Competitive Equilibrium with Chores: Combinatorial Algorithm and Hardness

Bhaskar Ray Chaudhury[*]
braycha@illinois.edu

Jugal Garg[†]
jugal@illinois.edu

Peter McGlaughlin [‡]
mcglghl2@illinois.edu

Ruta Mehta[§]
rutameht@illinois.edu

May 22, 2022

Abstract

We study the computational complexity of finding a competitive equilibrium (CE) with chores when agents have linear preferences. CE is one of the most preferred mechanisms for allocating a set of items among agents. CE with equal incomes (CEEI), Fisher, and Arrow-Debreu (exchange) are the fundamental economic models to study allocation problems, where CEEI is a special case of Fisher and Fisher is a special case of exchange. When the items are goods (giving utility), the CE set is convex even in the exchange model, facilitating several combinatorial polynomial-time algorithms (starting with the seminal work of Devanur, Papadimitriou, Saberi and Vazirani [DPSV08]) for all of these models. In sharp contrast, when the items are chores (giving disutility), the CE set is known to be non-convex and disconnected even in the CEEI model. Further, no combinatorial algorithms or hardness results are known for these models.

In this paper, we give two main results for CE with chores:

- A combinatorial algorithm to compute a $(1 - \varepsilon)$-approximate CEEI in time $\tilde{\mathcal{O}}(n^4 m^2/\varepsilon^2)$, where n is the number of agents and m is the number of chores.
- PPAD-hardness of finding a $(1 - 1/\mathrm{poly}(n))$-approximate CE in the exchange model under a sufficient condition.

To the best of our knowledge, these results show the first separation between the CEEI and exchange models when agents have linear preferences, assuming PPAD $\neq$ P. Furthermore, this is also the first separation between the two economic models when the CE set is non-convex in both cases.

Finally, we show that our new insight implies a straightforward proof of the existence of an allocation that is both envy-free up to one chore (EF1) and Pareto optimal (PO) in the discrete setting when agents have factored bivalued preferences. This result is recently obtained in [GMQ22, EPS22] using an involved analysis.

1 Introduction

Competitive equilibrium (CE) theory has been one of the most fundamental concepts in mathematical economics for more than a century. Problems in this domain, study the pricing and allocation of resources to agents based on the interaction of demand and supply.

---

[*]University of Illinois at Urbana-Champaign
[†]University of Illinois at Urbana-Champaign. Supported by NSF Grant CCF-1942321
[‡]University of Illinois at Urbana-Champaign
[§]University of Illinois at Urbana-Champaign. Supported by NSF Grant CCF-1750436

The existence and computation of CE has been extensively studied in several economic models. The two most fundamental economic models are the *Arrow-Debreu (exchange)* model and the *Fisher model*. The exchange model is like a barter system, where each agent comes with some initial endowment of items and exchanges them with other agents to maximize their utility. The goal is to determine the prices for the items such that (i) each agent gets her most preferred affordable[1] bundle of items and (ii) all items are completely allocated. The Fisher model is a special case of the exchange model where every agent owns a fixed amount of each good. This has been shown to be equivalent to the setting where each agent is endowed with some fixed amount of money (instead of items). A prominent special case of the Fisher model, which is also of high interest to the fair division community, is *Competitive Equilibrium with Equal Income (CEEI)*, where each agent is endowed with equal amount of money.

The items to be divided can be either goods (giving utility) or chores (giving disutility). We assume that agents have *linear* utility functions, which are also very commonly used in applications.[2]

CE with goods. CE with goods has been well-studied in all models since the 1950s: Earlier work settled the existence of equilibrium [Gal76] and derived various convex programming and linear complementarity problem (LCP) formulations that capture the set of CE [EG59, NP83, Cor89, Eav76]. As a result, there have been continuous optimization based algorithms (interior point [Jai07], ellipsoid [Ye08]) for determining CE. Later on, faster algorithms have been obtained using combinatorial methods [DPSV08, Orl10, DM15, DGM16, GV19]. In contrast to most continuous optimization based methods, the combinatorial algorithms simulate intuitive dynamics: they start with arbitrary prices of the goods and gradually adjust them according to demand and supply. The challenge lies in showing fast (polynomial time) convergence. Moreover, most of the combinatorial algorithms are flow-based algorithms and therefore provide more room for faster algorithms, given the deep understanding of flow based subroutines since the last four decades. Indeed, the fastest weakly polynomial time and strongly polynomial time algorithms for finding CE in all models are combinatorial algorithms [Orl10, DGM16, CM18, GV19].

CE with chores. CE with chores turns out to be significantly more challenging. In contrast to the case with goods, where the CE set is convex even in the (most general) exchange model, the set of CE with chores can be non-convex containing many disconnected sets in the CEEI model [BMSY17]. This "disconnectedness" may bring up several computational bottlenecks. In fact, [BMSY17] mention: *"we expect computational difficulties in problems with many agents and/or items."* However, quite recently, [BCM22] presents a $\tilde{\mathcal{O}}(n^6 m^3/\varepsilon^2)$ time[3] exterior point algorithm for determining an $(1-\varepsilon)$-approximate CE with chores in the CEEI model where n and m denote the number of agents and chores, respectively. We remark that there are instances where the set of $(1-\varepsilon)$-approximate CE with chores in the CEEI model are also disconnected. However, the algorithm in [BCM22] does not provide any insights into the dynamics of pricing and allocation under the influence of demand and supply. Therefore, a natural question that arises is whether there are combinatorial algorithms (which are also hopefully faster) to find an approximate CE with

---

[1]Affordable in exchange of her initial endowment of items.

[2]Utility from a bundle $x = \langle x_1, x_2, \ldots, x_m \rangle$ of items is defined as: $u(x) = \sum_j u_j x_j$, where u_j is the utility from one unit of item j. Spliddit (www.spliddit.org) is a user friendly online platform for computing fair allocation in a variety of problems, which has drawn tens of thousands of visitors in the last few years [GP14]. Spliddit uses additive preferences which are the parallel to linear utilities in the context of dividing indivisible goods.

[3]$\tilde{\mathcal{O}}(\cdot)$ hides poly-logarithmic factors. The algorithm in [BCM22] runs in $\mathcal{O}(n^3/\varepsilon^2)$ iterations (Lemma 13) and each iteration solves a convex quadratic program in nm variables, which takes $\tilde{\mathcal{O}}(n^3 m^3)$ time [GL91].

chores. Unfortunately, the combinatorial algorithms for CE with goods and their corresponding convergence analysis do not generalize to the chores setting as all of them exhibit the fundamental bottleneck of *non-monotone surpluses* (elaborated in Section 2).

This brings us to the first main contribution of the paper. We overcome the barrier of *non-monotone surpluses* and give the first combinatorial algorithm for determining a $(1-\varepsilon)$-approximate CE with chores in the CEEI model. Our algorithm is significantly faster than the algorithm in [BCM22] (by a factor of $\tilde{\Omega}(n^2m)$). Furthermore, when the disutility values are α-*rounded*, i.e., the disutility incurred by any agent from consuming one unit of any chore is a power of $(1+\alpha)$ for an $\alpha > 0$, we can compute an exact CE in $\tilde{\mathcal{O}}(n^2m^2/\alpha^2)$ time. We remark that algorithm in [BCM22] does not obtain an exact CE for rounded disutilities in polynomial time!

Theorem 1. *There exist combinatorial algorithms that*

- *compute a $(1 - \varepsilon)$-approximate CEEI with chores in time $\tilde{\mathcal{O}}(n^4m^2/\varepsilon^2)$, and*

- *compute an exact CEEI with chores in time $\tilde{\mathcal{O}}(n^2m^2/\alpha^2)$ when the disutility values are α-rounded.*

We mention some additional insights that our algorithm provides. Firstly, the convergence analysis of our combinatorial algorithm also works for the setting with goods, which gives a new form of convergence analysis in that case as well. Secondly, our algorithm also shows that finding an exact CE in the Fisher model is in PLS. Since [CGMM21] shows that this problem is in PPAD, which implies that the problem lies at PPAD ∩ PLS = CLS [FGHS21].

We now address the computational complexity of finding a CE in the exchange model. Similar to the case with goods [Gal76], a CE may not always exist in the exchange model for arbitrary instances. [CGMM22] gives a polynomial-time verifiable sufficient condition[4] under which a CE is guaranteed to exist. In our second main result, we show that given an instance that satisfies the sufficient condition, determining an approximate-CE in the exchange model is PPAD-hard.

Theorem 2. *Finding a $(1-1/\text{poly}(n))$-approximate CE in the exchange model under the sufficient conditions is PPAD-hard.*

Theorems 1 and 2 show the first separation between the CEEI model and the exchange model when agents have linear preferences, assuming PPAD $\neq$ P. To the best of our knowledge, these results also give the first separation between the two economic models where the CE set is non-convex in both cases! [5]

Organization of the rest of the paper. Section 2 defines the problem with all studied models. Sections 2.1 and 2.3 present a sketch of main techniques and ideas used in designing combinatorial algorithm and showing PPAD-hardness, respectively. Section 2.2 presents a sketch of how our new insight implies the existence of an allocation that is both envy-free up to one chore (EF1) and Pareto optimal (PO) in the discrete setting under bivalued preferences. Section 3 presents applications and further related work. Appendices 4 and 5 contain the full details of our two main results.

---

[4]The sufficient condition is automatically satisfied by the Fisher model.

[5]Such a separation is also known for constant elasticity of substitution (CES) and Leontief preferences, where equilibrium computation in the CEEI model is in P using convex formulation [Eis61], and it is PPAD-hard in the exchange model [CSVY06, CPY17, GMVY17]. However, the set of CE is convex in the CEEI model and non-convex in the exchange model.

2 Model and Technical Overview

A chore division problem consists of a set $[m]$ of divisible chores (bads), and a set $[n]$ of agents, where by $[a]$ we mean set $\{1, \ldots, a\}$. We assume that each agent i has a disutility of $d_{ij} \in (0, \infty]$ for one unit of chore j.[6] $d_{ij} = \infty$ implies that chore j cannot be allocated to agent i as she may not have the required skills. If agent i is assigned a bundle $x_i = \langle x_{i1}, x_{i2}, \ldots, x_{im} \rangle$, then her total disutility $D_i(x_i) = \sum_{j \in [m]} d_{ij} x_{ij}$. We first define CE in the exchange model as it is the more general of the two models under consideration in this paper.

Exchange model. In the exchange model, each agent i brings w_{ij} units of chore j to be done (by herself or other agents). Given prices $p = \langle p_1, p_2, \ldots, p_m \rangle \in \mathbb{R}^m_{\geq 0}$ for chores, where p_j denotes the payment for doing unit amount of chore j, agent i needs to earn $\sum_{j \in [m]} w_{ij} p_j$ in order to pay to get her own chores done. Clearly, i would like to choose a *feasible bundle* that minimizes her disutility – this defines her *optimal bundle* (or optimal chore set).

$$OB_i(p) = argmin_{x_i \in \mathbb{R}^m_{\geq 0} : \langle x_i, p \rangle \geq \langle w_i, p \rangle} \quad D_i(x_i). \tag{1}$$

Due to linearity of the disutility functions, in an optimal bundle, each agent i is assigned only those chores that minimize her disutility per dollar earned and agent i earns money exactly equal to the total price of her endowments. Let MPB_i denote the minimum pain per buck of agent i at prices p, i.e., $MPB_i = min_{j \in [m]} d_{ij}/p_j$. Formally, if $x_i \in OB_i(p)$, then,

$$\forall j \in [m], \quad x_{ij} > 0 \quad \Rightarrow \quad \frac{d_{ij}}{p_j} = MPB_i, \quad \text{and} \quad \sum_{j \in [m]} x_{ij} \cdot p_j = \sum_{j \in [m]} w_{ij} \cdot p_j.$$

In the above ratios, to deal with zero prices and infinite disutilities we assume that $\infty/a > b/0$ for any $a, b \geq 0$. Clearly, an optimal bundle of an agent contains only those chores for which she has finite disutility. The price vector p is said to be at a CE if all chores are completely assigned when every agent gets one of her optimal bundles, *i.e.*, $x_i \in OB_i(p) \quad \forall i \in [n]$, and $\sum_{i \in [n]} x_{ij} = \sum_{i \in [n]} w_{ij}, \ \forall j \in [m]$. It is without loss of generality to assume that each chore is available in one unit total, i.e. $\sum_{i \in [n]} w_{ij} = 1$ (through appropriate scaling of the disutility values).

At approximate competitive equilibrium, we require that agents approximately earn enough to pay for their chores. Next we define $(1 - \epsilon)$-approximate CE (as in [BCM22]), where setting $\epsilon = 0$ gives the exact CE. Now on, by $x = (1 \pm \epsilon)y$ we mean $(1 - \epsilon)y \leq x \leq (1 + \epsilon)y$.

Definition 3. *[BCM22] We say that price vector $p = \langle p_1, p_2, \ldots, p_m \rangle \in \mathbb{R}^m_{\geq 0}$ and allocation $x = \langle x_1, x_2, \ldots, x_n \rangle$ are at $(1 - \epsilon)$-approximate competitive equilibrium (CE), if*

1. **Complete allocation.** *For each item/chore $j \in [m]$, $\sum_{i \in [n]} x_{ij} = 1 (= \sum_{i \in [n]} w_{ij})$.*

2. **Consume best chores.** *For each agent $i \in [n]$, $x_{ij} > 0 \Rightarrow \frac{d_{ij}}{p_j} = MPB_i$.*

3. **Approximate earning.** *For each agent $i \in [n]$, $\sum_{j \in [m]} x_{ij} p_j = (1 \pm \epsilon) \cdot \sum_{j \in [m]} w_{ij} p_j$.*

In general, a CE may not always exist in the exchange model (with goods or bads) [CGMM22]. In the setting with goods, there exists a polynomial time verifiable necessary and sufficient condition, under which an instance will always admit a CE. However, in the setting with chores, determining whether an arbitrary instance admits a CE is NP-complete [CGMM22], even in the CEEI model. This rules out the possibility of obtaining a polynomial time verifiable necessary and

---

[6]If $d_{ij} = 0$, we can simply assign chore j entirely to agent i and remove it from the instance.

sufficient condition unless P = NP. Therefore, the best one can hope for is a polynomial time verifiable sufficient condition that captures interesting instances. Chaudhury et al. [CGMM22] show such a sufficient condition. We now elaborate their sufficient condition. To this end, we first define the *economy graph* and the *disutility graph* of an instance I. The economy graph $E(I)$ is a directed graph with vertices corresponding to the agents $[n]$ and there is an edge from i to i' in $E(I)$ if there exists a chore j such that $w_{ij} > 0$ and $d_{i'j} \neq \infty$. The disutility graph $G(I)$ is a bipartite graph with the set of agents $[n]$ and the set of chores $[m]$ as the two parts and there is an edge from $i \in [n]$ to $j \in [m]$ if $d_{ij} \neq \infty$. [CGMM22] show the following theorem.

