereas that of the Mosek solver increases significantly. These results highlight the efficiency of adaptive APM. Additionally, we observe a roughly negative correlation between the iteration number and Δ^* , consistent with our theory (i.e., Remark 5.2).

7 Conclusion and Future Directions

In this paper, we proposed a unified strongly convex formulation for computing CE of linear and quasi-linear Fisher markets. Based on this formulation, we developed new price-adjustment processes to overcome the limitations of tâtonnement methods. Specifically, by applying Nesterov’s acceleration, we developed APM that predicts the future excess-supply to adjust prices. We proved that APM finds ϵ -CE prices in $\tilde{O}(1/\sqrt{\epsilon})$ iterations, which significantly improves upon the iteration complexities of the tâtonnement methods. Furthermore, we constructed a recovery oracle that maps approximate CE prices to exact CE prices at a low computational cost. By coupling this recovery oracle with price-adjustment processes, we derived adaptive price-adjustment methods and showed that they find CE in finite steps. Finally, we conducted numerical experiments to demonstrate the fast convergence of APM and the efficient recovery of CE for adaptive APM.

Our developments suggest several directions for future research: (i) accelerating price-adjustment processes for other utilities; (ii) providing a conditional bound on the radius Δ^* for the recovery oracle; (iii) extending the recovery oracle to general market settings. Directions (i) and (iii) could contribute to the design of faster price-adjustment processes with finite-step convergence guarantees in general market settings. Direction (ii) may help develop a (conditional) polynomial-time complexity for adaptive APM to compute exact CE. We believe that exploring these directions will deepen our understanding of CE computation and advance its practical applications.

Bibliography

- [1] Kenneth J Arrow and Gerard Debreu. Existence of an equilibrium for a competitive economy. *Econometrica: Journal of the Econometric Society*, pages 265–290, 1954.
- [2] Kenneth J Arrow and Leonid Hurwicz. On the stability of the competitive equilibrium, I. *Econometrica: Journal of the Econometric Society*, pages 522–552, 1958.
- [3] Santiago Balseiro, Haihao Lu, and Vahab Mirrokni. Regularized online allocation problems: Fairness and beyond. In *International Conference on Machine Learning*, pages 630–639. PMLR, 2021.
- [4] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- [5] Benjamin Birnbaum, Nikhil R Devanur, and Lin Xiao. Distributed algorithms via gradient descent for Fisher markets. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, pages 127–136, 2011.
- [6] Anna Bogomolnaia, Hervé Moulin, Fedor Sandomirskiy, and Elena Yanovskaya. Competitive division of a mixed manna. *Econometrica*, 85(6):1847–1871, 2017.
- [7] Shant Boodaghians, Bhaskar Ray Chaudhury, and Ruta Mehta. Polynomial time algorithms to find an approximate competitive equilibrium for chores. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2285–2302. SIAM, 2022.
- [8] Stephen P Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, 2004.
- [9] William C Brainard and Herbert E Scarf. How to compute equilibrium prices in 1891. *American Journal of Economics and Sociology*, 64(1):57–83, 2005.
- [10] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.
- [11] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40:120–145, 2011.
- [12] Bhaskar Ray Chaudhury and Kurt Mehlhorn. Combinatorial algorithms for general linear Arrow-Debreu markets. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, 2018.
- [13] Bhaskar Ray Chaudhury, Christian Kroer, Ruta Mehta, and Tianlong Nan. Competitive equilibrium for chores: From dual Eisenberg-Gale to a fast, greedy, LP-based algorithm. In *Proceedings of the 25th ACM Conference on Economics and Computation*, EC '24, page 40, 2024.
- [14] He Chen, Chonghe Jiang, and Anthony Man-Cho So. Computing competitive equilibrium for chores: Linear convergence and lightweight iteration. *arXiv preprint arXiv:2410.04036*, 2024.
- [15] Li Chen, Rasmus Kyng, Yang P Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science*, pages 612–623. IEEE, 2022.
- [16] Yun Kuen Cheung. *Analyzing tâtonnement dynamics in economic markets*. PhD thesis, New York University, 2014.
- [17] Yun Kuen Cheung, Richard Cole, and Ashish Rastogi. Tâtonnement in ongoing markets of complementary goods. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 337–354, 2012.
- [18] Yun Kuen Cheung, Richard Cole, and Nikhil R Devanur. Tâtonnement beyond gross substitutes? Gradient descent to the rescue. *Games and Economic Behavior*, 123:295–326, 2020.
- [19] Richard Cole and Yixin Tao. Balancing the robustness and convergence of tâtonnement. *arXiv preprint arXiv:1908.00844*, 2019.
- [20] Richard Cole, Nikhil Devanur, Vasilis Gkatzelis, Kamal Jain, Tung Mai, Vijay V Vazirani, and Sadra Yazdanbod. Convex program duality, Fisher markets, and Nash social welfare. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 459–460, 2017.
- [21] Vincent Conitzer, Christian Kroer, Eric Sodomka, and Nicolas E Stier-Moses. Multiplicative pacing equilibria in auction markets. In *International Conference on Web and Internet Economics*, pages 142–155. Springer, 2018.

- [22] Vincent Conitzer, Christian Kroer, Debmalya Panigrahi, Okke Schrijvers, Eric Sodomka, Nicolas E Stier-Moses, and Chris Wilkens. Pacing equilibrium in first-price auction markets. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 287–288. ACM, 2019.
- [23] Vincent Conitzer, Christian Kroer, Debmalya Panigrahi, Okke Schrijvers, Nicolas E Stier-Moses, Eric Sodomka, and Christopher A Wilkens. Pacing equilibrium in first price auction markets. *Management Science*, 68(12):8515–8535, 2022.
- [24] Vincent Conitzer, Christian Kroer, Eric Sodomka, and Nicolas E Stier-Moses. Multiplicative pacing equilibria in auction markets. *Operations Research*, 70(2):963–989, 2022.
- [25] Alexandre d’Aspremont, Damien Scieur, and Adrien Taylor. Acceleration methods. *arXiv preprint arXiv:2101.09545*, 2021.
- [26] Nikhil R Devanur, Christos H Papadimitriou, Amin Saberi, and Vijay V Vazirani. Market equilibrium via a primal–dual algorithm for a convex program. *Journal of the ACM*, 55(5):1–18, 2008.
- [27] Simon Doods, Toon De Pessemier, and Luc Martens. Movietweetings: A movie rating dataset collected from twitter. In *Workshop on Crowdsourcing and Human Computation for Recommender Systems, CrowdRec at RecSys 2013*, 2013.
- [28] Ran Duan and Kurt Mehlhorn. A combinatorial polynomial algorithm for the linear Arrow–Debreu market. *Information and Computation*, 243:112–132, 2015.
- [29] Edmund Eisenberg and David Gale. Consensus of subjective probabilities: The pari-mutuel method. *The Annals of Mathematical Statistics*, 30(1):165–168, 1959.
- [30] Yuan Gao and Christian Kroer. First-order methods for large-scale market equilibrium computation. In *Advances in Neural Information Processing Systems*, volume 33, pages 20126–20138. Curran Associates, Inc., 2020.
- [31] Denizalp Goktas, Jiayi Zhao, and Amy Greenwald. Tâtonnement in homothetic Fisher markets. In *Proceedings of the 24th ACM Conference on Economics and Computation*, pages 760–781, 2023.
- [32] Andrew V Goldberg and Robert E Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM (JACM)*, 35(4):921–940, 1988.
- [33] Yicheng Huang, Wanyu Zhang, Hongpei Li, Dongdong Ge, Huikang Liu, and Yinyu Ye. Restarted primal-dual hybrid conjugate gradient method for large-scale quadratic programming. *arXiv preprint arXiv:2405.16160*, 2024.
- [34] Kamal Jain. A polynomial time algorithm for computing an Arrow–Debreu market equilibrium for linear utilities. *SIAM Journal on Computing*, 37(1):303–318, 2007.
- [35] Devansh Jalota, Marco Pavone, Qi Qi, and Yinyu Ye. Markets for efficient public good allocation with social distancing. In *International Conference on Web and Internet Economics*, pages 102–116. Springer, 2020.
- [36] Devansh Jalota, Marco Pavone, Qi Qi, and Yinyu Ye. Fisher markets with linear constraints: Equilibrium properties and efficient distributed algorithms. *Games and Economic Behavior*, 141:223–260, 2023.
- [37] Huikang Liu, Yicheng Huang, Hongpei Li, Dongdong Ge, and Yinyu Ye. PDHCG: A scalable first-order method for large-scale competitive market equilibrium computation. *arXiv preprint arXiv:2506.06258*, 2025.
- [38] Sanjay Mehrotra and Yinyu Ye. Finding an interior point in the optimal face of linear programs. *Mathematical Programming*, 62(1):497–515, 1993.
- [39] Tianlong Nan, Yuan Gao, and Christian Kroer. On the convergence of tâtonnement for linear Fisher markets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 14027–14035, 2025.
- [40] Arkadi S Nemirovskiy and Boris T Polyak. Iterative methods for solving linear ill-posed problems under precise information. *Engineering Cybernetics*, 22(4):50–56, 1984.
- [41] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103:127–152, 2005.
- [42] Yurii Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Doklady Akademii Nauk SSSR*, 269:543–547, 1983.
- [43] Yurii Nesterov. Smoothing technique and its applications in semidefinite optimization. *Mathematical Programming*, 110(2):245–259, 2007.

- [44] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2013.
- [45] James B Orlin. Max flows in $O(nm)$ time, or better. In *Proceedings of the forty-fifth Annual ACM Symposium on Theory of Computing*, pages 765–774, 2013.
- [46] Alexander Peysakhovich, Christian Kroer, and Nicolas Usunier. Implementing fairness constraints in markets using taxes and subsidies. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 916–930, 2023.
- [47] Léon Walras. *Léon Walras: Elements of Theoretical Economics: Or, the Theory of Social Wealth*. Cambridge University Press, 2014.
- [48] Fang Wu and Li Zhang. Proportional response dynamics leads to market equilibrium. In *Proceedings of the thirty-ninth Annual ACM Symposium on Theory of Computing*, pages 354–363, 2007.
- [49] Yinyu Ye. A path to the Arrow–Debreu competitive market equilibrium. *Mathematical Programming*, 111(1):315–348, 2008.

The appendix is structured as follows. In Appendices A and B, we provide the missing proofs for Sec. 4 and 5, respectively. In Appendix C, we show that the optimality test for Problem (\mathcal{P}) reduces to a max-flow problem and discuss its computational cost.

A Missing Proofs of Sec. 4

A.1 Proof of Fact 4.1

- (i)&(ii) Observe that the exponential sum $\sum_{j \in [m]} \exp(\mu_j)$ is $\exp(\bar{\mu} + \eta)$ -smooth, $\exp(\underline{\mu} - \eta)$ -strongly convex in the feasible region. Moreover, by [8, p. 74] and the composition rule, we see that the composition of linear and log-sum-exp functions, i.e., $\mu \mapsto \log(\sum_{j \in \{0\} \cup [m]} \exp(\frac{\log(v_{ij}) - \mu_j}{\delta}))$, is convex, with hessian smaller than $\frac{1}{\delta^2} \mathbf{I}_m$. It follows that the function F_δ is smooth with modulus $L = \exp(\bar{\mu} + \eta) + \|B\|_1/\delta$ and strongly convex with modulus $\sigma = \exp(\underline{\mu} - \eta)$.
- (iii) Notice that $0 \geq \sum_{j \in [m] \cup \{0\}} \lambda_{ij} \log(\lambda_{ij}) \geq -\log(m+1)$ for $\lambda_i \in \mathcal{D}_{m+1}$, $i \in [n]$. We see from (4.1) and (4.2) that $0 \leq F_\delta - F \leq \delta \log(m+1) \|B\|_1$.
- (iv) Let $\mathcal{J}_i(\mu) := \arg \max_{j \in \{0\} \cup [m]} \{\log(v_{ij}) - \mu_j\}$ for $i \in [n]$, $\mu \in \mathbb{R}^m$, and recall that $\mu_0 = \log(\alpha)$ is the parameter and $v_{i0} = 1$. Direct computation gives

$$\nabla F_\delta(\mu) \rightarrow d := (\exp(\mu_1), \dots, \exp(\mu_m)) - \sum_{i \in [n]} \frac{B_i}{|\mathcal{J}_i(\mu)|} \sum_{j \in \mathcal{J}_i(\mu)} \mathbf{e}_j \in \partial F(\mu) \quad \text{as } \delta \rightarrow 0.$$

A.2 Proof of Theorem 4.1

Denote the optimal value of Problem (\mathcal{P}^*) (resp. (\mathcal{P}_δ^*)) by F^* (resp. F_δ^*). Let $\hat{\mu}^\delta := \operatorname{argmin}_{\mu \in \mathbb{R}^m} F_\delta(\mu)$. We first prove the following lemma, which says that after the relaxation of the box constraints, the global minimizer of F_δ , i.e., $\hat{\mu}^\delta$, is located in the new box $[(\underline{\mu} - \eta)\mathbf{1}_m, (\bar{\mu} + \eta)\mathbf{1}_m]$.

Lemma A.1. *Suppose that the parameters $\delta > 0, \eta \geq 0$ satisfy $2\delta \log(m+1) \|B\|_1 \leq \exp(\underline{\mu} - \eta)\eta^2$. Then, $\hat{\mu}^\delta$ is also the optimal solution of the box-constrained problem (\mathcal{P}_δ).*

Proof. It suffices to show that $\underline{\mu} - \eta \leq \hat{\mu}_j^\delta \leq \bar{\mu} + \eta$, $j \in [m]$. Recall that $\mu^* := \operatorname{argmin}_{\mu \in \mathbb{R}^m} F(\mu)$ satisfies $\mu_j^* \in [\underline{\mu}, \bar{\mu}]$, $j \in [m]$. We only need to prove the inequality $\|\hat{\mu}^\delta - \mu^*\|_2 \leq \eta$.

Let us prove $\|\hat{\mu}^\delta - \mu^*\|_2 \leq \eta$ by contradiction. Suppose the converse that $\|\hat{\mu}^\delta - \mu^*\|_2 > \eta$. Then, there exists $s \in (0, 1)$ such that the vector $\mu^s := s\hat{\mu}^\delta + (1-s)\mu^*$ satisfies $\|\mu^s - \mu^*\|_2 = \eta$. It follows from the strong convexity of F and $\mu^* = \operatorname{argmin}_{\mu \in \mathbb{R}^m} F(\mu)$ that

$$F(\mu^s) < sF(\hat{\mu}^\delta) + (1-s)F(\mu^*) < F(\hat{\mu}^\delta). \quad (\text{A.1})$$

Further, notice that $\mu_j^s \geq \underline{\mu} - \eta$, $\mu_j^s \geq \underline{\mu}$ for $j \in [m]$, and F is $\exp(\underline{\mu} - \eta)$ -strongly convex on $[(\underline{\mu} - \eta)\mathbf{1}_m, +\infty)$. We have

$$\frac{\exp(\underline{\mu} - \eta)}{2} \|\mu^s - \mu^*\|_2^2 \leq F(\mu^s) - F(\mu^*) < F(\hat{\mu}^\delta) - F(\mu^*), \quad (\text{A.2})$$

where the second inequality is due to (A.1).

On the other hand, by Fact 4.1 (iii) and the definition $\hat{\mu}^\delta = \operatorname{argmin}_{\mu \in \mathbb{R}^m} F_\delta(\mu)$, we have

$$F(\hat{\mu}^\delta) - F(\mu^*) \leq F_\delta(\hat{\mu}^\delta) - F(\mu^*) \leq F_\delta(\mu^*) - F(\mu^*) \leq \delta \log(m+1) \|B\|_1. \quad (\text{A.3})$$

Combining (A.1) and (A.3), and recalling $\|\mu^s - \mu^*\|_2 = \eta$, we see that

$$\frac{\exp(\underline{\mu} - \eta)}{2} \eta^2 < \delta \log(m+1) \|B\|_1, \quad (\text{A.4})$$

which contradicts the condition $2\delta \log(m+1) \|B\|_1 \leq \exp(\underline{\mu} - \eta)\eta^2$. We complete the proof.

By Lemma A.1 and the first-order optimality of unconstrained problems, we know that the optimal solution of (\mathcal{P}_δ) , i.e., $\hat{\mu}^\delta$, satisfies $\nabla F_\delta(\hat{\mu}^\delta) = 0$. This, together with Nesterov [42, Theorem 2.1.10] and Nesterov [42, Theorem 2.1.5], implies the following corollary, which serves as the basis of the stopping criterion of APM.

Lemma A.2. *Suppose that the parameters δ, η in Problem (\mathcal{P}_δ) satisfy $2\delta \log(m+1)\|B\|_1 \leq \exp(\underline{\mu} - \eta)\eta^2$. Then, for all $\mu \in \mathbb{R}^m$, the following hold:*

- (i) $F_\delta(\mu) - F_\delta^* \leq \frac{1}{2\sigma} \|\nabla F_\delta(\mu)\|_2^2$;
- (ii) $F_\delta(\mu) - F_\delta^* \geq \frac{1}{2L} \|\nabla F_\delta(\mu)\|_2^2$.

Lemma A.2 ensures the validity of the stopping criterion of APM. Specifically, if the stopping criterion $\|\nabla F_\delta(\mu^t)\|_2 \leq \min\{\sigma\epsilon, \sqrt{\sigma\epsilon}\}$ holds for $t = \bar{t} \geq 0$, then Lemma A.2 (i) implies

$$F_\delta(\mu^{\bar{t}}) - F_\delta^* \leq \frac{\epsilon}{2}.$$

Observe that $F(\mu^{\bar{t}}) - F_\delta(\mu^{\bar{t}}) \leq 0$ by Fact 4.1 (iii); $F_\delta^* \leq F_\delta(\mu^*)$ since $\mu^* \in [\underline{\mu}, \bar{\mu}]^m$ is a feasible point of (\mathcal{P}_δ) while F_δ^* is its optimal value; and $F^* = F(\mu^*)$ by definition. We further have

$$\begin{aligned} F(\mu^{\bar{t}}) - F^* &= F(\mu^{\bar{t}}) - F_\delta(\mu^{\bar{t}}) + F_\delta(\mu^{\bar{t}}) - F_\delta^* + F_\delta^* - F^* \\ &\leq 0 + \frac{\epsilon}{2} + F_\delta(\mu^*) - F(\mu^*) \\ &\leq \frac{\epsilon}{2} + \delta \log(m+1)\|B\|_1 \\ &= \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon, \end{aligned} \tag{A.5}$$

where the second inequality is due to Fact 4.1 (iii) and the parameter choice $\delta = \frac{\epsilon}{2\log(m+1)\|B\|_1}$.

Given (A.5), we conclude that the stopping criterion ensures that the output $\mu^{\bar{t}}$ recovers an ϵ -CE price vector. Let us then focus on the lower bound on \bar{t} that guarantees the stopping criterion $\|\nabla F_\delta(\mu^{\bar{t}})\|_2 \leq \min\{\sigma\epsilon, \sqrt{\sigma\epsilon}\}$. By Lemma A.2 (ii), it suffices to have

$$F_\delta(\mu^{\bar{t}}) - F_\delta^* \leq \min\left\{\frac{\sigma^2\epsilon^2}{2L}, \frac{\sigma\epsilon}{2L}\right\} = \frac{\delta}{2(\delta \exp(\bar{\mu} + \eta) + \|B\|_1)} \cdot \min\{\sigma^2\epsilon^2, \sigma\epsilon\} = \mathcal{O}(\epsilon^3). \tag{A.6}$$

Notably, it still reduces to the function value gap $F_\delta - F_\delta^*$. As our APM can be viewed as Nesterov's acceleration method, i.e., Nesterov [44, Scheme 2.3.13], for the σ -strongly convex, L -smooth Problem (\mathcal{P}_δ) , the following result applies.