Theorem 4. *An instance I of chore division in the exchange model, admits a CE, if it satisfies the following two conditions:*

SC_1. *$E(I)$ is strongly connected, and*

SC_2. *$G(I)$ is a disjoint union of complete bipartite-graphs (bicliques).*

CEEI model. The CEEI model is a special case of the exchange model, where instead of the initial endowment of chores, all the agents have a fixed and the same earning requirement, say of one unit (dollar). In particular, the only change is in the definition of the feasible set of chores that can be allocated to an agent at prices p, namely $\{x_i \in \mathbb{R}^m_{\geq 0} \mid \sum_{j \in [m]} x_{ij} p_j \geq 1\}$. Under this feasibility set, $OB_i(p)$ is well-defined. Accordingly, in the Definition 3 of $(1 - \epsilon)$-approximate competitive equilibrium, the only change is in the last condition,

$$\textbf{Approximate earning. For each agent } i \in [n], \sum_{j \in [m]} x_{ij} p_j = (1 \pm \epsilon)$$

To understand why CEEI is a special case of exchange model, note that we can convert an instance of CEEI to an instance of exchange by setting all $w_{ij} = 1/n$ for each $i \in [n]$ and $j \in [m]$ and keeping the disutility values as is. Since the CE prices are scale invariant in the exchange model, wlog we can assume that they sum to n. This translates to earning requirement of 1 for each agent under this reduction.

Recall that determining whether an arbitrary instance of the CEEI model admits a CE is NP-complete [CGMM22]. It is straightforward to see that if we model an instance in the CEEI model as an instance in the exchange model, then the economy graph $E(I)$ is a clique and thus strongly connected. Therefore, the only sufficient condition that the instance needs to satisfy to admit a CE is that the disutility graph is a disjoint union of bicliques.

Corollary 5. *An instance I of chore division in the CEEI model, admits a CE, if $G(I)$ is a disjoint union of bicliques.*

Note that Corollary 5 contains all instances where all disutility values are finite. Furthermore, given any arbitrary instance I that satisfies the condition in Corollary 5, we can decompose the instances into sub-instances, where each sub-instance contains the agents and chores in a component in the disutility graph and solve each sub-instance separately. Thus, if there exists any polynomial time (approximation) algorithm for instances where all disutility values are finite, we get a polynomial time (approximation) algorithm for all instances that satisfy the sufficient condition in Corollary 5. Next, for each of our two main results, we elaborate the fundamental roadblocks and the main technical ideas to overcome them.

2.1 Combinatorial Algorithm for the CEEI Model

In this section, we highlight the issue of *non-monotone surpluses* that arises when we try to generalize combinatorial algorithms from the setting with goods to the setting with chores. Thereafter, we discuss a possible fix which will give us the crucial ideas needed for our combinatorial algorithm.

Combinatorial algorithms for goods. For the case of goods, think of d_{ij} as the utility (happiness) agent i gets per unit of *good (item)* j, and has \$1 of budget *to spend*. Clearly, at CE the agents would want to consume only the *goods* that give *maximum utility per dollar spent*, a.k.a. Max-Bang-Per-Buck (MBB). Based on this, in most of the known combinatorial algorithms, finding a CE is formulated as a flow problem as follows: Given prices of *goods* $p = \langle p_1, p_2, \ldots, p_m \rangle$ such that $\sum_{j \in [m]} p_j = n = $ total-budget, construct an MBB network $N_p = \{\{s\} \cup [n] \cup [m] \cup \{t\}, E\}$. It is a flow network comprising of a source vertex s, the set of agents $[n]$, the set of goods $[m]$ and a sink t. For each agent i, let $MBB_i = max_{j \in [m]} d_{ij}/p_j$. The edge set E of N_p comprises of

- edge (s, i) from the source s to agent i of capacity 1 (budget), for all $i \in [n]$,
- edge (i, j) with ∞ capacity, $\forall i \in [n], \forall j \in [m]$, where $d_{ij}/p_j = MBB_i$, and
- edge (j, t) from good j to sink t of capacity p_j, for all $j \in [m]$.

Observe that, determining a CE reduces to finding a set of prices p such that the maximum flow in N_p is $\sum_{j \in [m]} p_j = n$: Let f be a max-flow. Set the allocation x as $x_{ij} = f_{ij}/p_j$ for all $i \in [n]$ and $j \in [m]$. Note that f saturates all the edges from $[m]$ to t implying *complete allocation*. Each agent i is only allocated her MBB goods at prices p, and since f saturates (s, i) for all i, each agent i sends all of her budget, in turn implying that she gets her *optimal bundle*. These conditions imply a CE (see Definition 3 with $\epsilon = 0$).

Given any flow f in N_p, let *surplus* $r_f(i)$ of an agent i be the residual capacity remaining on the edge (s, i), i.e., $r_f(i) = 1 - f_{si} = 1 - \sum_{j \in [m]} f_{ij}$. If the prices are at a CE, the surplus of all agents will be zero. At a high-level, all combinatorial algorithms start with arbitrary prices and allocation and then adjust the prices and allocation such that the ℓ_2-norm of the surpluses of the agents decreases. Faster algorithms are obtained by achieving faster convergence rates with more sophisticated adjustment rules. We briefly sketch the intuitive price and flow adjustment:

Flow adjustment: During this phase, we make no changes to the prices. At a given price vector p, the flow is set to the one that minimizes the ℓ_2-norm of the surpluses of the agents.

Price adjustment: During this phase, we make no changes in the allocation. The price adjustment is done in a very intuitive way: *increase the prices of the goods* in-demand *and decrease the prices of the rest*. That is, let S be the set of agents with highest surplus, then the goods adjacent to them, say $\Gamma(S)$ in the MBB graph, are completely sold, as otherwise one can reduce surpluses of some of the agents in S and thereby reduce the ℓ_2-norm. Thus, the goods in $\Gamma(S)$ are *in-demand* as the demand exceeds supply for these goods.

We increase the prices of the goods in $\Gamma(S)$ and decrease the prices of the goods outside $\Gamma(S)$ continuously such that the sum of prices of the goods equals the sum of budgets of the agents, i.e., we scale the prices of goods in $\Gamma(S)$ by $x > 1$ and the prices of the goods in $[m] \setminus \Gamma(S)$ by $y < 1$ such that $x \cdot \sum_{j \in \Gamma(S)} p_j + y \cdot \sum_{j \in [m] \setminus \Gamma(S)} p_j = n$. See Figure 1 for an illustration. Note that, during this price change a feasible flow can be easily maintained by scaling the edge-flows appropriately. Note that the allocation still remains the same as f_{ij}/p_j remains the same.

$$p'_j = \begin{cases} xp_j & j \in \Gamma(S) \\ yp_j & j \notin \Gamma(S) \end{cases} \tag{2}$$

$$f'_{ij} = \begin{cases} xf_{ij}, & \text{if } j \in \Gamma(S) \\ yf_{ij}, & \text{if } j \notin \Gamma(S) \end{cases} \qquad f'_{jt} = \begin{cases} xf_{jt}, & \text{if } j \in \Gamma(S) \\ yf_{jt}, & \text{if } j \notin \Gamma(S) \end{cases} \qquad f'_{si} = \begin{cases} xf_{si}, & \text{if } i \in S \\ yf_{si}, & \text{if } i \notin S \end{cases} \qquad (3)$$

Note that, in this process the surplus of agents in S is decreasing. We keep updating the prices continuously until either a new MBB edge appears in the MBB graph (from some agent in S to a good outside $\Gamma(S)$) or the surplus of one of the agents in S becomes equal to the surplus of an agent outside S.
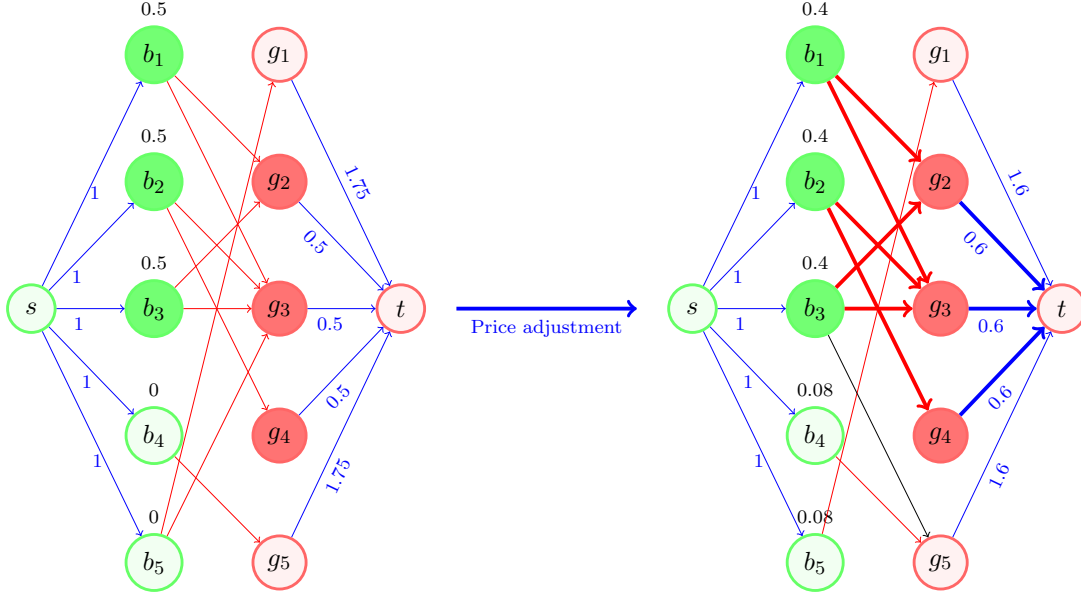

Figure 1: Illustration of the price adjustment phase. The MBB flow network N_p before the price adjustment phase is shown on the left. The numbers below the edges from s to $[n]$ and $[m]$ to t are the capacities of the edges (earning limit and prices respectively). The numbers on top of the agents show their respective surpluses w.r.t. a flow in N_p that minimizes the ℓ_2-norm of the surpluses. The set $S = \{b_1, b_2, b_3\}$ and $\Gamma(S) = \{g_2, g_3, g_4\}$. The price adjustment phase increases the prices of the goods in $\Gamma(S)$ and the inflow to $\Gamma(S)$ until either a new MBB edge appears in N_p or one of the surpluses of an agent in S equals that of some agent outside S. The figure on the right shows the MBB network after the price adjustment phase – a new MBB edge from b_3 to g_5 appears and the ℓ_1-norm of the surpluses decreases.

Correctness. We need to first argue that f' is a *valid-flow* in $N_{p'}$. Note that if f' augments flow on valid edges in $N_{p'}$, it is easy to see that it satisfies flow conservation and capacity constraints. So we focus only on showing that $f'_{ij} > 0$ if and only if $(i, j) \in N_{p'}$. Note that, since flow on edges are scaled multiplicatively, $f'_{ij} > 0$ if and only if $f_{ij} > 0$. Therefore, it suffices to prove that no edge with positive flow under f can disappear during the price update. Since the prices of goods in $\Gamma(S)$ are increased by factor x and those in $[m] \setminus \Gamma(S)$ are decreased by factor y, only the edges to $\Gamma(S)$ can disappear. However, all agents in S have MBB edges only to goods in $\Gamma(S)$ and since the price of all these goods increases by the same factor, none of the edges from S to $\Gamma(S)$ disappear. The only edges that may disappear are the ones from agents outside S to goods in $\Gamma(S)$. We next argue that these edges must have zero flow.

To this end, note that f was the flow that minimized the ℓ_2-norm of the surpluses of the agents in N_p. Suppose $i' \notin S$ has a positive flow to some good $j \in \Gamma(S)$, i.e., $f_{i'j} \geq \varepsilon > 0$. Since $j \in \Gamma(S)$, there must be an $i \in S$ with an MBB edge to j. Since S is the set of agents with highest surplus,

we can conclude that $r_f(i) > r_f(i')$. Then, we can push more flow from i to j and push back the same amount of flow from j to i' continuously. Note that in this process the ℓ_1-norm of the surplus remains the same while the ℓ_2-norm decreases (as two surpluses $r_f(i)$ and $r_f(i')$ are moving closer to each other), which is a contradiction. Thus $f_{i'j} = 0$.