Lemma A.3. ([44, Theorem 2.3.6]). Let f be an L_f -smooth σ_f -strongly convex function with $L_f, \sigma_f > 0$. Let $q_f = \frac{\sigma_f}{L_f} \in (0, 1)$ and $f^* = \min_{y \in \mathcal{Y}} f(y)$, where $\mathcal{Y} \subseteq \mathbb{R}^m$ is the feasible set. The iterates of Nesterov's acceleration method, denoted by $\{y^t\}_{t \geq 0}$, satisfy

$$f(y^t) - f^* \leq (1 - \sqrt{q_f})^t (f(y^0) - f^*), \quad \forall t \geq 0.$$

Before applying Lemma A.3, we estimate the condition number $q = \sigma/L$ for the function F_δ , which is key to determining the convergence rate of APM. Let us define the constants

$$c_1 = F_\delta(\mu_0) - F_\delta^*, \quad c_2 = \frac{\exp(\underline{\mu} - \eta)}{3 \log(m+1)\|B\|_1^2}, \quad \text{and} \quad c_3 = \frac{1}{3} \exp(\underline{\mu} - \bar{\mu} - 2\eta).$$

For the estimate of q , there are two cases based on the range of ϵ . The first case is $\epsilon \leq \exp(-\bar{\mu} - \eta) \log(m+1)\|B\|_1^2$. For this case, one has

$$q = \frac{\exp(\underline{\mu} - \eta)}{\exp(\bar{\mu} + \eta) + \frac{1}{\delta}\|B\|_1} = \frac{\exp(\underline{\mu} - \eta)\epsilon}{\exp(\bar{\mu} + \eta)\epsilon + 2 \log(m+1)\|B\|_1^2} \geq \frac{\exp(\underline{\mu} - \eta)\epsilon}{3 \log(m+1)\|B\|_1^2} = c_2\epsilon. \tag{A.7}$$

For the other case that $\epsilon > \exp(-\bar{\mu} - \eta) \log(m+1) \|B\|_1^2$, the monotonic increasing property of the function $y \mapsto \frac{y}{a_1 y + a_2}$ ($a_1, a_2 > 0$) implies that

$$\begin{aligned} q &= \frac{\exp(\underline{\mu} - \eta)\epsilon}{\exp(\bar{\mu} + \eta)\epsilon + 2 \log(m+1) \|B\|_1^2} \\ &\geq \frac{\exp(\underline{\mu} - \eta) \cdot \exp(-\bar{\mu} - \eta) \log(m+1) \|B\|_1^2}{3 \log(m+1) \|B\|_1^2} \\ &= \frac{1}{3} \exp(\underline{\mu} - \bar{\mu} - 2\eta) = c_3. \end{aligned} \tag{A.8}$$

Now, applying Lemma A.3, we have

$$\begin{aligned} F_\delta(\mu^t) - F_\delta^* &\leq (1 - \sqrt{q})^t \cdot (F_\delta(\mu_0) - F_\delta^*) \\ &= c_1 \cdot \exp(t \log(1 - \sqrt{q})) \\ &\leq c_1 \cdot \exp(-\sqrt{q} \cdot t), \end{aligned} \tag{A.9}$$

where the second inequality is due to the fact that $\log(1+y) \leq y$ for all $y \in (-1, \infty)$.

For the first case that $\epsilon \leq \exp(-\bar{\mu} - \eta) \log(m+1) \|B\|_1^2$, (A.9) and (A.7) yield $F_\delta(\mu^t) - F_\delta^* \leq c_1 \cdot \exp(-\sqrt{c_2 \epsilon} \cdot t)$. It follows that the desired inequality (A.6) holds whenever

$$\bar{t} \geq \frac{-\log\left(\frac{1}{c_1} \min\left\{\frac{\sigma^2 \epsilon^2}{2L}, \frac{\sigma \epsilon}{2L}\right\}\right)}{\sqrt{c_2 \epsilon}} = \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{\epsilon}}\right). \tag{A.10}$$

For the other case that $\epsilon > \exp(-\bar{\mu} - \eta) \log(m+1) \|B\|_1^2$, (A.9) and (A.8) yield $F_\delta(\mu^t) - F_\delta^* \leq c_1 \cdot \exp(-\sqrt{c_3} \cdot t)$. In this case, the desired inequality (A.6) holds whenever

$$\bar{t} \geq \frac{-\log\left(\frac{1}{c_1} \min\left\{\frac{\sigma^2 \epsilon^2}{2L}, \frac{\sigma \epsilon}{2L}\right\}\right)}{\sqrt{c_3}} = \mathcal{O}\left(\log\left(\frac{1}{\epsilon}\right)\right).$$

Combining the above bounds on \bar{t} in two cases, we conclude that APM takes at most $\tilde{\mathcal{O}}(\frac{1}{\sqrt{\epsilon}})$ iterations to find an ϵ -CE price vector. Thus, we complete the proof of Theorem 4.1.

Remark A.1 (Parameters of APM). For some other parameter choices of APM, the above arguments (especially, (A.9), (A.5), and the estimates on q) are also valid, and hence the result of Theorem 4.1 still holds. For instance, one can set $\epsilon \in (0, +\infty)$ and $\delta = \min\{\epsilon, \exp(\underline{\mu} - \eta)\eta^2\} / (2 \log(m+1) \|B\|_1)$.

A.3 Proof of Proposition 4.1

The proof of Proposition 4.1 is similar to that of Chen et al. [14, Lemma 5.1]. To begin, let us introduce a technical lemma adapted from Chen et al. [14, Lemma C.2] that will prove useful for the proof.

Lemma A.4. *Consider $a_1, a_2, \dots, a_m > 0$ with $m > 1$ and $\gamma > 1 + \log(m-1)$. The following inequality holds:*

$$\frac{\sum_{j \in [m]} a_j^{\gamma+1}}{\sum_{j \in [m]} a_j^\gamma} \geq \max_{j \in [m]} \{a_j\} \cdot \frac{1 - \frac{1.3}{\gamma+1}}{(m-1)^{1/(\gamma+1)}}.$$

Then, we present the following observations concerning the optimal utility in (B2).

Lemma A.5. *Consider a price vector $p \in \mathbb{R}_+^m$. Let $\bar{v}_i = \max_{x_i \in \mathbb{R}_+^n} \{u_i(x_i) : \sum_{j \in [m]} p_j x_{ij} \leq B_i\}$ for $i \in [n]$. The following hold:*

- (i) *For linear utility $u_i(x_i) = v_i^\top x_i$, we have $\bar{v}_i = B_i \max_{j \in [m]} \{v_{ij}/p_j\}$.*

(ii) For quasi-linear utility $u_i(x_i) = (v_i - p)^\top x_i$, we have $\bar{v}_i = B_i \max_{j \in \{0\} \cup [m]} \{v_{ij}/p_j\} - B_i$.

Proof (Proof of Lemma A.5). (i) Under linear utilities, for all $x_i \in \mathbb{R}_+^m$ satisfying $p^\top x_i \leq B_i$, we have

$$u_i(x_i) = v_i^\top x_i = \sum_{j \in [m]} \frac{v_{ij}}{p_j} p_j x_{ij} \leq \max_{j \in [m]} \left\{ \frac{v_{ij}}{p_j} \right\} \sum_{j \in [m]} p_j x_{ij} \leq B_i \max_{j \in [m]} \left\{ \frac{v_{ij}}{p_j} \right\}.$$

Clearly, the equality can be attained by choosing $x_i = B_i e_j / p_j$ with $j \in \arg \max_{j \in [m]} \{v_{ij}/p_j\}$. We conclude that $\bar{v}_i = B_i \max_{j \in [m]} \{v_{ij}/p_j\}$.

(ii) If $0 \in \arg \max_{j \in \{0\} \cup [m]} \{v_{ij}/p_j\}$, then by $v_{i0} = p_0 = 1$, we have $\max_{\{0\} \cup [m]} \{v_{ij}/p_j\} = 1$ and further $p_j \geq v_{ij}$ for $j \in [m]$. It follows that $u_i(x_i) = (v_i - p)^\top x_i \leq 0$ for all $x_i \in \mathbb{R}_+^m$. This yields

$$\bar{v}_i = \max_{x_i \in \mathbb{R}_+^m} \left\{ u_i(x_i) : \sum_{j \in [m]} p_j x_{ij} \leq B_i \right\} = 0 = B_i \max_{j \in \{0\} \cup [m]} \left\{ \frac{v_{ij}}{p_j} \right\} - B_i.$$

If $0 \notin \arg \max_{j \in \{0\} \cup [m]} \{v_{ij}/p_j\}$, then by $v_{i0} = 1$ and $p_0 = 1$, we know that $\max_{j \in [m]} \{v_{ij}/p_j\} > 1$ and hence $p_j < v_{ij}$ for $j \in \arg \max_{j \in [m]} \{v_{ij}/p_j\}$. Then, $u_i(x_i) = (v_i - p)^\top x_i > 0$ for some $x_i \in \mathbb{R}_+^m$. We further see that

$$\max_{x_i \in \mathbb{R}_+^m} \left\{ u_i(x_i) : \sum_{j \in [m]} p_j x_{ij} \leq B_i \right\} > 0,$$

where the optimal solution \bar{x}_i satisfies

$$\bar{x}_{ij} > 0 \text{ only if } j \in \arg \max_{j \in [m]} \left\{ \frac{v_{ij} - p_j}{p_j} \right\} = \arg \max_{j \in [m]} \left\{ \frac{v_{ij}}{p_j} \right\}; \quad p^\top \bar{x}_i = B_i.$$

Let $J_i = \arg \max_{j \in [m]} \{v_{ij}/p_j\}$. We have

$$v_i^\top \bar{x}_i = \sum_{j \in J_i} \frac{v_{ij}}{p_j} \bar{x}_{ij} p_j = \sum_{j \in J_i} \bar{x}_{ij} p_j \cdot \max_{j \in [m]} \left\{ \frac{v_{ij}}{p_j} \right\} = \sum_{j \in [m]} \bar{x}_{ij} p_j \cdot \max_{j \in [m]} \left\{ \frac{v_{ij}}{p_j} \right\} = B_i \max_{j \in [m]} \left\{ \frac{v_{ij}}{p_j} \right\}.$$

Hence, we have the following estimate for \bar{v}_i :

$$\bar{v}_i = \max_{x_i \in \mathbb{R}_+^m} \left\{ u_i(x_i) : \sum_{j \in [m]} p_j x_{ij} \leq B_i \right\} = v_i^\top \bar{x}_i - p^\top \bar{x}_i = B_i \max_{j \in [m]} \left\{ \frac{v_{ij}}{p_j} \right\} - B_i.$$

Combining the above two cases, we complete the proof.

With the above lemmas in hand, we are ready to verify the conditions (B1), (B2), and (B3). First, we show (B1) holds via the definition of x :

$$p^\top x_i = B_i \frac{\sum_{j \in [m]} \exp\left(\frac{\log(v_{ij}) - \mu_j}{\delta}\right)}{\sum_{j \in \{0\} \cup [m]} \exp\left(\frac{\log(v_{ij}) - \mu_j}{\delta}\right)} \leq B_i.$$

We then prove (B3). Recall $p_j = \exp(\mu_j)$, $j \in [m]$ and let

$$d_{ij} = B_i \cdot \frac{\exp\left(\frac{\log(v_{ij}) - \mu_j}{\delta}\right)}{\sum_{j \in \{0\} \cup [m]} \exp\left(\frac{\log(v_{ij}) - \mu_j}{\delta}\right)}; \quad i \in [n], j \in [m].$$

By direct computation, we have $\nabla_j F_\delta(\mu) = p_j - \sum_{i \in [n]} d_{ij}$. This, together with the stopping criterion $\|\nabla F_\delta(\mu)\|_2 \leq \min\{\sigma\epsilon, \sqrt{\sigma\epsilon}\}$ and the fact $\sigma = \exp(\underline{\mu} - \eta) \leq \exp(\mu_j) = p_j$, yields $|\nabla_j F_\delta(\mu)| \leq p_j\epsilon$ and further

$$\epsilon \geq \left| \frac{\nabla_j F_\delta(\mu)}{p_j} \right| = \left| 1 - \sum_{i \in [n]} \frac{d_{ij}}{p_j} \right|, \quad \forall j \in [m].$$

By definition, we have $x_{ij} = d_{ij}/p_j$. It follows that $|\sum_{i \in [n]} x_{ij} - 1| \leq \epsilon$, which accords with (B3).

It is left to prove (B2). We first consider the *linear utilities*, i.e., $u_i(x_i) = v_i^\top x_i$ and $\alpha = +\infty$. By Lemma A.5 (i), to show (B2), it suffices to prove

$$u_i(x_i) = v_i^\top x_i \geq \left(1 - \frac{2\epsilon}{\|B\|_1}\right) B_i \max_{j \in [m]} \left\{ \frac{v_{ij}}{p_j} \right\}. \quad (\text{A.11})$$

Let $p_0 := \exp(\mu_0)$. We next estimate $v_i^\top x_i$:

$$v_i^\top x_i = \sum_{j \in [m]} v_{ij} \frac{d_{ij}}{p_j} = B_i \sum_{j \in [m]} \frac{v_{ij}}{p_j} \cdot \frac{\left(\frac{v_{ij}}{\exp(\mu_j)}\right)^{\frac{1}{\delta}}}{\sum_{j \in \{0\} \cup [m]} \left(\frac{v_{ij}}{\exp(\mu_j)}\right)^{\frac{1}{\delta}}} = B_i \sum_{j \in [m]} \frac{\left(\frac{v_{ij}}{p_j}\right)^{\frac{1}{\delta}+1}}{\sum_{j \in \{0\} \cup [m]} \left(\frac{v_{ij}}{p_j}\right)^{\frac{1}{\delta}}}. \quad (\text{A.12})$$

Notice $p_0 = \exp(\mu_0) = \alpha = +\infty$ for linear utilities. (A.11) automatically holds if $m = 1$. Consider $m \geq 2$ and apply Lemma A.4. We have

$$v_i^\top x_i = B_i \frac{\sum_{j \in [m]} \left(\frac{v_{ij}}{p_j}\right)^{\frac{1}{\delta}+1}}{\sum_{j \in [m]} \left(\frac{v_{ij}}{p_j}\right)^{\frac{1}{\delta}}} \geq B_i \max_{j \in [m]} \left\{ \frac{v_{ij}}{p_j} \right\} \cdot \frac{1 - \frac{1.3}{\frac{1}{\delta}+1}}{(m-1)^{1/(\frac{1}{\delta}+1)}} \geq B_i \max_{j \in [m]} \left\{ \frac{v_{ij}}{p_j} \right\} \cdot \frac{1 - 1.3\delta}{(m-1)^\delta}. \quad (\text{A.13})$$

Here, the second inequality follows from the monotonic increasing property of the function $x \mapsto (1 - 1.3x^{-1})/(m-1)^{x-1}$ on $(1.3, +\infty)$ and the inequality $1/\delta + 1 > 1/\delta = 2 \log(m+1)\|B\|_1/\epsilon > 1.3$ (recall $\epsilon \leq \log(m+1)\|B\|_1$).

Then, let us estimate $(1 - 1.3\delta)/(m-1)^\delta$. By direct computation, we have

$$\begin{aligned} (1.3 + \log(m-1))\delta &= \frac{1.3 + \log(m-1)}{2 \log(m+1)\|B\|_1} \epsilon \leq \left(\frac{1.3}{2 \log(m+1)} + \frac{\log(m-1)}{2 \log(m+1)} \right) \frac{\epsilon}{\|B\|_1} \\ &\leq \left(\frac{1.3}{2 \log(2)} + \frac{1}{2} \right) \frac{\epsilon}{\|B\|_1} \leq \frac{2\epsilon}{\|B\|_1}. \end{aligned}$$

It follows from $1 \geq \exp(-\delta \log(m-1)) \geq 1 - \delta \log(m-1)$ that

$$\begin{aligned} \frac{2\epsilon}{\|B\|_1} &\geq 1.3\delta + \delta \log(m-1) \\ &\geq 1.3\delta \cdot \exp(-\delta \log(m-1)) + 1 - \exp(-\delta \log(m-1)) \\ &= 1 - (m-1)^{-\delta} (1 - 1.3\delta). \end{aligned} \quad (\text{A.14})$$

Hence, we see that $(m-1)^{-\delta} (1 - 1.3\delta) \geq 1 - 2\epsilon/\|B\|_1$. This, together with (A.13), yields the desired (A.11). Therefore, (B2) holds for linear utilities.

We then prove (B2) under *quasi-linear utilities*, where $p_0 = \exp(\mu_0) = \alpha = 1$. We first estimate the left-hand side of (B2). Using the definition of d_{ij} , $\exp(\frac{\log(v_{ij}) - \mu_j}{\delta}) = (v_{ij}/p_j)^{1/\delta}$, and $v_{i0} = p_0 = 1$, we have

$$p^\top x_i = \sum_{j \in [m]} p_j \frac{d_{ij}}{p_j} = \sum_{j \in [m]} d_{ij} = B_i \frac{\sum_{j \in [m]} \left(\frac{v_{ij}}{p_j}\right)^{\frac{1}{\delta}}}{\sum_{j \in \{0\} \cup [m]} \left(\frac{v_{ij}}{p_j}\right)^{\frac{1}{\delta}}} = B_i - B_i \frac{1}{1 + \sum_{j \in [m]} \left(\frac{v_{ij}}{p_j}\right)^{\frac{1}{\delta}}}.$$

This, together with (A.12), yields

$$(v_i - p)^\top x_i + B_i = B_i \frac{1 + \sum_{j \in [m]} \left(\frac{v_{ij}}{p_j} \right)^{\frac{1}{\delta} + 1}}{1 + \sum_{j \in [m]} \left(\frac{v_{ij}}{p_j} \right)^{\frac{1}{\delta}}}. \quad (\text{A.15})$$

We then estimate the right-hand side of (B2). By Lemma A.5 (ii), we have

$$\max_{x'_i \in \mathbb{R}_+^m} \left\{ u_i(x'_i) + B_i : \sum_{j \in [m]} p_j x'_{ij} \leq B_i \right\} = B_i \max_{j \in \{0\} \cup [m]} \left\{ \frac{v_{ij}}{p_j} \right\} - B_i + B_i = B_i \max_{j \in \{0\} \cup [m]} \left\{ \frac{v_{ij}}{p_j} \right\}. \quad (\text{A.16})$$

Given the left-hand and right-hand side expressions of (B2), i.e., (A.15) and (A.16), the proof of (B2) reduces to that for linear utilities, especially the arguments of (A.13) and (A.14). We complete the proof.