Now, we are ready to argue why the above procedure should converge. The price and flow update influences the surpluses of the agents. We show that the ℓ_1-norm of the surplus monotonically decreases. Note that, after a price adjustment phase,

$$\sum_{i \in [n]} r_{f'}(i) = \sum_{i \in [n]} (1 - \sum_{j \in [m]} f'_{ij}) = \sum_{j \in [m]} (p'_j - \sum_{i \in [n]} f'_{ij}) \qquad (\sum_{j \in [m]} p'_j = n)$$

$$= \sum_{j \in \Gamma(S)} (p'_j - \sum_{i \in [n]} f'_{ij}) + \sum_{j \notin \Gamma(S)} (p'_j - \sum_{i \in [n]} f'_{ij}) = x \sum_{j \in \Gamma(S)} (p_j - \sum_{i \in [n]} f_{ij}) + y \sum_{j \notin \Gamma(S)} (p_j - \sum_{i \in [n]} f_{ij})$$

$$= y \sum_{j \notin \Gamma(S)} (p_j - \sum_{i \in [n]} f_{ij}) \qquad \text{(goods in } \Gamma(S) \text{ are fully sold)}$$

$$< \sum_{j \in [m]} (p_j - \sum_{i \in [n]} f_{ij}) = \sum_{i \in [n]} (1 - \sum_{j \in [m]} f_{ij}) = \sum_{i \in [n]} r_f(i). \qquad (y < 1, \sum_{j \in [m]} p_j = n)$$

Furthermore, every flow that minimizes the ℓ_2-norm of the surpluses is also a maximum flow [DPSV08]. Thus, after the flow update phase, the ℓ_1-norm of the surpluses does not increase, implying that after each iteration the ℓ_1-norm of the surpluses of the agents decreases. To get faster convergence, many algorithms show that the ℓ_2-norm of the surpluses decreases significantly. Intuitively it is equivalent to arguing that surpluses of the agents are getting more balanced, and this is straight-forward to see. Observe that

- the ℓ_1-norm decreases,
- the surplus of the agents in S (the high surplus agents) decreases, and
- the surplus of the agents in $[n] \setminus S$ (the low surplus agents) increases.

A proper analysis can show that the ℓ_2-norm decreases as well. The decrease in the ℓ_2-norm is used to prove polynomial convergence in the exchange model too [DM15, DGM16, CM18].

Non-monotone surpluses for CE with chores. We now elaborate the fundamental problem in generalizing the overall structure of the above approach to the setting with *chores*.

At CE, since every agent consumes only her Min-Pain-per-Buck (MPB) chores, we can define the MPB-network analogously to the MBB-network for goods. That is, network N_p at prices p where only difference is that edge (i, j) is present if $\frac{d_{ij}}{p_j} = min_k \frac{d_{ik}}{p_k} = MPB_i$. Like earlier, then the goal is to find a set of prices p at which the maximum flow in N_p is $n = \sum_{j \in [m]} p_j$. Given a flow f in N_p, we define the surplus of an agent $r_f(i) = 1 - f_{si} = 1 - \sum_{j \in [m]} f_{ij}$. Now, we wish to come up with an allocation and price adjustment rules that help decrease the ℓ_1 or ℓ_2-norm of the surpluses. At a fixed price, we always set the allocation to the one that minimizes the ℓ_2-norm of the surpluses. Now, we show the crucial problem during a price adjustment phase. Let S be the set of agents with highest surplus and $\Gamma(S)$ be the set of chores that are their neighbors in N_p. Note that, in case of chores, agents prefer *higher prices* since they fetch better earning.

- **Increasing prices of chores in $\Gamma(S)$ and decreasing prices of chores outside $\Gamma(S)$:** Increasing prices of chores $\Gamma(S)$ (*in-demand* chores) is counter-intuitive, since it will further

8

increase their demand. Formally, after the flow-adjustment step, each agent $i \notin S$ has flow only to the chores outside of $\Gamma(S)$, but may have an MPB edge to a $j \in \Gamma(S)$. Now if prices of chores in $\Gamma(S)$ are increased, and of those outside $\Gamma(S)$ are decreased, then i would strictly prefer j over any chore outside of $\Gamma(S)$. Hence all of her flow-carrying edges will disappear, and to maintain a valid flow we will have to reduce flow on (s, i) edge accordingly which will significantly increase the surplus of agent i and the total surplus.

- **Decreasing prices of chores in $\Gamma(S)$ and increasing prices of chores outside $\Gamma(S)$:** While this definitely seems more intuitive, the ℓ_1 and the ℓ_2-norm of the surpluses can still increase. So we update the prices and flows as in equations (2) to (3) with $x < 1$ and $y > 1$ this time. Then using the same derivation as in the goods case and the fact that $y > 1$ we get that $\sum_{i \in [n]} r_{f'}(i) > \sum_{i \in [n]} r_f(i)$. That is, the ℓ_1-norm of the surpluses increases (diverges).

A possible workaround for the second bullet above is to allow negative surpluses (or equivalently relaxing the capacity constraints on the flow on the edges from s to $[n]$). This will always ensure that f will saturate all edges from t to $[m]$ or equivalently f has total flow-value of $\sum_{j \in [m]} p_j$. This way, we have sum of the surpluses is always zero. However, our goal now becomes to get the ℓ_2-norm of the surpluses to be zero (as this will ensure that each agent i earns exactly 1 through her MPB chores). But observe that the ℓ_2-norm may increase: Since we are decreasing the outflow from agents in S (who are the high-surplus agents), the surplus of each agent in S increases. Similarly, since we are increasing the outflow from each agent outside S (the low surplus agents), the surplus of the agents outside S decreases. As such, the gap between the high surplus and the low surplus increases, and in turn the ℓ_2-norm increases. In fact, this is the main bottleneck in using any potential that "balances" the surpluses.

Overcoming non-monotone surpluses: Nash-welfare. We show how to overcome the problem of non-monotone surpluses. Decreasing prices of the high-demand chores and increasing that of the low-demand chores is a natural price adjustment scheme. Our solution lies in identifying the correct convergence analysis.

Recall that when we decrease the prices of the high-demand chores and increase the price of the low demand chores, the ℓ_2-norm of the surplus increases as this process increases the gap between the surpluses of the agents in S and the ones outside S while keeping sum of surpluses the same. However, such a price adjustment can also bring new MPB edges (say (i, j)) in the graph from agents in S to chores outside of $\Gamma(S)$. Let i' be an agent outside S that has positive outflow to j. Note that at this point we can definitely balance the surplus by pushing some flow from i to j and pushing back some flow from j to i'. What is unclear is whether we can balance the flow sufficient enough to compensate for the increase in ℓ_2-norm prior to the appearance of the MPB edge! This observation leads us to our new potential function.

We want a potential function which does not change during the price adjustment, but improves during the allocation update with the help of the new MPB edge.

We define our potential function as the product of the disutilities: $\prod_{i \in [n]} D_i(x_i)$. Crucially observe that the allocation $x_{ij} = f_{ij}/p_j$ does not change during the price adjustment as f_{ij} and p_j are scaled by the same scalar. Therefore, our potential does not change during price update. We now make subtle changes in the price and the allocation adjustment process and then argue about the convergence of our algorithm.

Given a set of prices p, let N_p be the MPB flow network. We make a subtle change in the capacities of the edges. We make the capacity of all edges from s to $[n]$ infinite, implying that

any max-flow will push a total flow of $\sum_{j\in[m]} p_j$ saturating all edges from $[m]$ to t. Given a flow f in N_p, let $outflow_f(i)$ denote the total outflow from agent i, i.e., $outflow_f(i) = \sum_{j\in[m]} f_{ij}$. Our goal now is to determine prices p and a max flow such that $outflow_f(i) = 1$ for all $i \in [n]$ and $\sum_{j\in[m]} p_j = n$. Note that we can write $D_i(x_i) = MPB_i \cdot outflow_f(i)$. Therefore, we can re-write our potential $\prod_{i\in[n]} D_i(x_i) = \prod_{i\in[n]}(MPB_i \cdot outflow_f(i)) = \prod_{i\in[n]} MPB_i \cdot \prod_{i\in[n]} outflow_f(i)$.

Let S be the set of agents with smallest value of $outflow_f(i)$ and let $\Gamma(S)$ be the set of chores adjacent to S in N_p. We adjust the allocation and prices as follows,

Flow adjustment. We do not change prices and set flow to the one that maximizes $\prod_{i\in[n]} outflow_f(i)$.

Price adjustment. We decrease the prices of the chores in $\Gamma(S)$ by factor of $x < 1$ until a new MPB edge appears from some agent in S to a chore outside of $\Gamma(S)$. And we keep a valid flow.

$$p'_j = \begin{cases} xp_j & j \in \Gamma(S) \\ p_j & j \notin \Gamma(S) \end{cases} \tag{4}$$

$$f'_{ij} = \begin{cases} xf_{ij} & j \in \Gamma(S) \\ f_{ij} & j \notin \Gamma(S) \end{cases} \quad f'_{jt} = \begin{cases} xf_{jt} & j \in \Gamma(S) \\ f_{jt} & j \notin \Gamma(S) \end{cases} \quad f'_{si} = \begin{cases} xf_{si} & i \in S \\ f_{si} & i \notin S \end{cases} \tag{5}$$

Correctness. Crucially, we can still argue that f' is a valid flow in $N_{p'}$ like earlier.[7] To show convergence, we show that our potential improves after every price and flow adjustment. First note that the potential does not change during the price adjustment: Let x and x' be the allocations corresponding to the flows f and f' respectively. Note that $x_{ij} = f_{ij}/p_j$ and $x'_{ij} = f'_{ij}/p'_j$ for all $i \in [n]$, $j \in [m]$. From equations (4) and (5), we have $f'_{ij}/p'_j = f_{ij}/p_j$ for all $i \in [n]$, $j \in [m]$, implying that $x = x'$. Therefore $D_i(x'_i) = D_i(x_i)$ and the potential does not change during price adjustment.

Now we show that the potential improves during flow adjustment. Note that, since our potential is $\prod_{i\in[n]} MPB_i \cdot \prod_{i\in[n]} outflow_{f'}(i)$, where $\prod_{i\in[n]} MPB_i$ does not change during the flow adjustment, it suffices to show that there is a flow f'' in $N_{p'}$ such that $\prod_{i\in[n]} outflow_{f''}(i) > \prod_{i\in[n]} outflow_{f'}(i)$. Let (i, j) with $i \in S$ and $j \notin \Gamma(S)$ be the new MPB edge that appears in $N_{p'}$ when we update the flow from f to f'. Let $i' \notin S$ be an agent that had positive outflow to j. Note that $outflow_f(i) < outflow_f(i')$ and the flow and price adjustments outlined in equations (4) and (5) show that the outflow of agent i decreases and that of i' remains the same. This implies that $outflow_{f'}(i) < outflow_f(i) < outflow_f(i') = outflow_{f'}(i')$. Also, $f'_{i'j} > 0$ as $f_{i'j} > 0$. Let f'' be the flow obtained from f' by pushing more flow from i to j and pushing back the same amount of flow from j to i', while still maintaining $outflow_{f''}(i) \leq outflow_{f''}(i')$. Note that $outflow_{f'}(i) < outflow_{f''}(i) \leq outflow_{f''}(i') < outflow_{f'}(i')$ and $outflow_{f''}(i) + outflow_{f''}(i') = outflow_{f'}(i) + outflow_{f'}(i')$. Therefore, we have $\prod_{i\in[n]} outflow_{f''}(i) > \prod_{i\in[n]} outflow_{f'}(i)$ (as the outflows are getting more balanced).

---

[7]The most crucial argument is to show that all positive flow carrying edges do not disappear during the price adjustment, i.e., all (i, j) such that $f_{ij} > 0$ are present in $N_{p'}$ (equations of (5) would then automatically imply that $f'_{ij} > 0$ only if $(i, j) \in N_{p'}$). Since we only decrease the prices of the chores in $\Gamma(S)$, the only edges that can disappear are the ones that are incoming to $\Gamma(S)$ from agents outside S. Post flow-adjustment step, these edges will carry zero flow. Otherwise we have a chore $j \in \Gamma(S)$, an agent $i \in S$ and an agent $i' \notin S$ such that $f_{ij} > \varepsilon > 0$ and $f_{i'j} > \varepsilon > 0$. Note that we can push more flow from i to j and push back some flow from j to i' such that $outflow_f(i)$ and $outflow_f(i')$ move closer to each other while keeping the sum of outflows the same, i.e., $\sum_{i\in[n]} outflow_f(i)$ remains same. This improves $\prod_{i\in[n]} outflow_f(i)$, which is a contradiction. Thereafter, it is straightforward to see that the f' satisfies flow conservation and capacity constraints in $N_{p'}$.

We refer the reader to Section 4 for a more detailed and rigorous explanation of the combinatorial algorithm. The remaining bulk of the effort is put into proving fast (polynomial time) convergence when either (i) the disutility values are α-rounded or (ii) we are only aiming for $(1-\varepsilon)$-CEEI. Lastly, note that our algorithm moves from a Nash welfare maximizing allocation at a particular MPB configuration to a Nash welfare maximizing allocation at another MPB configuration. In this process, it also strictly improves the Nash welfare. This shows that finding a CEEI (exactly) is in PLS. Since [CGMM21] show that this problem is in PPAD, we can conclude that this problem lies in PPAD $\cap$ PLS = CLS.

2.2 Discrete Setting with Bivalued Preferences

In the discrete chore division problem, a set of *indivisible* chores needs to be divided among agents in a fair and efficient manner. In this setting, envy-freeness up to one chore (EF1) and Pareto optimality (PO) are popular fairness and efficiency notions, respectively.

EF1 and PO. An allocation x is said to be EF1 if for every pair of agents (i, i'), there exists a chore $c(i, i') \in x_i$ such that $D_i(x_i \setminus c(i, i')) \leq D_i(x_{i'})$, i.e., the disutility of agent i from x_i after the removal of a chore $c(i, i')$ is at most the disutility of i for $x_{i'}$.[8]

An allocation x is said to Pareto dominates another allocation x' if $D_i(x_i) \leq D_i(x'_i), \forall i$ with a strict inequality for at least one agent. We say that x is PO if no other allocation x' dominates it.

A major open question in the discrete setting is whether there always exists an allocation that is both EF1 and PO. Very recently, [GMQ22, EPS22] answered this question affirmatively when agents have bivalued preferences, i.e., $d_{ij} \in \{1, \beta\}$, for some $\beta > 1$. However, their analysis is quite involved, especially due to lack of a potential function for the convergence. Next, we briefly sketch how our new insight on the potential function of the product of disutility values implies a straightforward argument for the convergence.

First, we briefly sketch the approaches of [GMQ22, EPS22]. They both utilize the notions of *integral* competitive equilibrium (CE) and *price envy-freeness up to one chore* (pEF1) [BKV18]. Since pEF1 implies both EF1 and PO, we only need to find an allocation that is pEF1. For that, they start with a trivial CE (x, p) and achieve pEF1 by *directly* transferring chores (on MPB edges) from a *big earner* to a *least earner*.[9] The main challenge in both the papers is to show that this process converges. Let $p(x_i) = \sum_{j \in x_i} p_j$ is the total payment of doing chores in x_i. Big earner, say b, is an agent who earns the maximum after the removal of a chore, i.e., $b = \arg\max_i \min_{c \in x_i} p(x_i \setminus c)$. Similarly, least earner, say l, is an agent who earns the least, i.e., $l = \arg\min_i p(x_i)$. If $p(x_b) - p_c \leq p(x_l)$, where c is a chore in x_b, then x is a pEF1 allocation.

Using our new insight on the potential function of the product of disutilities, we next show that it strictly increases in the above setting as well, thereby giving a straightforward proof of the convergence. Since any form of price-update doesn't change the potential function (because the allocation remains the same), we only need to show that it strictly increases during an allocation update. Further, the above discussion implies that we only need to show that if $p(x_b) - p_c > p(x_l)$ then transferring c from b to l results in an increased potential function. Let x' be the new allocation where $x'_b = x_b \setminus c$ and $x'_l = x_l \cup c$. Recall from Section 2.1 that we can write $D_i(x_i) = \prod_i MPB_i \cdot \prod_i p(x_i)$. Since MPB_i's doesn't change during allocation change, we need to show that

$$\prod_i p(x_i) < \prod_i p(x'_i),$$

---

[8]In the discrete setting, we assume x to be integral, i.e., $x_{ij} \in \{0, 1\}, \forall (i, j)$.

[9]We note that this direct transfer is always possible in the case of bivalued preferences, which may not be the case for general additive preferences, and hence the problem for general additive preferences is still open.

which is

$$p(x_b)p(x_l) < p(x_b')p(x_l').$$

Expanding it, we get

$$p(x_b)p(x_l) < (p(x_b) - p_c)(p(x_l) + p_c).$$

Simplifying it, we get

$$p_c < p(x_b) - p(x_l),$$

which is clearly true if x is not an pEF1 allocation. Since bivalued preferences are already $(\beta - 1)$-rounded utilities, our running time analysis in Appendix 4 can be adapted to show convergence polynomial in n, m and $1/(\beta - 1)$.

2.3 PPAD-Hardness for the Exchange Model

In this section, we sketch the proof of Theorem 2, i.e., computing a $(1 - 1/poly(n))$-approximate CE of an exchange instance is PPAD-hard, even if the instance satisfies sufficiency conditions (SC_1 and SC_2) of Theorem 4 (see Section 5 for the formal and detailed proof). We give a polynomial time reduction from the problem of computing a *Nash equilibrium* of a *normalized polymatrix game*, a known PPAD-hard problem [CPY17].

A polymatrix game is represented by a game graph where each node is a player who plays a two-player game with each of her neighbors. She has to play the same strategy with each of her neighbors and her payoff is the sum of the payoffs on each of her incident edges. If there are n players and each of them has exactly two strategies to choose from, then such a game can be represented by $2n \times 2n$ matrix. When thought of it as $n \times n$ block matrix, where each block is 2×2 matrix, then $(i, j)^{th}$ block is the payoff matrix of player j for the game on edge (i, j).

Problem 6. *(Normalized Polymatrix Game) [CPY17]*
 Given: *A $2n \times 2n$ rational matrix $\mathbf{M}$ with every entry in $[0, 1]$ and $\mathbf{M}_{i,2j-1} + \mathbf{M}_{i,2j} = 1$ for all $i \in [2n]$ and $j \in [n]$.*
Find: *$1/n$-approximate Nash equilibrium: Strategy vector $x \in \mathbb{R}_{\geq 0}^{2n}$ such that $x_{2i-1} + x_{2i} = 1$ and*

$$x^T \cdot \mathbf{M}_{*,2i-1} > x^T \cdot \mathbf{M}_{*,2i} + \tfrac{1}{n} \implies x_{2i} = 0.$$
$$x^T \cdot \mathbf{M}_{*,2i} > x^T \cdot \mathbf{M}_{*,2i-1} + \tfrac{1}{n} \implies x_{2i-1} = 0.$$

where $M_{*,k}$ represents the kth column of the matrix $\mathbf{M}$.

Given an instance $I = \langle \mathbf{M} \rangle$ of the polymatrix game, we create an instance of chore division $E(I) = \langle A \cup B, d(\cdot, \cdot), w(\cdot, \cdot) \rangle$, such that given any CE in $E(I)$, we can determine in polynomial time an equilibrium strategy vector x for I. Since we will have to create many different *types* of agents and chores, for ease of notation, we use $d(i, j), w(i, j)$ and $p(j)$ instead of d_{ij}, w_{ij} and p_j respectively in this section and in Section 5.

Sketch of the Reduction. The entries of matrix M are encoded into endowments $w(\cdot, \cdot)$ of agents, and the equilibrium strategy vector x of the polymatrix game I can be *extracted* from the price vector at a CE of $E(I)$. The key properties that our hard instance $E(I)$ exhibits are *pairwise equal endowments, fixed earning, price equality, price regulation* and *reverse ratio amplification* (we will give a precise definition of these properties shortly). These techniques (constructing hard instances exhibiting these properties) have been used earlier to prove PPAD-hardness for the exchange model with only goods when agents have constant elasticity of substitution (CES) utilities [CPY17] and even for the Fisher model when agents have separable piecewise linear concave

(SPLC) utilities [CT09]. However, the challenge is to construct these gadgets, in particular the *reverse ratio amplification gadget* (a brief description will be given towards the end of this section) and make them work together only using linear disutility functions; as clearly this is not possible in case of goods, when agents have linear utility functions.

We now describe our instance $E(I)$. To encode the entries of the matrix $\mathbf{M}$ of the poly-matrix game, we introduce two sets of $2n$ chores in $E(I)$, namely $B = \{b_1, b_2, \ldots, b_{2n}\}$ and $B' = \{b'_1, b'_2, \ldots, b'_{2n}\}$. For each $i, j \in [2n]$, we add an agent a_{ij} who brings $\mathbf{M}_{i,j}$ units of chore b'_i. The disutility values of these agents are as follows:

$$\forall i \in [2n], j \in [n], \ d(a_{i,2j-1}, b_{2j-1}) = (1-\alpha) \quad \text{and} \quad d(a_{i,2j-1}, b_{2j}) = (1+\alpha)$$
$$d(a_{i,2j}, b_{2j-1}) = (1+\alpha) \quad \text{and} \quad d(a_{i,2j}, b_{2j}) = (1-\alpha),$$

for some infinitesimally small $\alpha > 0$. The disutility values that are not specified are all infinity. Observe that the agents $a_{i,j}$s own chores of B', but can only do chores of B. Now, to complete the loop, we introduce another set of $2n$ agents, namely $a'_1, a'_2, \ldots a'_{2n}$, where for each $i \in [2n]$ agent a'_i brings n units of chore b_i, and can do only the chores in set B'.

$$\forall i \in [n], \ d(a'_{2i-1}, b'_{2i-1}) = (1-\alpha') \quad \text{and} \quad d(a'_{2i-1}, b'_{2i}) = (1+\alpha')$$
$$d(a'_{2i}, b'_{2i-1}) = (1+\alpha') \quad \text{and} \quad d(a'_{2i}, b'_{2i}) = (1-\alpha'),$$

for some small $\alpha' \gg n^2 \cdot \alpha$. Next, we add a set of $2n$ agents for whom we will ensure certain *fixed earnings*, namely $a^f_1, \ldots, a^f_{2n}$, and they can do only chores in set B.

$$d(a^f_{2i-1}, b_{2i-1}) = (1-\alpha) \quad \text{and} \quad d(a^f_{2i-1}, b_{2i}) = (1+\alpha)$$
$$d(a^f_{2i}, b_{2i-1}) = (1+\alpha) \quad \text{and} \quad d(a^f_{2i}, b_{2i}) = (1-\alpha),$$

The instance $E(I)$ has additional agents and chores, and none of them has finite disutility towards the chores in set B. At any CE, our instance satisfies the following five key properties.

- *Pairwise equal endowments:* In $E(I)$, the total endowment of chore b_{2i-1} equals the total endowment of the chore b_{2i}, and similarly total endowment of b'_{2i-1} equals total endowment of b'_{2i}. Also the total endowments of each of these chores is $\mathcal{O}(n)$; from the above construction it is easy to see that this holds for chores in set B and B'.

- *Fixed earning:* In any CE, for each $i \in [2n]$, we have the total earning of agent a^f_i is $(1-\alpha') \cdot (2n - \sum_{j \in [2n]} \mathbf{M}_{j,i})$.

- *Price equality:* In any CE, the sum of prices of the chores b_{2i-1} and b_{2i} is the same for all $i \in [n]$, and it is equal to the sum of prices of the chores b'_{2i-1} and b'_{2i} for all $i \in [n]$. Since the prices of the chores at a CE are scale invariant, we can assume, without loss of generality, that for all $i \in [n]$, we have $p(b_{2i-1}) + p(b_{2i}) = p(b'_{2i-1}) + p(b'_{2i}) = 2$.

- *Price regulation:* In any CE, for all $i \in [2n]$ we have

$$\frac{1-\alpha}{1+\alpha} \le \frac{p(b_{2i-1})}{p(b_{2i})} \le \frac{1+\alpha}{1-\alpha} \quad \text{and} \quad \frac{1-\alpha'}{1+\alpha'} \le \frac{p(b'_{2i-1})}{p(b'_{2i})} \le \frac{1+\alpha'}{1-\alpha'}.$$

- *Reverse ratio amplification:* In any CE, for all $i \in [2n]$, if $\frac{p(b_{2i-1})}{p(b_{2i})} = \frac{1+\alpha}{1-\alpha}$, then we have $\frac{p(b'_{2i-1})}{p(b'_{2i})} = \frac{1-\alpha'}{1+\alpha'}$ and similarly when $\frac{p(b_{2i-1})}{p(b_{2i})} = \frac{1-\alpha}{1+\alpha}$, we have $\frac{p(b'_{2i-1})}{p(b'_{2i})} = \frac{1+\alpha'}{1-\alpha'}$.

We show that our instance $E(I)$ satisfies both SC_1 and SC_2 conditions of Theorem 4, and thereby it admits a CE. Using the above five properties satisfied by a CE, we next describe how the prices at CE give the equilibrium strategy vector of the polymatrix game I. Given a CE price vector p of $E(I)$, construct a vector x as follows.

$$\forall i \in [2n], \quad x_i = \frac{p(b_i') - (1 - \alpha')}{2 \cdot \alpha'} \tag{6}$$

We will show that x satisfies the Nash equilibrium conditions described in Problem 6 for instance I, thereby completing the reduction. First, let us argue why x is a valid strategy profile for the game, i.e., for each $i \in [n]$, we have $x_{2i-1} \geq 0$, $x_{2i} \geq 0$, and $x_{2i-1} + x_{2i} = 1$. Using the *price equality* and *price regulation* properties, it follows that for each $i \in [2n]$, $(1 - \alpha') \leq p(b_i') \leq (1 + \alpha')$. This immediately implies $x_i \geq 0$, $\forall i \in [2n]$. Furthermore, for each $i \in [n]$, we have $x_{2i-1} + x_{2i} = \frac{p(b_{2i-1}') + p(b_{2i}') - 2(1 - \alpha')}{2 \cdot \alpha'} = \frac{2\alpha'}{2\alpha'} = 1$.

Next, we will show that x satisfies the equilibrium conditions, namely, if $x^T \cdot \mathbf{M}_{*,2i} > x^T \cdot \mathbf{M}_{*,2i-1} + \frac{1}{n}$, then $x_{2i-1} = 0$; the proof for the symmetric condition is similar. To show that $x_{2i-1} = 0$, it suffices to show that $p(b_{2i-1}') = (1 - \alpha')$ (by (6)). So let us assume that $x^T \cdot \mathbf{M}_{*,2i} > x^T \cdot \mathbf{M}_{*,2i-1} + \frac{1}{n}$. The positive correlation between x and p imply that the total endowment money of the agents $\{a_{j,2i} \mid j \in [2n]\}$ is larger than that of the agents $\{a_{j,2i-1} \mid j \in [2n]\}$. Since the agents $\{a_{j,2i} \mid j \in [2n]\}$ prefer chore b_{2i} to b_{2i-1} and the agents $\{a_{j,2i-1} \mid j \in [2n]\}$ prefer chore b_{2i-1} to b_{2i}, the difference in their endowment money would further enforce that chore b_{2i} is higher priced than chore b_{2i-1}. Next we formalize this intuition, and how that leads to $p(b_{2i-1}') = (1 - \alpha')$.