B Missing Proofs of Sec. 5

B.1 Proof of Lemma 5.1

For each $j \in [m]$, the optimality condition (5.3) implies that at the optimal solution $\mu^* \in \mathbb{R}^m$, there exists an index $i \in [n]$ such that $\lambda_{ij} > 0$. It follows that $j \in \mathcal{J}_i(\mu^*) = J_i^*$ by the definition of λ_i . Therefore, we have

$$[m] \subseteq \bigcup_{i \in [n]} J_i^* = \bigcup_{l \in [s]} \tilde{J}_l^*,$$

where the equality is due to the definition of $\tilde{J}_l^*, l \in [s]$.

We then focus on showing the equivalence between the optimality condition (5.3) and equations (5.4), (5.5) over $l \in [s]$. We first prove that (5.4) and (5.5) can be implied by (5.3). Observe that the definitions of \mathcal{J}_i and J_i^* ensure (5.4). We focus on (5.5). Note that (5.3) implies that under the condition $0 \notin \tilde{J}_l^*$,

$$\exp(\mu_j) = \sum_{i \in [n]} \lambda_{ij}, \quad \forall j \in [m], \quad \text{where} \quad \lambda_i \in \text{conv}\{B_i e_j : j \in J_i^*\}.$$

Summing up the above equality over $j \in \tilde{J}_l^*$ (noticing $0 \notin \tilde{J}_l^*$), we have

$$\sum_{j \in \tilde{J}_l^*} \exp(\mu_j) = \sum_{i \in [n]} \sum_{j \in \tilde{J}_l^*} \lambda_{ij}. \quad (\text{B.1})$$

Recall that $\{J_i^*, i \in I_l^*\}, l \in [s]$ is a connection class and $\tilde{J}_l^* = \bigcup_{i \in I_l^*} J_i^*$. By definition, we have $J_i^* \cap \tilde{J}_l^* = \emptyset$ for $i \notin I_l$ and $J_i^* \subseteq \tilde{J}_l^*$ for $i \in I_l^*$. Since $\lambda_i \in \text{conv}\{B_i e_j : j \in J_i^*\}$, we see that $\lambda_{ij} = 0$ for $i \notin I_l^*, j \in \tilde{J}_l^*$; and $\sum_{j \in \tilde{J}_l^*} \lambda_{ij} = B_i$ for $i \in I_l^*$. These, together with (B.1), yield

$$\sum_{j \in \tilde{J}_l^*} \exp(\mu_j) = \sum_{i \in I_l^*} \sum_{j \in \tilde{J}_l^*} \lambda_{ij} = \sum_{i \in I_l^*} B_i,$$

which accords with (5.5). We conclude that (5.4) and (5.5) are implied by (5.3), and thus the solution set of (5.4) and (5.5) contains that of (5.3), i.e., $\{\mu^*\}$.

The remaining task is to show that the solution of (5.4) and (5.5) is unique. Select an arbitrary index pair (i^*, j^*) satisfying $j^* \in J_{i^*}^*$ and $i^* \in I_l^*$. By $\tilde{J}_l^* = \bigcup_{i \in I_l^*} J_i^*$ and the definition of connection class, for each $j \in \tilde{J}_l^*$, there is an index $i \in I_l^*$ such that $j \in J_i^*$; and indices $i_1, i_2, \dots, i_c \in I_l^*$ and $j_1, j_2, \dots, j_{c+1} \in \tilde{J}_l^*$, $c \in [|I_l^*|]$ such that

$$j_1 \in J_{i^*}^* \cap J_{i_1}^*; \quad j_2 \in J_{i_1}^* \cap J_{i_2}^*; \quad \dots, \quad j_c \in J_{i_{c-1}}^* \cap J_{i_c}^*; \quad j_{c+1} \in J_{i_c}^* \cap J_{i^*}^*.$$

It follows that

$$j^*, j_1 \in J_{i_1}^*; \quad j_1, j_2 \in \tilde{J}_{i_1}^*; \quad \dots \quad ; \quad j_{c-1}, j_c \in J_{i_{c-1}}^*; \quad j_c, j_{c+1} \in J_{i_c}^*; \quad j_{c+1}, j \in J_i^*.$$

This, together with (5.4), implies

$$\begin{aligned} \log(v_{i^*j^*}) - \mu_{j^*} &= \log(v_{i^*j_1}) - \mu_{j_1}; \\ \log(v_{i_1j_1}) - \mu_{j_1} &= \log(v_{i_1j_2}) - \mu_{j_2}; \quad \dots \quad ; \quad \log(v_{i_cj_c}) - \mu_{j_c} = \log(v_{i_cj_{c+1}}) - \mu_{j_{c+1}}; \\ \log(v_{i_jj_{c+1}}) - \mu_{j_{c+1}} &= \log(v_{ij}) - \mu_j. \end{aligned} \quad (\text{B.2})$$

By (B.2), we see that for every μ satisfying (5.4) and $j \in \tilde{J}_l^*$, there is a constant $a_j^{j^*}$, which can be computed through the addition and subtraction of $\log(v_{ij})$ for $i \in I_l^*$, $j \in \tilde{J}_l^*$, such that

$$\mu_j = \mu_{j^*} + a_j^{j^*}, \quad \forall j \in \tilde{J}_l^*. \quad (\text{B.3})$$

Clearly, $a_j^{j^*}$ is unique as we have shown that (5.4) and (5.5) are solvable with a solution μ^* .

Now, we are ready to prove that the solution of equations (5.4) and (5.5) is unique and exactly computable. First, consider the case that $0 \in \tilde{J}_l^*$, where the utilities must be quasi-linear and $\mu_0 = \log(\alpha) = 0$. Set $j^* = 0$. Then, the solution coordinates $\mu_j, j \in \tilde{J}_l^*$ are uniquely implied by (B.3), i.e.,

$$\mu_j = a_j^0, \quad j \in \tilde{J}_l^*. \quad (\text{B.4})$$

For the case that $0 \notin \tilde{J}_l^*$, substituting (B.3) into (5.5), we have

$$\sum_{j \in \tilde{J}_l^*} \exp(\mu_{j^*} + a_j^{j^*}) = \sum_{i \in I_l^*} B_i,$$

which yields a unique solution $\mu_{j^*} = \log(\sum_{i \in I_l^*} B_i) - \log(\sum_{j \in \tilde{J}_l^*} \exp(a_j^{j^*}))$. It follows that

$$\mu_j = \log\left(\sum_{i \in I_l^*} B_i\right) - \log\left(\sum_{j \in \tilde{J}_l^*} \exp(a_j^{j^*})\right) + a_j^{j^*}, \quad j \in \tilde{J}_l^*. \quad (\text{B.5})$$

Combining the two cases, we complete the proof.

B.2 Proof of Lemma 5.2

We prove the equality from two directions: (i) $\mathcal{J}_i(\mu^*) \subseteq \{j \in \{0\} \cup [m] : \log(v_{ij}) - \mu_j \geq h_i(\mu) - 2r\}$; (ii) $\mathcal{J}_i(\mu^*) \supseteq \{j \in \{0\} \cup [m] : \log(v_{ij}) - \mu_j \geq h_i(\mu) - 2r\}$.

Direction (i): Consider an arbitrary index $j \in \mathcal{J}_i(\mu^*)$. For $\mu \in \mathbb{B}(\mu^*, r)$, the triangle inequality yields

$$\begin{aligned} |\log(v_{ij}) - \mu_j - h_i(\mu)| &\leq |\log(v_{ij}) - \mu_j^* - h_i(\mu^*)| + |\mu_j^* - \mu_j| + |h_i(\mu^*) - h_i(\mu)| \\ &\leq 0 + \|\mu^* - \mu\|_2 + \|\mu^* - \mu\|_2 \\ &\leq 2r, \end{aligned}$$

where the second inequality is due to $j \in \mathcal{J}_i(\mu^*)$, $|\mu_j^* - \mu_j| \leq \|\mu^* - \mu\|_2$, and the 1-Lipschitz continuity of the function h_i . It follows that $\log(v_{ij}) - \mu_j \geq h_i(\mu) - 2r$. This proves Direction (i).

Direction (ii): Consider an arbitrary element j of $\{j \in \{0\} \cup [m] : \log(v_{ij}) - \mu_j \geq h_i(\mu) - 2r\}$. Notice that $\log(v_{ij}) - \mu_j \leq h_i(\mu)$ by definition of h_i . We see that $-2r \leq \log(v_{ij}) - \mu_j - h_i(\mu) \leq 0$ and hence

$$|\log(v_{ij}) - \mu_j - h_i(\mu)| \leq 2r.$$

This, together with the triangle inequality, yields

$$\begin{aligned} |\log(v_{ij}) - \mu_j^* - h_i(\mu^*)| &\leq |\log(v_{ij}) - \mu_j - h_i(\mu)| + |\mu_j - \mu_j^*| + |h_i(\mu) - h_i(\mu^*)| \\ &\leq 2r + \|\mu - \mu^*\|_2 + \|\mu - \mu^*\|_2 \\ &\leq 4r < \Delta^*, \end{aligned} \tag{B.6}$$

where the second inequality uses $|\mu_j^* - \mu_j| \leq \|\mu^* - \mu\|_2$, $\|\mu^* - \mu\|_2 \leq r$, and the 1-Lipschitz continuity of the function h_i ; the third inequality is due to the condition $r < \Delta^*/4$.

By the definition of Δ^* , i.e., (5.1), if $j \notin \mathcal{J}_i(\mu^*)$, then one has $h_i(\mu^*) - (\log(v_{ij}) - \mu_j^*) \geq \Delta^*$, which contradicts (B.6). Hence, we have $j \in \mathcal{J}_i(\mu^*)$ for j in $\{j \in \{0\} \cup [m] : \log(v_{ij}) - \mu_j \geq h_i(\mu) - 2r\}$. This implies the desired $\{j \in \{0\} \cup [m] : \log(v_{ij}) - \mu_j \geq h_i(\mu) - 2r\} \subseteq \mathcal{J}_i(\mu^*)$. The proof is complete.