The only agents who can consume chores b_{2i-1} and b_{2i} are the ones in $\{a_{j,2i}, a_{j,2i-i} \mid j \in [2n]\} \cup \{a_i^f \mid i \in [2n]\}$. Out of all these agents, the agents $A(b_{2i}) = \{a_{j,2i} \mid j \in [2n]\} \cup \{a_{2i}^f \mid i \in [n]\}$ are the agents that have $(1-\alpha)$ disutility towards b_{2i} and $(1+\alpha)$ disutility towards b_{2i-1}. Symmetrically define set $A(b_{2i-1})$. The total endowment money of the agents in $A(b_{2i})$ (say $P(b_{2i})$) is,

$$= \sum_{j \in [2n]} \mathbf{M}_{j,2i} \cdot p(b_j') + (1 - \alpha') \cdot (2n - \sum_{j \in [2n]} \mathbf{M}_{j,2i}) \qquad \text{(fixed earning of } a_{2i}^f)$$

$$= \sum_{j \in [2n]} \mathbf{M}_{j,2i} \cdot (2\alpha' \cdot x_j + (1 - \alpha')) + (1 - \alpha') \cdot (2n - \sum_{j \in [2n]} \mathbf{M}_{j,2i}) \qquad \text{(by equation (6))}$$

$$= \sum_{j \in [2n]} 2\alpha' \cdot x_j \cdot \mathbf{M}_{j,2i} + (1 - \alpha') \cdot \sum_{j \in [2n]} \mathbf{M}_{j,2i} + (1 - \alpha') \cdot (2n - \sum_{j \in [2n]} \mathbf{M}_{j,2i})$$

$$= 2\alpha' x^T \cdot \mathbf{M}_{*,2i} + 2n \cdot (1 - \alpha').$$

Similarly, the total endowment price of the agents in $A(b_{2i-1})$ (say $P(b_{2i-1})$) is $2\alpha' x^T \cdot \mathbf{M}_{*,2i-1} + 2n \cdot (1 - \alpha')$ which is less than $P(b_{2i})$. Recall again that, agents $A(b_{2i}) \cup A(b_{2i-1})$ can only consume chores b_{2i} and b_{2i-1}, and no other agents can consume these two chores. We now prove that if $A(b_{2i})$ earn all their endowment money of $P(b_{2i})$ by only consuming (doing) chore b_{2i}, then the price of b_{2i} ($p(b_{2i})$) has to be significantly high, leading to the violation of the *price-regulation* property: By *pairwise equal endowments* property, we have equal linear total endowments of both b_{2i} and b_{2i-1}, say cn. Therefore, if the agents in $A(b_{2i})$ earn all their endowment money of $P(b_{2i})$ only from chore b_{2i}, we have $p(b_{2i}) \geq P(b_{2i})/(cn)$, and $p(b_{2i-1}) \leq P(b_{2i-1})/(cn)$, implying that

$$p(b_{2i}) - p(b_{2i-1}) \geq \frac{P(b_{2i}) - P(b_{2i-1})}{cn} = \frac{2\alpha' \cdot (x^T \cdot \mathbf{M}_{*,2i} - x^T \cdot \mathbf{M}_{*,2i})}{cn} \geq \frac{2\alpha' \cdot 1/n}{cn} \gg 2\alpha \quad (\text{as } \frac{\alpha'}{n^2} \gg \alpha).$$

Since $p(b_{2i}) + p(b_{2i-1}) = 2$ (by the *price equality* property), this immediately implies that $p(b_{2i})/p(b_{2i-1}) > (1 + \alpha)/(1 - \alpha)$ (violation of the *price regulation* property). This implies that the

agents in $A(b_{2i})$ earn some of their endowment money of $P(b_{2i})$ from the chore b_{2i-1} as well. Since all agents in $A(b_{2i})$ have a disutility of $1-\alpha$ for the chore b_{2i} and $1+\alpha$ for the chore b_{2i-1}, agents in $A(b_{2i})$ can do some amount of b_{2i-1} only if $\frac{p(b_{2i-1})}{p(b_{2i})} \geq \frac{1+\alpha}{1-\alpha}$. Then, the *price regulation* property dictates that $\frac{p(b_{2i-1})}{p(b_{2i})} = \frac{1+\alpha}{1-\alpha}$. Finally, by *reverse ratio amplification* property, we have $\frac{p(b'_{2i-1})}{p(b'_{2i})} = \frac{1-\alpha'}{1+\alpha'}$. This implies $p(b'_{2i-1}) = (1-\alpha')$ and thereby $x_{2i-1} = 0$, when $x^T \cdot \mathbf{M}_{*,2i} > x^T \cdot \mathbf{M}_{*,2i-1} + \frac{1}{n}$. A very symmetric argument will show that if $x^T \cdot \mathbf{M}_{*,2i-1} > x^T \cdot \mathbf{M}_{*,2i} + 1/n$, then $x_{2i} = 0$.

While it is still intuitive for our choice of $\alpha' \gg n^2 \cdot \alpha$, that we can enforce $\frac{p(b_{2i-1})}{p(b_{2i})} = \frac{1+\alpha}{1-\alpha}$ when $x^T \cdot \mathbf{M}_{*,2i} > x^T \cdot \mathbf{M}_{*,2i-1} + \frac{1}{n}$, perhaps the most subtle and mysterious gadget in our reduction is the *reverse ratio amplification* gadget. This is also the most crucial gadget as such a gadget cannot be realized in the linear exchange setting with divisible goods (all the other gadgets can be realized). We now briefly sketch the *reverse ratio amplification* gadget.

Reverse Ratio Amplification Gadget. We sketch the reverse ratio amplification gadget for $\alpha' = 3\alpha/2$ and in the end elaborate how to make it work for $\alpha' \gg n^2 \cdot \alpha$. For each $i \in [n]$, we add an agent $\bar{a}_i$ that has an endowment of $\delta = n\alpha/2$ of both chores b'_{2i-1} and b'_{2i} and has a disutility of 1 for both chores b_{2i-1} and b_{2i} (and ∞ for all other chores), implying that $\bar{a}_i$ will earn her endowment money from the more expensive chore among b'_{2i-1} and b'_{2i}. We now show that when $\frac{p(b_{2i-1})}{p(b_{2i})} = \frac{1+\alpha}{1-\alpha}$, then we have $\frac{p(b'_{2i-1})}{p(b'_{2i})} = \frac{1-\alpha'}{1+\alpha'}$. So assume that $\frac{p(b_{2i-1})}{p(b_{2i})} = \frac{1+\alpha}{1-\alpha}$. The *price equality* property dictates that $p(b_{2i-1}) = 1 + \alpha$ and $p(b_{2i}) = 1 - \alpha$.

Now recall agents a'_{2i-1} and a'_{2i}. They own n units of chores b_{2i-1} and b_{2i} respectively and therefore their respective total endowment are $n \cdot (1+\alpha)$ and $n \cdot (1-\alpha)$. Like earlier, we now prove that if a'_{2i-1} earns her entire endowment money of $n(1+\alpha)$ from the chore b'_{2i-1}, then this causes violation of the *price regulation* property: If a'_{2i-1} earns her entire endowment from b'_{2i-1}, we have the price of b'_{2i-1} to be at least $\frac{n(1+\alpha)}{c'n} = \frac{1+\alpha}{c'}$, where $c'n$ is the total endowment of chores b'_{2i-1} and b'_{2i} (they are equal and linear in n by *pairwise equal endowments* property). Similarly, the only other agents having finite disutility towards the chore b'_{2i} are the agents a'_{2i} and $\bar{a}_i$. Therefore, using $\delta = n \cdot \alpha/2$, and $p(b'_{2i-1}) + p(b'_{2i}) = 2$ by the *price equality* property, we have,

$$p(b'_{2i}) \leq \frac{n(1-\alpha) + \delta \cdot (p(b'_{2i-1}) + p(b'_{2i}))}{c'n} = \frac{n(1-\alpha) + 2\delta}{c'n} = \frac{n(1-\alpha) + n\alpha}{c'n} = \frac{1}{c'}.$$

Therefore, we have that $p(b'_{2i-1}) \geq (1+\alpha)/c' > 1/c' \geq p(b'_{2i})$. This implies that the agent $\bar{a}_i$ earns her entire endowment price of $\delta \cdot (p(b'_{2i-1}) + p(b'_{2i})) = 2\delta$ from the chore b'_{2i-1} (as the disutility to price ratio of b'_{2i-1} is strictly less than that of b_{2i} for $\bar{a}_i$). Thus, we have that $p(b'_{2i-1}) \geq \frac{n(1+\alpha)+2\delta}{c'n}$ and $p(b'_{2i}) \leq \frac{n(1-\alpha)}{c'n}$, implying that, if we set $\alpha' = 3\alpha/2$, then

$$\frac{p(b'_{2i-1})}{p(b'_{2i})} \geq \frac{n \cdot (1+\alpha) + 2\delta}{n \cdot (1-\alpha)} = \frac{n \cdot (1+\alpha) + n\alpha}{n \cdot (1-\alpha)} = \frac{1+2\alpha}{1-\alpha} > \frac{1+\alpha'}{1-\alpha'} \qquad \text{(as } \delta = n \cdot \alpha/2)$$

Therefore, we have violation of *price regulation*. This implies that at a CE, agent a'_{2i-1} earns some of her endowment money from the chore b'_{2i}. Since a'_{2i-1} has a disutility of $(1-\alpha)$ for b'_{2i-1} and $(1+\alpha)$ for b'_{2i}, she can do the chore b'_{2i} only if $p(b'_{2i-1})/p(b'_{2i}) \geq (1-\alpha)/(1+\alpha)$. Then, by the *price regulation property*, we have $p(b'_{2i-1})/p(b'_{2i}) = (1-\alpha)/(1+\alpha)$. The proof for the case $\frac{p(b_{2i-1})}{p(b_{2i})} = \frac{1-\alpha}{1+\alpha} \implies \frac{p(b'_{2i-1})}{p(b'_{2i})} = \frac{1+\alpha'}{1-\alpha'}$ is symmetric.[10]

---

[10]Note that we are crucially using the fact that increasing the price of a chore makes it more attractive to an

When $\alpha' \gg n^2 \cdot \alpha$, i.e., when we need significantly higher amplification, we create a sequence of $K \in \mathcal{O}(\log(n))$ sets of chores $B^1 = \{b_1^1, b_2^1, \ldots, b_{2n}^1\}$, $B^2 = \{b_1^2, b_2^2, \ldots, b_{2n}^2\}, \ldots, B^K = \{b_1^K, b_2^K, \ldots, b_{2n}^K\}$ with $B^1 = B$ and $B^K = B'$. We enforce reverse ratio amplification between every consecutive sets of chores, i.e., for all $\ell \in [K]$ and for all $i \in [n]$, if $\frac{p(b_{2i-1}^\ell)}{p(b_{2i}^\ell)} = \frac{1+\alpha_\ell}{1-\alpha_\ell}$, then we have $\frac{p(b_{2i-1}^{\ell+1})}{p(b_{2i}^{\ell+1})} = \frac{1-\alpha_{\ell+1}}{1+\alpha_{\ell+1}}$ and similarly when $\frac{p(b_{2i-1}^\ell)}{p(b_{2i}^\ell)} = \frac{1-\alpha_\ell}{1+\alpha_\ell}$, then we have $\frac{p(b_{2i-1}^{\ell+1})}{p(b_{2i}^{\ell+1})} = \frac{1+\alpha_{\ell+1}}{1-\alpha_{\ell+1}}$, where $\alpha_{\ell+1} = 3\alpha_\ell/2$, thereby giving us the desired amplification from B to B'.

We re-emphasize that while the concepts of the gadgets used in our reduction are standard [CPY17, CT09], the key contribution is to realize these gadgets with just linear disutility functions. As a consequence, there are subtle, yet important technical differences to gadgets in [CPY17, CT09]. For instance, our amplification sequence has to be *alternating*, while the one in [CPY17] is not.

The formal proof with all the details can be found in Section 5, where we give the complete construction and prove that the instance exhibits all of the above five properties, as well as meets the sufficient conditions (Theorem 4) for existence of CE.

3   Applications and Further Related Work

The CEEI model is applicable to allocation problems naturally arising in a wide range of real-life settings such as dividing tasks among various team members, deciding teaching assignments between faculty, or splitting liabilities when dissolving a partnership. The exchange model also occurs naturally in many day to day scenarios, e.g., a set of university students teaching each other in a group study, to optimize the time and effort required. At a larger scale, *timebanks*[11] are such reciprocal service exchange platforms which have around 30,000 to 40,000 users from the United States. In a timebank, individuals from a certain community give services to one another and earn *time credit*. Thereafter, each individual uses their time credit to receive services. Competitive equilibrium (CE) provides a systematic way to do the exchange: it consists of prices (payment)[12] for chores and an allocation such that all chores are completely assigned and each agent gets her most preferred bundle (*optimal bundle*) subject to her budget constraint[13].

The problem of computing CE has been extensively studied. We only discuss previous work that appears most relevant. For linear CEEI (Fisher) model with goods, the CE set is captured by the Eisenberg-Gale convex program [EG59], which maximizes the Nash welfare defined as the geometric mean of agents' utilities. Later, Shmyrev [Shm09] obtained another convex program for this problem. [CDG+17] provides a dual connection between these and other related convex programs. A combinatorial polynomial-time algorithm for computing a CE is given in [DPSV08]. Later, strongly polynomial-time algorithms are obtained for this problem [Orl10, Vég12].

For the linear exchange model with goods, many convex programming formulations were obtained; see [DGV16] for details. Initial polynomial-time algorithms for this model are based on ellipsoid and interior-point methods [Jai07, Ye08]. The first combinatorial polynomial-time algorithm is given in [DM15], which was later improved in [DGM16]. Recently, [GV19] gives the first strongly polynomial-time algorithm for this problem.

The equilibrium computation problem in all models with goods is basically PPAD-hard when-

---

agent (agent $\bar{a}_i$ is only interested in b'_{2i-1} whenever it has a higher price than b'_{2i}). This property is not satisfied in the linear exchange setting with divisible goods (in fact this is the fundamental difference between the two settings) where we cannot realize this gadget.

[11] https://timebanks.org/

[12] Equivalent of time credit in time banks.

[13] Here the budget constraint of an agent is that she has to earn enough to pay for her initial set of chores.

ever the CE set is non-convex. In the CEEI (Fisher) model, it is polynomial-time for weak gross substitutes (WGS) and homogeneous utility functions [CMPV05, Eis61], and PPAD-hard for separable piecewise linear concave (SPLC) utilities [CT09]. In the exchange model, it is polynomial-time for WGS utilities [CMPV05, GHV21], and beyond that, it is essentially PPAD-hard [CPY17, CDDT09, CSVY06, GMVY17]. More recently, [CCPY22] shows PPAD-hardness for the Hylland-Zeckhauser model [HZ79], which is essentially the CEEI model under linear preferences with additional matching constraints.

For the linear CEEI model with chores, [BMSY17] consider the case where all (dis)utility values to be finite. [BMSY17] show that critical points of the geometric mean of agent's disutilities on the (Pareto) efficiency frontier are the CE profiles. By building on this characterization, [BCM22] give an FPTAS for finding an approximate CE. For the special case of constantly many agents (or chores), polynomial-time algorithms are known for computing a CE [BS19, GM20].

For the linear exchange model with chores, [CGMM21] gives a linear complementarity problem (LCP) formulation and a complementary pivot algorithm for computing a CE.

4   Combinatorial Algorithm

In this section, we give an exact $\tilde{\mathcal{O}}(n^2 m^2/\alpha^2)$-time algorithm for instances where every disutility value can be expressed as a power of $(1 + \alpha)$, i.e., $d_{ij} = (1 + \alpha)^{k_{ij}}$ for some polynomially bounded integer k_{ij}. Then, we show how to modify the same algorithmic framework to get a FPTAS for determining a CEEI when we make no assumption on the disutility values.

Our algorithm is iterative and each iteration comprises of two phases: (1) the price update phase that changes the prices of the chores without changing the allocation and then (2) allocation update phase which changes the allocation without altering the prices. Throughout the algorithm, we maintain that all chores are completely allocated and all agents earn their money from their respective MPB chores. Thus, the main goal of our algorithm is to find identical earnings for the agents (as this ensures that we are at a CEEI) while maintaining the invariants. Thereafter, we show that at the end of each iteration, we can improve the Nash welfare of the allocation by some multiplicative factor. We now elaborate the two phases.

4.1   Price Update

In this phase, we identify the *chores in demand*: Let e_i denote the earning of agent i, i.e., $e_i = \sum_j x_{ij} p_j$ where x_{ij} is the amount of chore j assigned to i and p_j is the payment for doing unit amount of chore j. Let S be the set of agents that have earnings less than one, i.e., $S = \{i \mid e_i < 1\}$ and let $\Gamma(S)$ be the set of MPB chores for the agents in S, i.e., $\Gamma(S) = \{j \mid d_{ij}/p_j = MPB_i \text{ for some } i \in S\}$. We call every chore in $\Gamma(S)$ as a chore in demand. Thereafter, we decrease the prices of all the chores in demand (chores in $\Gamma(S)$) until some agent in S gets interested in a chore in $[m] \setminus \Gamma(S)$, i.e., we have a new MPB edge from an agent in S to a chore outside $\Gamma(S)$. Note that although this update further decreases the earnings of the agents in S, it does not decrease Nash welfare as the disutility of each agent is independent of the prices (depends only on the allocation). The price update is summarized in Algorithm 1.

Algorithm 1 Price-Update(x, p, S)

$\Gamma(S) \leftarrow \{j \mid \exists i \in S, \text{ s.t. } d_{ij}/p_j = MPB_i\}$

$\gamma \leftarrow min_{i \in S, j \notin \Gamma(S)} \frac{MPB_i \cdot p_j}{d_{ij}}.$

for $j \in \gamma(S)$ **do**

  $p_j \leftarrow \gamma \cdot p_j.$

for $i \in S$ **do**

  $e_i \leftarrow \gamma \cdot e_i.$

4.2 Allocation Update

This is the second phase of an iteration in our algorithm. In this phase, we aim to determine an allocation that balances the earnings of the agents with the help of the new MPB edges created during the price update. Let E denote the set of new MPB edges that appear after the price update phase and let $J = \{j \mid (i, j) \in E\}$. Let $e_{max} = max_{i \in S} e_i$ and $e_{min} = min_{i \notin S} e_i$. The earning balancing is done depending on the total prices of the chores in J. In particular, if $\sum_{j \in J} p_j > (e_{min} - e_{max})/2$, then we compute an allocation (or equivalently a money flow) that maximizes the product of earnings (we call this procedure Balance-allocation$(\cdot)$)[14]. If $\sum_{j \in J} p_j \leq (e_{min} - e_{max})/2$, then we increase the earning of the agents in S and decrease that of agents in $[n] \setminus S$ by completely allocating the chores in J to agents in S along the MPB edges. The whole procedure is summarized in Algorithm 2.

Algorithm 2 Allocation-Update(x, p, S).

$E \leftarrow$ all new MPB edges from S to $[m] \setminus \Gamma(S)$ after price-update.

$J \leftarrow \{j \mid (i, j) \in E\}.$

$e_{max} \leftarrow max_{i \in S} e_i$ and $e_{min} \leftarrow min_{i \in [n] \setminus S} e_i.$

if $\sum_{j \in J} p_j > (e_{min} - e_{max})/2$ **then**

  $(x, p, S) \leftarrow$ **Balance-allocation**(p)

else

  for all $j \in J$ **do**

   Pick an arbitrary $i \in S$ such that $(i, j) \in E$.

   for all $i' \in [n] \setminus S$ such that $x_{i'j} > 0$, set $x_{i'j} \leftarrow 0$ and update $e_{i'} = e_{i'} - x_{i'j} p_j$

   Set $x_{ij} \leftarrow 1$ and update $e_i = e_i + p_j.$

Algorithm 3 Balance-allocation(p)

$x \leftarrow argmax_{x \in \mathbb{R}_{\geq 0}^{nm} \text{and} \sum_{i \in [n]} x_{ij} = 1} \prod_{i \in [n]} e_i$, where $e_i = \sum_{j \in [m]} x_{ij} p_j.$

$S \leftarrow$ the set of agents with lowest surplus.

  Overall, in the allocation-update phase, we determine an allocation that improves the Nash welfare of the allocation subject to the agents earning only along their MPB edges.

---

[14]We will show later in this section that we choose S in such a way that $e_{min} \geq e_{max}$.

4.3 Main Algorithm

Our overall algorithm alternates between the price-update and allocation-update phases. We start with an initial set of prices (where each price is of the form $(1 + \alpha)^k$ for some integer k[15]) and an allocation of the chores to the agents along MPB edges. We set S to be the set of agents with lowest earning. Thereafter, we run the price-update and allocation-update phases alternatively until all agents have equal budgets (see Algorithm 4). The convergence of our algorithm follows from the fact that after polynomially many iterations, we can observe an improvement of $1 + \Omega(\alpha^2)$ in the Nash welfare. Since the Nash welfare is upper bounded by $(n \cdot D_{max})^n$ where $D_{max} = \max_{(i,j):d_{ij} < \infty} d_{ij}$, our algorithm will converge after $\text{poly}(n, m, \log(D_{max})/\log(1 + \alpha^2)) = \text{poly}(n, m, \log(D_{max}), 1/\alpha)$ iterations.

Algorithm 4 Full Algorithm

---

    for all chores $j \in [m]$ **do**
        $p_j \leftarrow min_{i \in [n]} d_{ij}$.
    $(x, p, S) \leftarrow$ **Balance-allocation**(p).
    while $\exists i, j \in [n]$ such that $e_i \neq e_j$ **do**
        Price-update(x, p, S).
        Allocation-update(x, p, S)
    Return x, p.

We first argue about the correctness of the algorithm. The next series of observations will show that all chores are allocated and are allocated only along MPB edges. Therefore, if we have $e_i = e_j$ for all $i, j \in [n]$, then we are at a CEEI.

Observation 7. *Throughout the algorithm, the prices of the chores are powers of $(1 + \alpha)$. Furthermore, any price update phase decreases the prices of the chores in $\Gamma(S)$ and the earnings of the agents in S by at least $1 + \alpha$.*

Proof. We prove our claim by induction on the iterations. The claim holds by definition during the first iteration as we initialize each price p_j as $max_{i \in [n]} d_{ij}$ which is a power of $1 + \alpha$. Now consider any arbitrary iteration of the algorithm. By induction hypothesis, we have the prices to be powers of $1 + \alpha$. Since the disutilities are also powers of $1 + \alpha$, we have that MPB_i is also a power of $1 + \alpha$ for all $i \in [n]$. Let $i^* \in S$ and $j^* \notin \Gamma(S)$ be such that $\gamma = \frac{p_{j^*} \cdot MPB_{i^*}}{d_{i^* j^*}}$. Note that γ is also a power of $1 + \alpha$ and since the prices are scaled by γ, the new set of prices will also remain powers of $1 + \alpha$. We are left to show that $\gamma \leq (1 + \alpha)^{-1}$. Since γ is a power of $(1 + \alpha)$, it suffices to show that $\gamma < 1$ or equivalently $MPB_{i^*} < d_{i^* j^*}/p_{j^*}$, which is true as otherwise $j^* \in \Gamma(S)$. $\qquad \square$

Observation 8. *At the beginning of every iteration, for all agents $i \in [n] \setminus S$ and chores $j \in \Gamma(S)$, we have $x_{ij} = 0$.*

Proof. We prove our claim by induction on the iterations of the algorithm. We first show why this holds for the first iteration: we initialize S as the set of agents with lowest earning after computing an allocation that maximizes the product of budgets. Assume that there is an agent $i' \in [n] \setminus S$ and a chore $j \in \Gamma(S)$ such that $x_{i'j} > 0$. Let i be an agent in S such that $d_{ij}/p_j = MPB_i$ (note that such an i exists by the definition of S and $\Gamma(S)$). Also note that $e_i < e_{i'}$. We can change the allocation by increasing x_{ij} by $\varepsilon < x_{i'j}$ and decreasing $x_{i'j}$ by ε such that e_i increases and $e_{i'}$

---

decreases, but the sum of the earnings remain the same (in particular $e_i + e_{i'}$ remains same). Thus, the product of budgets improves, contradicting the fact that we started with an allocation with maximum product.

Now, consider any iteration of the algorithm. Let S' and $\Gamma(S')$ denote the set S and $\Gamma(S)$ at the beginning of the previous iteration. If the set S was updated from S' through a call to balance-allocation() in the previous iteration, then we can make a same argument as the induction base case. Otherwise, the set $S = S'$, however $\Gamma(S')$ contains more chores than $\Gamma(S)$ (as a result of new MPB edges appearing after the price-update phase). Notice that for all chores j in $\Gamma(S) \cap \Gamma(S')$ (which is $\Gamma(S')$), we have $x_{ij} = 0$ for all $j \in [n] \setminus S$ by induction hypothesis. For all chores $j \in \Gamma(S) \setminus \Gamma(S')$, we have set $x_{ij} = 0$ for all $i \in [n] \setminus S$ and $j \in \Gamma(S)$ by allocating all the chores in $\Gamma(S) \setminus \Gamma(S')$ completely to agents in S in Algorithm 2. $\qquad\square$

Observation 8 helps us prove that throughout the algorithm, all agents earn their money from their MPB chores and all chores remain fully allocated.

Lemma 9. *Throughout the algorithm,*

- *each agent earns her money only from her MPB chores, and*

- *for all $j \in [m]$ we have $\sum_{i \in [n]} x_{ij} = 1$.*

Proof. In the very first iteration of the algorithm, we initialize the prices such that every chore has at least one MPB edge incident to it: Note that for all $i \in [n]$, we have $MPB_i \geq 1$ as $d_{ij}/p_j \geq 1$ for all $i \in [n]$ and $j \in [m]$ as p_j is set to $min_{i \in [n]} d_{ij}$. Thus, for each chore j there exists an agent $i \in argmin_{i \in [n]} d_{ij}$ such that (i, j) is an MPB edge. Then, a call to Balance-allocation() ensures that all chores are completely allocated along MPB edges to the agents as otherwise we can increase the earning of some agent by allocating an underallocated chore along an MPB edge and thereby increase the product of earnings. We now show that the invariants are maintained during the price update and allocation update phases.

Price Update: During this phase, the prices of the chores in $[m] \setminus \Gamma(S)$ and the earnings of the agents in S are scaled by $\gamma < 1$ until a new MPB edge appears from some agent(s) in S to some chore in $\Gamma(S)$. Let p' and e' denote the new price and earning vector after price update. First note that there are no MPB edges from agents in S to chores in $[m] \setminus \Gamma(S)$ (by definition of $\Gamma(S)$). Now consider an agent $i \in S$. When we decrease the prices of chores in $\Gamma(S)$, all MPB chores of i will remain MPB chores as they are in $\Gamma(S)$ and the prices of all chores in $\Gamma(S)$ decreases by the same factor γ. So agent i still earns from her MPB chores. Also, $e'_i = \gamma \cdot e_i = \gamma \cdot \sum_{j \in [m]} x_{ij} \cdot p_j = \sum_{j \in \Gamma(S)} x_{ij} \cdot (\gamma \cdot p_j) = \sum_{j \in \Gamma(S)} x_{ij} \cdot p'_j$. Thus, all agents in S earn their entire earning from MPB chores after price update. Now, consider an agent $i' \notin S$. By Observation 8, $x_{i'j} = 0$ for all $j \in \Gamma(S)$, i.e., agent i' earns all of $e_{i'}$ from her MPB chores in $[m] \setminus \Gamma(S)$. Since the prices of the chores in $[m] \setminus S$ and the earnings of the agents in S remains unchanged ($e'_{i'} = e_{i'}$), agent i' earns all her money of $e'_{i'}$ from her MPB chores in $\Gamma(S)$. Thus, all agents in $[n] \setminus S$ also earn all their money from MPB chores after price update. Lastly, since the allocation remains unchanged during price update, all the chores remain completely allocated at the end of price update.

Allocation update: In this phase, the prices remain unaltered, implying that the MPB configuration does not change. If the allocation is updated through a call to Balance-allocation(), then each agent will earn their money from MPB chores and all chores will be completely allocated as otherwise we can increase the earning of some agent by allocating an underallocated chore along an MPB edge and thereby increase the product of earnings. If the allocation update does not involve a call to

Balance-allocation(), then all the chores in J[16] are allocated to agents in S along MPB edges. The earnings of the agents are adjusted accordingly. Therefore, all agents earn their money from MPB chores and all chores are completely allocated. □

We now prove a crucial invariant about the earnings of the agents in the S that our algorithm maintains throughout.

Observation 10. *At the beginning of every iteration of our algorithm, we have $max_{i \in S} e_i < min_{i \notin S} e_i$ where e is the earning vector.*

Proof. We prove our claim by induction on the iterations. The base case is trivially true as in the first iteration, S is set by Balance-allocation() to be the set of agents with lowest earning. Therefore we have $e_{max} < e_{min}$. Now consider any arbitrary iteration of the algorithm. If S was set by a call to Balance-allocation() in the previous iteration of the algorithm, then we trivially have $e_{max} < e_{min}$. So let us assume otherwise, i.e., Balance-allocation() was not invoked in the previous iteration. Let S' be the set S and e' be the earning vector at the beginning of the previous iteration. By induction hypothesis, we have $max_{i \in S'} e'_i < min_{i \notin S'} e'_i$. Let $\tilde{e}$ be the earning vector after the price-update phase (and before the allocation update phase). In the price update phase, we decrease the prices of the chores in $\Gamma(S)$ and the earnings of the agents in S and thus we have $max_{i \in S} \tilde{e}_i \leq max_{i \in S} e'_i \leq min_{i \notin S} e'_i = min_{i \notin S} \tilde{e}_i$.