B.3 Proof of Lemma 5.3

(i) For the sake of distinction, let $I_s^{\text{out}}, \tilde{J}_s^{\text{out}}, s \in [s^{\text{out}}]$ (resp. I_s, \tilde{J}_s) denote the outputs (resp. variables in process) of Algorithm 1. We use outer (resp. inner) iteration to refer to the operations from Line 3 to Line 15 (resp. Line 8 to Line 14), indexed by $s \in [s^{\text{out}}]$ (resp. $t \in [|\tilde{J}_s^{\text{out}}|]$). Notice that the updates in Line 6, 10, 12, 13 ensure $\tilde{J}_s^{\text{out}} = \bigcup_{i \in I_s^{\text{out}}} J_i$. We focus on showing that $\{J_i : i \in I_s^{\text{out}}\}, s \in [s^{\text{out}}]$ are the connection classes.

We first show that $\tilde{J}_s^{\text{out}} \cap \tilde{J}_{s'}^{\text{out}} = \emptyset$ for $s \neq s'$. Indeed, by the update of \tilde{J}_s in Line 6, 12, 13, we have

$$\tilde{J}_s^{\text{out}} = J_{i^s} \cup \left(\bigcup_{t \in [|\tilde{J}_s^{\text{out}}|]} \bigcup_{i \in \mathcal{I}_{\text{new}}(j_t^s)} J_i \right). \tag{B.7}$$

Observe that the s -th outer iteration ends with $\tilde{J}_s^{\text{out}} = J_{\text{check}} = \{j_t^s : t \in [|\tilde{J}_s^{\text{out}}|]\}$ and $j_t^s \notin J_i$ for $i \in I^c$. We see that the s -th outer iteration ends with

$$\left(\bigcup_{i \in I^c} J_i \right) \cap \tilde{J}_s^{\text{out}} = \emptyset. \tag{B.8}$$

In other words, for $s' > s$, the s' -th outer iteration begins with (B.8). Note that the set I^c at the beginning of the s' -th ($s' \geq 2$) outer iteration satisfies $I^c \supseteq I_{s'}^{\text{out}}$ and further $\bigcup_{i \in I^c} J_i \supseteq \tilde{J}_{s'}^{\text{out}}$. It follows that $\tilde{J}_{s'}^{\text{out}} \cap \tilde{J}_s^{\text{out}} = \emptyset$ for all $s' > s$. Clearly, this can be restated as $\tilde{J}_{s'}^{\text{out}} \cap \tilde{J}_s^{\text{out}} = \emptyset$ for $s' \neq s$.

We then prove that $I_s^{\text{out}} = \{i \in [n] : J_i \sim J_{i^s}\}, s \in [s^{\text{out}}]$. Note that the update of \tilde{J}_s in Line 6, 10, 12, 13 ensures that $\tilde{J}_s = \bigcup_{i \in I_s} J_i$ at Line 8. Hence, in Line 9 of the s -th outer iteration, $j_t^s \in J_{\bar{i}}$ for some $\bar{i} \in I_s$, and then $J_i \sim J_{\bar{i}}$ for all $i \in \mathcal{I}_{\text{new}}(j_t^s)$ by definition. This, together with the initialization $I_s = \{i^s\}$ and the update in Line 10, yields

$$I_s^{\text{out}} \subseteq \{i \in [n] : J_i \sim J_{i^s}\}.$$

On the other hand, recalling that $\tilde{J}_s^{\text{out}} \cap \tilde{J}_{s'}^{\text{out}} = \emptyset$ for $s \neq s'$ and $\tilde{J}_s^{\text{out}} = \bigcup_{i \in I_s^{\text{out}}} J_i$, we see that $J_i \cap J_{i'} = \emptyset$ for all $i \in I_s^{\text{out}}, i' \in I_{s'}^{\text{out}}$ with $s \neq s'$. It follows that $J_i \not\sim J_{i'}$ for $i \in I_s^{\text{out}}, i' \notin I_s^{\text{out}}$ by checking the definition. Hence, for $i^s \in I_s^{\text{out}}$, we have

$$\{i \in [n] : J_i \sim J_{i^s}\} \subseteq I_s^{\text{out}}.$$

Combined with $I_s^{\text{out}} \subseteq \{i \in [n] : J_i \sim J_{i^s}\}$, it yields $I_s^{\text{out}} = \{i \in [n] : J_i \sim J_{i^s}\}, s \in [s^{\text{out}}]$. We conclude that $\{J_i : i \in I_s^{\text{out}}\}$ forms a connection class for all $s \in [s^{\text{out}}]$.

(ii) We analyze the computational cost of the operations for each line.

Line 2, 4, 5: Clearly, the one-time cost is at most $\mathcal{O}(n)$, and they are repeated for at most $s^{\text{out}} = \mathcal{O}(n)$ times. Hence, their total computational cost is at most $\mathcal{O}(n^2)$.

Line 6: The one-time cost is at most $\mathcal{O}(m)$, it is repeated for at most $s^{\text{out}} = \mathcal{O}(n)$ times. Hence, the total computational cost is at most $\mathcal{O}(mn)$.

Line 7, 8, 13, 14: The operations include merging and subtracting the subsets of $\{0\} \cup [m]$, and checking whether $J_{check} \subsetneq \tilde{J}_s$. Hence, the one-time cost is at most $\mathcal{O}(m)$. Since the inner iteration is repeated for $|\tilde{J}_s^{\text{out}}|$ times in the s -th outer iteration, their repetition number is

$$\sum_{s \in [s^{\text{out}}]} |\tilde{J}_s^{\text{out}}| = \left| \bigcup_{s \in [s^{\text{out}}]} \tilde{J}_s^{\text{out}} \right| \leq |\{0\} \cup [m]| = m + 1, \quad (\text{B.9})$$

where the first equality is due to $\tilde{J}_s^{\text{out}} \cap \tilde{J}_{s'}^{\text{out}} = \emptyset$ for $s \neq s'$; and the second equality is due to the fact that $\tilde{J}_s^{\text{out}} \subseteq \{0\} \cup [m]$ for all $s \in [s^{\text{out}}]$.

We see that the total computational cost of Line 7, 8, 13, and 14 is at most $\mathcal{O}(m^2)$.

Line 9, 10, 11: The operations include merging and subtracting the subsets of $[n]$, and checking whether⁷ $J_i, i \in [n]$ contains j_t^s . We see that the one-time cost is at most $\mathcal{O}(n)$. The repetition number of Line 9, 10, and 11 is the same as Line 7, 8, 13, and 14, given by (B.9). Therefore, the total computational cost of Line 9, 10, and 11 is at most $\mathcal{O}(mn)$.

Line 12: The operation of Line 12 merges $|\mathcal{I}_{new}(j_t^s)|$ subsets of $\{0\} \cup [m]$ at a one-time cost of $\mathcal{O}(m \cdot |\mathcal{I}_{new}(j_t^s)|)$. We see that for the s -th outer iteration, the cost of operation of Line 12 is

$$\mathcal{O} \left(m \sum_{t \in [|\tilde{J}_s^{\text{out}}|]} |\mathcal{I}_{new}(j_t^s)| \right) = \mathcal{O} \left(m \left| \bigcup_{t \in [|\tilde{J}_s^{\text{out}}|]} \mathcal{I}_{new}(j_t^s) \right| \right) = \mathcal{O}(m |\mathcal{I}_s^{\text{out}}|). \quad (\text{B.10})$$

Here, the first equality is due to $\mathcal{I}_{new}(j_t^s) \cap \mathcal{I}_{new}(j_{t'}^s) = \emptyset$ for $j_t^s \neq j_{t'}^s$, which is ensured by Line 11 and definition of $\mathcal{I}_{new}(j_t^s)$ in Line 9; and the second equality uses the update in Line 10, which ensures that $\bigcup_{j_t^s \in \tilde{J}_s} \mathcal{I}_{new}(j_t^s) \subseteq \mathcal{I}_s^{\text{out}}$.

The total cost of the operation of Line 12 is a summation of (B.10) over $s \in [s^{\text{out}}]$, which is equal to

$$\mathcal{O} \left(m \sum_{s \in [s^{\text{out}}]} |\mathcal{I}_s^{\text{out}}| \right) = \mathcal{O} \left(m \left| \bigcup_{s \in [s^{\text{out}}]} \mathcal{I}_s^{\text{out}} \right| \right) = \mathcal{O}(mn). \quad (\text{B.11})$$

Similar to (B.9), the first equality of (B.11) is due to the fact that $\mathcal{I}_s^{\text{out}} \cap \mathcal{I}_{s'}^{\text{out}} = \emptyset$ for $s \neq s'$; the second equality uses $\mathcal{I}_s^{\text{out}} \subseteq [n]$, $s \in [s^{\text{out}}]$.

Combining the above analysis, we conclude that the computational cost of \mathcal{E} is at most $\mathcal{O}(m^2 + mn + n^2) = \mathcal{O}((m+n)^2)$.

B.4 Proof of Lemma 5.4

We derive the expression of μ^* based on the index sets output by the classification procedure:

$$\{I_l^*, \tilde{J}_l^*, i^l, j_t^l, \mathcal{I}_{new}(j_t^l) : t \in [|\tilde{J}_l^*|], l \in [s]\} = \mathcal{E}(\{J_i^*, i \in [n]\}),$$

and then show that μ^* equals the output $\mu = \mathcal{P}(\mathcal{E}(\{J_i^*, i \in [n]\}))$.

We first show $\mu_j^* = \mu_{j_1^l}^* + a_j$ for $j \in \tilde{J}_l^*$ (for any $l \in [s]$). Notice $\tilde{J}_l^* = J_{i^l}^* \cup (\bigcup_{t \in [|\tilde{J}_l^*|]} \bigcup_{i \in \mathcal{I}_{new}(j_t^l)} J_i^*)$. We prove the equality $\mu_j^* = \mu_{j_1^l}^* + a_j$ by induction.