Let E denote the set of new MPB edges that appear after price update and let $J = \{j \mid (i,j) \in E\}$. Let $\tilde{e}_{max} = max_{i \in S} \tilde{e}_i$ and $\tilde{e}_{min} = min_{i \notin S} \tilde{e}_i$. Since we are in the case where Balance-allocation() is not invoked, we have $(\sum_{j \in J} p_j < (\tilde{e}_{min} - \tilde{e}_{max})/2)$. Observe that Algorithm 2 allocates the chores in J to the agents in S along the MPB edges. As a result, the total earnings of the agents in S increases and that of the agents in $[n] \setminus S$ decreases (by the same amount). Let $\delta_i = e_i - \tilde{e}_i$ for all $i \in S$ and $\delta_i = \tilde{e}_i - e_i$ for all $i \notin S$. Note that $\sum_{i \in S} \delta_i = \sum_{i \in [n] \setminus S} \delta_i = \sum_{j \in J} p_j$. Therefore we have,

$$
\begin{aligned}
max_{i \in S} e_i &\leq \tilde{e}_{max} + \sum_{i \in S} \delta_i \\
&= \tilde{e}_{max} + \sum_{j \in J} p_j \\
&< \tilde{e}_{max} + (\tilde{e}_{min} - \tilde{e}_{max})/2 \\
&= \tilde{e}_{min} - (\tilde{e}_{min} - \tilde{e}_{max})/2 \\
&< \tilde{e}_{min} - \sum_{j \in J} p_j \\
&= \tilde{e}_{min} - \sum_{i \in S} \delta_i \\
&\leq min_{i \in [n] \setminus S} e_i \ .
\end{aligned}
$$

□

We now show that the allocation-update can only increase the product of earnings of the agents.

Lemma 11. *Let $e = \langle e_1, e_2, \ldots, e_n \rangle$ and $e' = \langle e'_1, e'_2, \ldots, e'_n \rangle$ be the earning vector before and after the call to Balance-allocation(p) in an allocation-update phase of the algorithm. We have $\prod_{i \in [n]} e'_i \geq (1 + \alpha^2/16) \prod_{i \in [n]} e_i$.*

---

[16]Recall from Algorithm 2, that E is the set of new MPB edges that appear from agents in S to chores in $[m] \setminus \Gamma(S)$ and $J = \{j \mid (i,j) \in E\}$

Proof. Let $\tilde{e} = \langle \tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_n \rangle$ be the earning vector before the price-update phase in the current iteration of the algorithm. By Observation 8, we have $max_{i \in S} \tilde{e}_i \leq min_{i \notin S} \tilde{e}_i$. By Observation 7, we decrease the prices of the chores in $\Gamma(S)$ and the earnings of the agents in S by a factor of at least $1 + \alpha$. Let e be the new earning vector before the allocation update phase. We have $(1 + \alpha) \cdot e_i \leq \tilde{e}_i$ for all $i \in S$, and $e_i = \tilde{e}_i$ for all $i \notin S$. Therefore, we have $(1 + \alpha) \cdot e_{max} < e_{min}$ [17].

Since the balanced allocation outputs an allocation that maximizes the product of the earnings, it suffices to show the existence of an allocation x' such that $\prod_{i \in [n]} e'_i \geq (1 + \alpha^2/16) \prod_{i \in [n]} e_i$ where $e'_i = \sum_{j \in [m]} x'_{ij} p_j$. Let x be the allocation before the allocation-update phase of the algorithm. Let E denote the set of new MPB edges after the price update and let $J = \{j \mid (i, j) \in E\}$. Note that prior to allocation update, $x_{ij} = 0$ for all $i \in S$ and $j \in J$. We initially set $x' = x$ and then change the allocation of the chores in J as follows: For each new MPB edge $(i, j) \in E$, and for each agent $i' \notin S$ such that $x'_{i'j} > 0$, we increase the consumption of the chore j for agent i and decrease the consumption of chore j for i' *by the same amount* until

- i' does not consume j anymore, i.e., $x'_{i'j} = 0$, or

- i consumes j entirely, i.e., $x'_{ij} = 1$, or

- the total increase in the earnings of the agents in S is $(e_{min} - e_{max})/2$, i.e., $\sum_{i \in S} e'_i - \sum_{i \in S} e_i = (e_{min} - e_{max})/2$.

First note that x' is a valid allocation as all the chores remain completely allocated in x'. Let $\boldsymbol{\Delta}$ denote the total increase in the earnings of the agents in S when we update the allocation to x' from x. Note that $\boldsymbol{\Delta}$ also equals the total decrease in the earnings of the agents in $[n] \setminus S$ as every time we increase the earning of an agent in S by some $\delta > 0$, we also decrease the earning of some agent in $[n] \setminus S$ by δ. It is also clear that $\boldsymbol{\Delta} = (e_{min} - e_{max})/2$ as otherwise all the goods in J would be completely allocated to agents in S and this would imply an increase in the earning of the agents in S by $\sum_{j \in [m]} p_j$ which is larger than $e_{min} - e_{max} > (e_{min} - e_{max})/2$ (if Balance allocation() is called, $\sum_{j \in J} p_j > e_{min} - e_{max}$). We are now ready to show the improvement in the product of earnings.

Let $\delta_i = e'_i - e_i$ for all $i \in S$ and $\delta_i = e_i - e'_i$ for all $i \notin S$. Note that $\delta_i \geq 0$ for all $i \in [n]$ and $\sum_{i \in S} \delta_i = \sum_{i \in [n] \setminus S} \delta_i = \boldsymbol{\Delta} = (e_{min} - e_{max})/2$. We now make a technical claim which helps us analyze the improvement in the product of earnings.

Claim 12. *Let $0 \leq y_1 \leq y_2 \leq \cdots \leq y_k$. Let $\delta_i \geq 0$ and $\gamma_i \geq 0$ for all $i \in [k]$. Furthermore, let $y_1 \geq \sum_{j \in [k]} \gamma_j$. Then, we have*

1. *$\prod_{i \in [k]}(y_i + \delta_i) \geq \prod_{i \in [k-1]} y_i \cdot (y_k + \sum_{j \in [k]} \delta_j)$, and similarly,*

2. *$\prod_{i \in [k]}(y_i - \gamma_i) \geq (y_1 - \sum_{j \in [k]} \gamma_j) \cdot \prod_{2 \leq i \leq k} y_i$.*

Proof. The proof follows immediately by the repeated applications of the following two facts. Given four numbers $x, y, \lambda_1, \lambda_2$ such that $\lambda_1 + \lambda_2 \leq x \leq y$ and

- $(x + \lambda_1)(y + \lambda_2) \geq x(y + \lambda_1 + \lambda_2)$: We have,

$$\begin{aligned}
(x + \lambda_1)(y + \lambda_2) &= xy + \lambda_1 y + \lambda_2 x + \lambda_1 \lambda_2 \\
&\geq xy + \lambda_1 y + \lambda_2 x \\
&\geq xy + \lambda_1 x + \lambda_2 x \qquad\qquad \text{(as } x \leq y) \\
&= x(y + \lambda_1 + \lambda_2).
\end{aligned}$$

---
[17]Recall from Algorithm 2, $e_{min} = min_{i \in [n] \setminus S} e_i$ and $e_{max} = max_{i \in S} e_i$.

- $(x - \lambda_1)(y - \lambda_2) \geq (x - \lambda_1 - \lambda_2)y$: We have,

$$
\begin{aligned}
(x - \lambda_1)(y - \lambda_2) &= xy - \lambda_1 y - \lambda_2 x + \lambda_1 \lambda_2 \\
&\geq xy - \lambda_1 y - \lambda_2 x \\
&\geq xy - \lambda_1 y - \lambda_2 y && \text{(as } x \leq y) \\
&= (x - \lambda_1 - \lambda_2)y. && \square
\end{aligned}
$$

Observe that,

$$
\begin{aligned}
\prod_{i \in [n]} e'_i &= \prod_{i \in S}(e_i + \delta_i) \cdot \prod_{i \in [n] \setminus S}(e_i - \delta_i) \\
&\geq \prod_{i \in S} e_i \cdot \frac{e_{max} + \boldsymbol{\Delta}}{e_{max}} \cdot \frac{e_{min} - \boldsymbol{\Delta}}{e_{min}} \cdot \prod_{i \in [n] \setminus S} e_i && \text{(by Claim 12)} \\
&= \prod_{i \in [n]} e_i \cdot \left(1 + \frac{\boldsymbol{\Delta}}{e_{max}}\right) \cdot \left(1 - \frac{\boldsymbol{\Delta}}{e_{min}}\right) \\
&= \prod_{i \in [n]} e_i \cdot \left(1 + \boldsymbol{\Delta}\frac{e_{min} - e_{max}}{e_{max}e_{min}} - \frac{\boldsymbol{\Delta}^2}{e_{max}e_{min}}\right) \\
&= \prod_{i \in [n]} e_i \cdot \left(1 + \frac{2\boldsymbol{\Delta}^2}{e_{max}e_{min}} - \frac{\boldsymbol{\Delta}^2}{e_{max}e_{min}}\right) && \left(\text{Substituting } \boldsymbol{\Delta} = \frac{e_{min} - e_{max}}{2}\right) \\
&= \prod_{i \in [n]} e_i \cdot \left(1 + \frac{\boldsymbol{\Delta}^2}{e_{max}e_{min}}\right) && (7)
\end{aligned}
$$

Since $(1 + \alpha)^{-1}e_{min} \geq e_{max}$, we have

$$
\begin{aligned}
\boldsymbol{\Delta} &\geq \frac{1 - (1 + \alpha)^{-1}}{2}e_{min} \\
&= \frac{\alpha}{2(1 + \alpha)}e_{min} \\
&\geq \frac{\alpha}{4}e_{min} && (\text{as } \alpha < 1).
\end{aligned}
$$

Substituting this lower bound on $\boldsymbol{\Delta}$, and using the fact that $e_{min} \geq e_{max}$, we have $\prod_{i \in [n]} e'_i \geq (1 + \alpha^2/16)\prod_{i \in [n]} e_i$. $\square$

Lemma 13. *Let $e = \langle e_1, e_2, \ldots, e_n \rangle$ and $e' = \langle e'_1, e'_2, \ldots, e'_n \rangle$ be the earning vector before and after an allocation-update phase of the algorithm. We have, $\prod_{i \in [n]} e'_i \geq \prod_{i \in [n]} e_i$.*

Proof. If the allocation update involves a call to Balance-allocation(), then the product of earnings improves by Lemma 11. We now show that the product of the earnings improve.

Now consider the case where there is no call to Balance-allocation(). In this case, we have $\left(\sum_{j \in J} p_j < (e_{min} - e_{max})/2\right)$[18]. Observe that Algorithm 2 allocates the chores in J to the agents in S along the MPB edges. As a result, the total earnings of the agents in S increases and that of

---

[18]Recall that E is the set of new MPB edges that appear from S to $[m] \setminus S$ after the price update phase and $J = \{j \mid (i, j) \in E\}$.

the agents in $[n] \setminus S$ decreases (by the same amount). Let $\delta_i = e'_i - e_i$ for all $i \in S$ and $\delta_i = e_i - e'_i$ for all $i \notin S$. Note that $\sum_{i \in S} \delta_i = \sum_{i \in [n] \setminus S} \delta_i = \sum_{j \in J} p_j$. Therefore we have,

$$
\begin{aligned}
\prod_{i \in [n]} e'_i &= \prod_{i \in S}(e_i + \delta_i) \cdot \prod_{i \in [n] \setminus S}(e_i - \delta_i) \\
&\geq \prod_{i \in S} e_i \cdot \frac{e_{max} + \sum_{i \in S} \delta_i}{e_{max}} \cdot \frac{e_{min} - \sum_{i \in [n] \setminus S} \delta_i}{e_{min}} \cdot \prod_{i \in [n] \setminus S} e_i \qquad \text{(by Claim 12)} \\
&= \prod_{i \in [n]} e_i \cdot \frac{e_{max} + \sum_{j \in J} p_j}{e_{max}} \cdot \frac{e_{min} - \sum_{j \in J} p_j}{e_{min}}.
\end{aligned}
$$

It suffices to show that $\frac{e_{max} + \sum_{j \in J} p_j}{e_{max}} \cdot \frac{e_{min} - \sum_{j \in J} p_j}{e_{min}} > 1$. This is indeed the case as $e_{max} + \sum_{j \in J} p_j \leq e_{min} - \sum_{j \in J} p_j$, i.e., the values e_{max} and e_{min} move closer to each other while still maintaining the same sum, implying that the product improves. $\qquad \square$

We are now ready to show convergence of our algorithm. We briefly sketch the overall idea: the price-update step does not change our potential as our potential only depends on the allocation which remains unaltered (even though the prices of the chores and the earnings of the agents change). Thereafter, with the help of new MPB edges we are able to improve the product of disutilities. However, note that during an allocation update, we do not change the prices of the chores. Therefore, the disutility of an agent is proportional to her earning as $D_i(x_i) = MPB_i \cdot e_i$ or equivalently $NSW(\mathbf{x}) = \prod_{i \in [n]} MPB_i \cdot \prod_{i \in [n]} e_i$. Since an allocation update increases $\prod_{i \in [n]} e_i$, it also improves the Nash welfare of the allocation. We now present the theorem and its formal proof.