Let us begin with μ_j^* , $j \in J_{i^l}^*$. By Line 8 of Algorithm 1, we see that $j_1^l \in J_{i^l}^*$. It follows from the definition of $J_{i^l}^*$ that $\log(v_{i^l j_1^l}) - \mu_{j_1^l}^* = \log(v_{i^l j}) - \mu_j^*$ for $j \in J_{i^l}^*$. This, together with $a_j = \log(v_{i^l j}) - \log(v_{i^l j_1^l})$ in Line 3 of Algorithm 2, yields

$$\mu_j^* = \mu_{j_1^l}^* + \log(v_{i^l j}) - \log(v_{i^l j_1^l}) = \mu_{j_1^l}^* + a_j, \quad \forall j \in J_{i^l}^*. \quad (\text{B.12})$$

⁷ One can first spend a time complexity of $\mathcal{O}(mn)$ to define a matrix $M \in \mathbb{R}^{n \times (m+1)}$, where $M_{i,j+1} = 1$ if $j \in J_i$ and $M_{i,j+1} = 0$ otherwise. Then, checking whether $J_i, i \in [n]$ contains j^* is equivalent to checking whether $M_{i,j^*+1} = 1$ for $i \in [n]$, where the cost is $\mathcal{O}(n)$ for each $j^* \in \{0\} \cup [m]$.

Now, we assume that $\mu_j^* = \mu_{j_1^l}^* + a_j$ holds for $j \in J_{done}^{t-1}$ with some $t \in [|\tilde{J}_l^*|]$, where

$$J_{done}^{t-1} := J_{i^l}^* \cup \left(\bigcup_{\tau \in [t-1]} \bigcup_{i \in \mathcal{I}_{new}(j_\tau^l)} J_i^* \right);$$

and prove that $\mu_j^* = \mu_{j_1^l}^* + a_j$ for $j \in J_{i^l}^* \cup (\bigcup_{\tau \in [t]} \bigcup_{i \in \mathcal{I}_{new}(j_\tau^l)} J_i^*)$ to finish the induction. Given the assumption, it suffices to prove that $\mu_j^* = \mu_{j_1^l}^* + a_j$ for $j \in \bigcup_{i \in \mathcal{I}_{new}(j_t^l)} J_i^* \setminus J_{done}^{t-1}$ to complete the induction.

Notice that $j_t^l \in J_i^*$ for all $i \in \mathcal{I}_{new}(j_t^l)$. We have

$$\mu_j^* = \mu_{j_t^l}^* + \log(v_{ij}) - \log(v_{ij_t^l}), \quad \forall j \in \bigcup_{i \in \mathcal{I}_{new}(j_t^l)} J_i^*, \quad i \in \mathcal{I}_{new}(j_t^l).$$

On the other hand, since $j_t^l \in J_{i^l}^* \cup (\bigcup_{\tau \in [t-1]} \bigcup_{i \in \mathcal{I}_{new}(j_\tau^l)} J_i^*)$, one has $\mu_{j_t^l}^* = \mu_{j_1^l}^* + a_{j_t^l}$ by assumption. It follows that

$$\begin{aligned} \mu_j^* &= \mu_{j_t^l}^* + \log(v_{ij}) - \log(v_{ij_t^l}) \\ &= \mu_{j_1^l}^* + a_{j_t^l} + \log(v_{ij}) - \log(v_{ij_t^l}) \\ &= \mu_{j_1^l}^* + a_j, \quad \forall j \in \bigcup_{i \in \mathcal{I}_{new}(j_t^l)} J_i^* \setminus J_{done}^{t-1}, \end{aligned}$$

where the third equality is due to the definition of a_j in Line 5 of Algorithm 2. Therefore, we complete the induction, ensuring that $\mu_j^* = \mu_{j_1^l}^* + a_j$ for all $j \in \tilde{J}_l^*$.

Next, we compute μ_j^* for $j \in \tilde{J}_l^*$. We consider two cases: (i) $0 \in \tilde{J}_l^*$; (ii) $0 \notin \tilde{J}_l^*$.

(i) For the case that $0 \in \tilde{J}_l^*$, it must be quasi-linear setting, where $\alpha = 1$. We have $0 = \log(\alpha) = \mu_0^* = \mu_{j_1^l}^* + a_0$, and hence $\mu_{j_1^l}^* = -a_0$, which accords with $\mu_{j_1^l} = -a_0$ given by Line 9 of Algorithm 2. Further, since the output μ also satisfies $\mu_j = \mu_{j_1^l} + a_j$, $j \in \tilde{J}_l^*$, we see that $\mu_j^* = \mu_j$ for $j \in \tilde{J}_l^*$.

(ii) For the case that $0 \notin \tilde{J}_l^*$, substitute $\mu_j^* = \mu_{j_1^l}^* + a_j$, $j \in \tilde{J}_l^*$ into the necessary condition (5.5). We obtain

$$\mu_{j_1^l}^* = \log \left(\sum_{i \in I_i^*} B_i \right) - \log \left(\sum_{j \in \tilde{J}_l^*} \exp(a_j) \right),$$

which accord with $\mu_{j_1^l}$ given by Line 12 of Algorithm 2. As $\mu_j = \mu_{j_1^l} + a_j$, $j \in \tilde{J}_l^*$, we have $\mu_j^* = \mu_j$ for $j \in \tilde{J}_l^*$.

Combining the two cases, we see that $\mu_j^* = \mu_j$ for all $j \in \bigcup_{l \in [s]} \tilde{J}_l^*$. Then, the desired $\mu^* = \mu$ follows from $[m] \subseteq \bigcup_{l \in [s]} \tilde{J}_l^*$ given by Lemma 5.1. We conclude that the output vector is the optimal solution of (\mathcal{P}) by Lemma 5.1, i.e., $\mathcal{P}(\mathcal{E}(\{J_i^*, i \in [n]\})) = \mu^*$.

The dominant computation cost is due to the operations of selecting $j \in J_i \setminus J_{done}$ for $i \in \mathcal{I}_{new}(j_t^l)$ in Line 5, and computing $J_{done} \cup (\bigcup_{i \in \mathcal{I}_{new}(j_t^l)} J_i)$ in Line 6. Observe that their cost is of the same order as Line 12 of Algorithm 1, which is $\mathcal{O}(mn)$ by the arguments of (B.10), (B.11) in the proof of Lemma 5.3 (ii). We conclude that the computational cost of Algorithm 2 is $\mathcal{O}(mn)$.

B.5 Proof of Remark 5.1

It suffices to show that the price vector p^* and allocation x^* satisfy (E1), (E2), and (E3) of Definition 2.1. First, the definition of x^* and the first equation of (5.2) directly yields (E1):

$$\sum_{j \in [m]} p_j^* x_{ij}^* = \sum_{j \in [m]} \lambda_{ij}^* \leq B_i,$$

For (E3), we notice $p_j^* = \exp(\mu_j^*)$ and use the second equation of (5.2) to compute

$$\sum_{i \in [n]} x_{ij}^* = \frac{\sum_{i \in [n]} \lambda_{ij}^*}{p_j^*} = \frac{\exp(\mu_j^*)}{p_j^*} = 1; \quad \forall j \in [m].$$

It is left to prove (E2). Notice that $x_{ij}^* = \lambda_{ij}^*/p_j^*$ and $p_j^* > 0$. Using $\lambda_{ij}^* = 0$ for $j \notin \mathcal{J}_i(\mu^*)$, we see that $x_{ij}^* > 0$ only if $j \in \mathcal{J}_i(\mu^*)$. By the definition of $\mathcal{J}_i(\mu^*)$, we further have

$$x_{ij}^* > 0 \quad \text{only if} \quad j \in \arg \max_{j \in \{0\} \cup [m]} \{\log(v_{ij}) - \mu_j^*\} = \arg \max_{j \in \{0\} \cup [m]} \left\{ \frac{v_{ij}}{p_j^*} \right\}.$$

Then, we consider linear and quasi-linear utilities, respectively. First, for *linear utilities*, we have

$$u_i(x_i^*) = \sum_{j \in [m]} v_{ij} x_{ij}^* = \sum_{j \in [m]} \frac{v_{ij}}{p_j^*} \lambda_{ij}^* = \sum_{j \in [m]} \lambda_{ij}^* \max_{j \in \{0\} \cup [m]} \left\{ \frac{v_{ij}}{p_j^*} \right\} = B_i \max_{j \in \{0\} \cup [m]} \left\{ \frac{v_{ij}}{p_j^*} \right\}.$$

Here, the last equality is due to the first equation of (5.2) and $0 \notin \mathcal{J}_i(\mu^*)$, $\lambda_{i0} = 0$ for linear utilities ($p_0^* = +\infty$). The above equation, together with Lemma A.5 (i), implies (E2) for linear utilities.

Next, we consider *quasi-linear utilities*.

Case (i): $0 \in \mathcal{J}_i(\mu^*) = \arg \max_{j \in \{0\} \cup [m]} \{v_{ij}/p_j^*\}$.

Recall $v_{i0} = 1$ and $p_0 = 1$. We know that (i) $v_{ij} \leq p_j^*$ for $j \in \{0\} \cup [m]$; (ii) $x_{ij}^* > 0$ and $j \in \mathcal{J}_i(\mu^*)$ only if $v_{ij} = p_j^*$. It follows that

$$u_i(x_i^*) = \sum_{j \in [m]} (v_{ij} - p_j^*) x_{ij}^* = 0 = \max_{x_i \in \mathbb{R}_+^m} \left\{ u_i(x_i) : \sum_{j \in [m]} p_j^* x_{ij} \leq B_i \right\},$$

where the last equality is due to Lemma A.5 (ii). This proves (E2) for Case (i) of quasi-linear utilities.

Case (ii): $0 \notin \mathcal{J}_i(\mu^*) = \arg \max_{j \in \{0\} \cup [m]} \{v_{ij}/p_j^*\}$.

In this case, we have $\lambda_{i0}^* = 0$ and hence $\sum_{j \in [m]} \lambda_{ij}^* = B_i$ by the first equation of (5.2). We obtain

$$u_i(x_i^*) = \sum_{j \in [m]} (v_{ij} - p_j^*) x_{ij}^* = \sum_{j \in [m]} \left(\frac{v_{ij}}{p_j^*} - 1 \right) \lambda_{ij}^*.$$

Also, by $\lambda_{ij}^* = 0$ for $j \notin \mathcal{J}_i(\mu^*)$ and $v_{ij}/p_j^* = \max_{j \in \{0\} \cup [m]} \{v_{ij}/p_j^*\}$ for $j \in \mathcal{J}_i(\mu^*)$, we further have

$$u_i(x_i^*) = \sum_{j \in [m]} \left(\frac{v_{ij}}{p_j^*} - 1 \right) \lambda_{ij}^* = \sum_{j \in [m]} \lambda_{ij}^* \max_{j \in \{0\} \cup [m]} \left\{ \frac{v_{ij}}{p_j^*} - 1 \right\} = B_i \max_{j \in \{0\} \cup [m]} \left\{ \frac{v_{ij}}{p_j^*} - 1 \right\}.$$

This, along with Lemma A.5 (ii), implies (E2) for Case (ii) of quasi-linear utilities. We complete the proof.

C Test of Optimality

In this section, we consider the test of the optimality of Problem (\mathcal{P}), which is used for the stopping criterion of the adaptive APM in Sec. 5. Due to the convexity of the objective function F , it suffices to check whether $0 \in \partial F(\mu)$, or equivalently the feasibility of (5.3) w.r.t. $\lambda \in \mathbb{R}^{n \times m}$. Further, by introducing $\lambda_{i0} := B_i - \sum_{j \in [m]} \lambda_{ij}$, the test can be further formulated as checking the feasibility of the following linear system w.r.t. $\lambda \in \mathbb{R}^{n \times (m+1)}$:

$$\begin{aligned} \sum_{j \in \{0\} \cup [m]} \lambda_{ij} &= B_i, & \forall i \in [n]; & \quad \sum_{i \in [n]} \lambda_{ij} = \exp(\mu_j), \forall j \in [m]; \\ \lambda_{ij} &\geq 0, \quad \forall j \in \{0\} \cup [m], i \in [n]; & \quad \lambda_{ij} = 0, & \quad \forall j \notin \mathcal{J}_i(\mu), i \in [n]. \end{aligned} \tag{C.1}$$

We demonstrate that testing the feasibility of (C.1) is equivalent to solving a max-flow problem. Specifically, let us define the vertex set V and the edge set E by

$$V := \{s; g_j, j \in [m]; b_i, i \in [n]; t\};$$

$$E := \{(s, g_j), j \in [m]; (g_j, b_i), j \in \mathcal{J}_i(\mu), i \in [n]; (b_i, t), i \in [n]\}.$$

Let the capacities of the edges (s, g_j) and (b_i, t) be $\exp(\mu_j)$ and B_i , respectively. Further, let the capacities of the edges (g_j, b_i) be B_i for $j \in \mathcal{J}_i(\mu)$, $i \in [n]$. For $i \in [n]$, $j \in [m]$, let q_j , p_i , and λ_{ij} denote the flows of (s, g_j) , (b_i, t) , and (g_j, b_i) , respectively. In particular, we set $\lambda_{ij} \equiv 0$ for $j \notin \mathcal{J}_i(\mu)$, which is equivalent to the fact that the edge (g_j, b_i) does not exist. Then, we obtain the following max-flow problem in optimization form, where the redundant constraints $\lambda_{ij} \leq B_i$, $i \in [n]$, $j \in \mathcal{J}_i(\mu)$ are omitted.

$$\begin{aligned} & \max_{q_i, p_j, \lambda_{ij} \geq 0} && \sum_{i \in [n]} q_j \\ \text{subject to} &&& q_j \leq \exp(\mu_j), \quad \sum_{i \in [n]} \lambda_{ij} = q_j, \quad \forall j \in [m] \\ &&& p_i \leq B_i, \quad \sum_{j \in \mathcal{J}_i(\mu)} \lambda_{ij} = p_i, \quad \forall i \in [n] \\ &&& \lambda_{ij} = 0, \quad \forall j \notin \mathcal{J}_i(\mu). \end{aligned} \tag{C.2}$$

Proposition C.1 (Max-flow Formulation for Optimality Test). *The linear system (C.1) is feasible if and only if the optimal value of the max-flow problem (C.2) equals $\sum_{j \in [m]} \exp(\mu_j)$.*

Given Proposition C.1, to test the optimality of F at $\mu \in \mathbb{R}^m$, we only need to solve the max-flow problem (C.2) and check whether its optimal value equals $\sum_{j \in [m]} \exp(\mu_j)$. Hence, the involved computational cost is bounded by the cost of solving a max-flow problem with the vertex set V and edge set E , which has been extensively studied in the literature. Efficient methods include the push-relabel algorithm [32], the algorithm of Orlin [45], and the high-probability algorithm by Chen et al. [15], which have a time complexity of $\mathcal{O}(|V|^3) = \mathcal{O}((m+n)^3)$, $\mathcal{O}(|V| \cdot |E|) = \mathcal{O}((m+n)mn)$, and $\mathcal{O}(|E|^{1+o(1)}L) = \mathcal{O}((mn)^{1+o(1)}L)$, respectively. Here, L is the bit-length of the input data (i.e., edge capacities). Note that computing an ϵ -CE ($\epsilon \leq 1$) through price-adjustment methods requires a least time complexity of $\mathcal{O}(mn \cdot \min\{m, n\})^8$. The optimality tests would not significantly increase the total computational cost of the adaptive price-adjustment methods presented in Sec. 5.

Proof (Proof of Proposition C.1). First, suppose that the linear system (C.1) is feasible. We show that the optimal value of (C.2), denoted by \bar{v} , equals $\sum_{j \in [m]} \exp(\mu_j)$. To see this, let $\lambda^* \in \mathbb{R}^{n \times (m+1)}$ be a solution of (C.1) and $\bar{\lambda} \in \mathbb{R}^{n \times m}$ be defined by $\bar{\lambda}_{ij} = \lambda_{ij}^*$ for $i \in [n]$, $j \in [m]$. Then, we have

$$\sum_{i \in [n]} \bar{\lambda}_{ij} = \exp(\mu_j); \quad \sum_{j \in \mathcal{J}_i(\mu)} \bar{\lambda}_{ij} \leq B_i; \quad \bar{\lambda}_{ij} = 0 \quad \forall j \notin \mathcal{J}_i(\mu).$$

Clearly, the pair $(\bar{q}, \bar{p}, \bar{\lambda})$ with $\bar{q}_j = \sum_{i \in [n]} \bar{\lambda}_{ij} = \exp(\mu_j)$ and $\bar{p}_i = \sum_{j \in \mathcal{J}_i(\mu)} \bar{\lambda}_{ij} \leq B_i$ is feasible for the max-flow problem (C.2). It follows a lower bound on the optimal value of (C.2):

$$\bar{v} \geq \sum_{i \in [n]} \bar{q}_j = \sum_{j \in [m]} \exp(\mu_j).$$

On the other hand, by the constraints $q_j \leq \exp(\mu_j)$, $j \in [m]$, it is direct to see $\bar{v} \leq \sum_{j \in [m]} \exp(\mu_j)$. We conclude that $\bar{v} = \sum_{i \in [n]} \bar{q}_j = \sum_{j \in [m]} \exp(\mu_j)$.

⁸ This complexity can be deduced from two facts: (i) The cost of each iteration of price-adjustment methods is $\mathcal{O}(mn)$; (ii) The iteration number needed to compute an ϵ -CE ($\epsilon \leq 1$) is at least $\mathcal{O}(\min\{m, n\})$; see, e.g., (A.10) for APM.

The remaining task is to show that the equality $\bar{v} = \sum_{j \in [m]} \exp(\mu_j)$ implies the feasibility of the linear system (C.1). Given $\bar{v} = \sum_{j \in [m]} \exp(\mu_j)$, we let the pair $(\bar{q}, \bar{p}, \bar{\lambda})$ be the optimal solution of the max-flow problem (C.2). Then, we have

$$\sum_{j \in [m]} \bar{q}_j = \bar{v} = \sum_{j \in [m]} \exp(\mu_j).$$

This, together with the constraints $\bar{q}_j \leq \exp(\mu_j)$, $j \in [m]$, yields $\bar{q}_j = \exp(\mu_j)$. Using other constraints of (C.2), we further see that $\bar{\lambda}_{ij} \geq 0$ for $i \in [n]$, $j \in [m]$ and

$$\sum_{i \in [n]} \bar{\lambda}_{ij} = \bar{q}_j = \exp(\mu_j), \quad \forall j \in [m]; \quad \sum_{j \in [m]} \bar{\lambda}_{ij} = \bar{p}_i \leq B_i, \quad \forall i \in [n]; \quad \bar{\lambda}_{ij} = 0 \quad \forall j \notin \mathcal{J}_i(\mu), i \in [n].$$

Now, define $\lambda^* \in \mathbb{R}^{n \times (m+1)}$ by $\lambda_{ij}^* = \bar{\lambda}_{ij}$ for $i \in [n]$, $j \in [m]$ and $\lambda_{i0}^* = B_i - \bar{p}_i$. Then, the above conditions on $\bar{\lambda}$ guarantee that λ^* is a feasible solution for (C.1). We complete the proof.