Lemma 14. *Algorithm 4 computes a CEEI in $\mathcal{O}(nm/\alpha^2 \cdot \log(nD_{max}))$ iterations which involves at most $\mathcal{O}(n/\alpha^2 \cdot \log(nD_{max}))$ calls to Balance-allocation().*

Proof. To this end, we first argue that the Nash welfare never decreases throughout the algorithm: Note that $NSW(x) = \prod_{i \in [n]} D_i(x_i)$ is independent of the prices of the chores and therefore does not change during the price update phase. Prior to the allocation update, we can write each $D_i(x_i)$ as $MPB_i \cdot e_i$. Therefore, $NSW(x) = \prod_{i \in [n]} MPB_i \cdot \prod_{i \in [n]} e_i$. Note that we do not alter the prices of the chores during an allocation update and thus $\prod_{i \in [n]} MPB_i$ remains the same. By Lemma 13, $\prod_{i \in [n]} e_i$ increases during an allocation update phase and thus the Nash welfare increases.

We now bound the total number of iterations that invokes a Balance-allocation(). By Lemma 11, $\prod_{i \in [n]} e_i$ increases by $(1 + \alpha^2/16)$ every time Balance-allocation() is invoked. Since $\prod_{i \in [n]} MPB_i$ remains unaltered during allocation update, and $NSW(x) = \prod_{i \in [n]} MPB_i \cdot \prod_{i \in [n]} e_i$, we can conclude that each time Balance-allocation() is called, the Nash welfare increases by a factor of $1(+\alpha^2/16)$. Since $NSW(x) \leq (nD_{max})^n$ and the Nash welfare never decreases throughout the algorithm by Lemma 13, we can have at most $\mathcal{O}(\frac{\log((nD_{max})^n)}{\log(1+\alpha^2/16)}) \in \mathcal{O}(n/\alpha^2 \log(nD_{max}))$ many calls to Balance-allocation().

We now bound the number of iterations where Balance-allocation() is not invoked. In this case, note that $|\Gamma(S)|$ increases during an allocation update phase as the chores in J get added to $\Gamma(S)$. Therefore, we can have at most $\mathcal{O}(m)$ consecutive iterations that does not invoke Balance-allocation(). This implies that the total number of iterations of the algorithm is at most $\mathcal{O}(nm/\alpha^2 \cdot \log(nD_{max}))$. $\qquad \square$

To obtain a quadratic running time for our algorithm, it suffices to show how to implement Balance-allocation() in $\tilde{\mathcal{O}}(nm(n + m))$ time. We now elaborate this.

4.4 Implementing Balance-allocation() in $\mathcal{O}(nm(n+m)\log(nm))$ Time

We show that the problem of determining a Balance-allocation() can be reduced to the problem of finding a lexicographically optimal flow [Fuj80], which can be determined in $\mathcal{O}(nm\cdot(n+m)\log(nm))$ time.

Definition 15. *[Fuj80] We are given a directed graph $G = (V, E)$ with edge capacities. S^+ is a set of k source vertices, i.e., vertices in V without an incoming edge and t is a single sink vertex, i.e., a vertex in V without an outgoing edge. Given any flow f in G, let $\delta^+(v)$ denote the total outflow from a vertex $v \in V$. A lexicographically optimal flow is a maximum flow f from S^+ to t such that the vector $\langle \delta^+(s_1), \delta^+(s_2), \ldots, \delta^+(s_k) \rangle$ is lexicographically maximum subject to $s_\ell \in S^+$ for all $\ell \in [k]$ and $\delta^+(s_1) \le \delta^+(s_2) \le \cdots \le \delta^+(s_k)$.*

We now show how lexicographically optimum flows can be used to implement the Balance-allocation() subroutine. Given a price vector p, we define the market-network $M_p = ([n] \cup [m] \cup \{t\}, E)$ as a digraph that contains the agents $[n]$, chores $[m]$ and a sink vertex t. The directed edges from the agents to the chores are MPB edges at the prices p, i.e., $(i, j) \in E$ is $d_{ij}/p_j = MPB_i$. Edges from agents to chores have infinite capacity (meaning one can push any amount of flow through them). There is a directed edge from each chore j to the sink t with a capacity of p_j. Note that the set of agents have no incoming edges. The main goal of Balance-allocation() is to find a flow in the MPB graph that maximizes the product of the outflow of the sources. Once we have such a flow f, it is easy to find the allocation as $x_{ij} = f_{ij}/p_j$. We make an observation that any flow that maximizes the product of outflows of the sources is a lexicographically optimum flow.

Lemma 16. *Given a market-network G, a maximum flow f maximizes the product of outflows of the sources, $\prod_{s \in S^+} \delta^+(s)$ if and only if it is a lexicographically optimum flow.*

Proof. We crucially use the following characterization of a lexicographically optimum flow given in [Fuj80]

Claim 17. *([Fuj80]) A maximum flow f is a lexicographically optimal flow if and only if it minimizes the ℓ_2-norm of the outflow vector of the sources, i.e., the flow that minimizes $\sum_{s \in S^+} (\delta^+(s))^2$.*

Therefore, it suffices to show that any flow that minimizes the ℓ_2-norm of the outflows of the sources also maximizes the product of the outflows from the sources. This follows immediately when we write the KKT conditions for both the convex programs. Consider the convex program for minimizing the ℓ_2-norm of the outflows

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i \in [n]} \Big(\sum_{(i,j) \in E} f_{ij} \Big)^2 \\
\text{subject to} \quad & \sum_{(i,j) \in E} f_{ij} = p_j, \quad \forall (i,j) \in E \\
& f_{ij} \ge 0, \quad\quad\quad \forall (i,j) \in E
\end{aligned}
$$

The optimum solution will satisfy the KKT conditions. Let λ_j be the *free* dual variable corresponding to the constraint $\sum_{(i,j) \in E} f_{ij} = p_j$ and $\mu_{ij} \ge 0$ be the dual variable corresponding to $f_{ij} \ge 0$. The KKT conditions require, in addition to primal and dual feasibility,

- **Stationarity:** For all $(i,j) \in E$, $2(\sum_{(i,j) \in E} f_{ij}) + \lambda_j - \mu_{ij} = 0 \implies \sum_{(i,j) \in E} f_{ij} = (\mu_{ij} - \lambda_j)/2$

- **Complementary Slackness:** For all $(i,j) \in E$, $\mu_{ij} \cdot f_{ij} = 0$.

25

Consider any i and j such that $f_{ij} > 0$. Then $\mu_{ij} = 0$. Then, we have $\sum_{(i,j) \in E} f_{ij} = -\lambda_j/2$. This implies that all nodes i that have positive surplus towards the node j have the same outflow which is equal to $-\lambda_j/2$. Since $\mu_{ij} \geq 0$, this also implies that $-\lambda_j/2 \leq -(\lambda_j + \mu_{ij})/2$. This gives us the following characterization of the optimum flow.

Observation 18. *Let f be any flow in the market-network that minimizes the ℓ_2-norm of the outflows. If there are two source nodes i and i' that are adjacent to node j, and $f_{ij} > 0$, then (i) $\sum_{(i,k) \in E} f_{ik} = \sum_{(i',k) \in E} f_{i'k}$ or (ii) $\sum_{(i,k) \in E} f_{ik} < \sum_{(i',k) \in E} f_{i'k}$ and $f_{i'j} = 0$.*

We will now show that any flow that satisfies the property in Observation 18, also maximizes the product of outflows, implying that any flow that minimizes the ℓ_2-norm of the surpluses also maximizes the product of outflows. To this end, consider the program of maximizing the product of outflows or equivalently maximizing the sum of the logarithms of the outflows.

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i \in [n]} (\log(\sum_{(i,j) \in E} f_{ij})) \\
\text{subject to} \quad & \sum_{(i,j) \in E} f_{ij} = p_j, \qquad \forall (i,j) \in E \\
& f_{ij} \geq 0, \qquad \qquad \forall (i,j) \in E
\end{aligned}
\tag{8}
$$

Let λ_j be the *free* dual variable corresponding to the constraint $\sum_{(i,j) \in E} f_{ij} = p_j$ and $\mu_{ij} \geq 0$ be the dual variable corresponding to $f_{ij} \geq 0$. The KKT conditions require, in addition to primal and dual feasibility,

- **Stationarity:** For all $(i,j) \in E$, $-1/(\sum_{(i,j) \in E} f_{ij}) + \lambda_j - \mu_{ij} = 0$

- **Complementary Slackness:** For all $(i,j) \in E$, $\mu_{ij} \cdot f_{ij} = 0$.

Consider any flow f that satisfies the property in Observation 18. We will find the dual variables λ_js and μ_{ij}s that satisfy the KKT conditions of the convex program 8. We set $\lambda_j = 1/(\sum_{(i,j) \in E} f_{ij})$ where i is a source node such that $f_{ij} > 0$. We set $\mu_{ij} = \lambda_j - 1/(\sum_{(i,j) \in E} f_{ij})$. We prove that f_{ij}'s, μ_{ij}'s and λ_j's satisfy the KKT conditions for the convex program (8).

- **Stationarity:** We set $\mu_{ij} = \lambda_j - 1/(\sum_{(i,j) \in E} f_{ij})$ for all $(i,j) \in E$. Therefore, for all i, j, we have $-1/(\sum_{(i,j) \in E} f_{ij}) + \lambda_j - \mu_{ij} = 0$.

- **Complementary slackness:** For all $(i,j) \in E$ where $f_{ij} = 0$, we have $f_{ij} \cdot \mu_{ij} = 0$. For all i, j such that $f_{ij} > 0$, we have $\lambda_j = 1/(\sum_{(i,j) \in E} f_{ij})$, implying that $\mu_{ij} = \lambda_j - 1/(\sum_{(i,j) \in E} f_{ij}) = 0$.

- **Primal feasibility:** This is satisfied by the definition of a flow in a market-network.

- **Dual feasibility:** We want to show that $\mu_{ij} \geq 0$ for all i, j. For all i, j such that $f_{ij} > 0$, we have $\lambda_j = 1/(\sum_{(i,j) \in E} f_{ij})$, implying that $\mu_{ij} = \lambda_j - 1/(\sum_{(i,j) \in E} f_{ij}) = 0$. For all i, j such that $f_{ij} = 0$, let i' be a source node such that $f_{i'j} > 0$. Note that by Observation 8, we have $\sum_{(i,j) \in E} f_{ij} \geq \sum_{(i,j) \in E} f_{i'j}$. Therefore, we have,

$$
\begin{aligned}
\mu_{ij} &= \lambda_j - 1/(\sum_{(i,j) \in E} f_{ij}) \\
&\geq \lambda_j - 1/(\sum_{(i,j) \in E} f_{i'j}) \\
&= 0 \qquad \qquad \qquad \qquad \text{as } \lambda_j = 1/(\sum_{(i,j) \in E} f_{i'j}).
\end{aligned}
$$

□

Thus, Balance-allocation() can be implemented with one call to finding a lexicographically optimum flow in the MPB graph. [Fuj80] give a fast algorithm to determine a lexicographically optimum flow.

Lemma 19. *[Fuj80] Given a digraph $G = (V, E)$ with edge capacities, source set S^+ and a sink t, one can find a lexicographically optimum flow in $\mathcal{O}(|V| \cdot |E| \log(|V| \cdot |E|))$.*

Lemma 19 immediately gives us an efficient algorithm to implement Balance-allocation().

Corollary 20. *We can implement the subroutine Balance-allocation() in $\mathcal{O}(nm^2 \cdot \log(nm))$ time*

Proof. Follows immediately from Lemma 19 after substituting $|V| = n + m$ and $|E| = nm$ and from the fact that $m > n$. □

We are ready to bound the running time of our algorithm.

Theorem 21. *Given a chore division instance with rounded disutilities, we can determine a CEEI in $\mathcal{O}(n^2 m^2/\alpha^2 \cdot \log(nD_{max}) \log(nm)) \in \tilde{\mathcal{O}}(n^2 m^2/\alpha^2)$ time.*

Proof. By lemma 14, Algorithm 4 has a total of $\mathcal{O}(nm/\alpha^2 \log(nD_{max}))$ iterations, out of which at most $\mathcal{O}(n/\alpha^2 \log(nD_{max}))$ involves invocation to Balance-allocation(). We now bound the total time taken by our algorithm for all price updates and all allocation updates. Note that each price update can be implemented in $\mathcal{O}(nm)$ as it mainly involves finding the agent $i \in S$ and chore $j \notin \Gamma(S)$ such that $d_{ij}/(MPB_i \cdot p_j)$ is minimum. Since there are at most $\mathcal{O}(nm/\alpha^2 \log(nD_{max}))$ iterations, the total time taken on all price update phases is $\mathcal{O}(n^2 m^2/\alpha^2 \log(nD_{max}))$.

Now, we bound the total time taken on all allocation updates. First, look into all the allocation update calls that involves a call to Balance-allocation(). Each such iteration can be implemented in $\mathcal{O}(nm^2 \log(nm))$ time by Corollary 20 and there are at most $\mathcal{O}(n/\alpha^2 \log(nD_{max}))$ such iterations. Thus, the total time taken on all allocations that involve call to Balance-allocation() is $\mathcal{O}(n^2 m^2/\alpha^2 \log(nD_{max}) \log(nm))$. Lastly, look at all allocation update phases that does not involve call to Balance-allocation(). Each such phase can be implemented in $\mathcal{O}(nm)$ time as it mainly involves allocating the chores in J to agents in S along MPB edges and there are at most $\mathcal{O}(nm)$ MPB edges. Since there are at most $\mathcal{O}(nm/\alpha^2 \log(nD_{max}))$ many such iterations, the total time spend on iterations that do not involve call to Balance-allocation() is $\mathcal{O}(n^2 m^2/\alpha^2 \log(nD_{max}))$. Overall, the total running time is $\mathcal{O}(n^2 m^2/\alpha^2 \cdot \log(nD_{max}) \log(nm)) \in \tilde{\mathcal{O}}(n^2 m^2/\alpha^2)$. □

4.5 Modification of Algorithm 4 to get a FPTAS

We show how to extend the combinatorial algorithmic framework to get a $(1 - \varepsilon)$-CEEI in $\mathcal{O}(n^4 m^2 \cdot \log(nD_{max}))$ time. With subtle changes, Algorithm 4 can be adapted to give a $(1 - \varepsilon)$-CEEI, when the disutility values are arbitrary. We make no changes to the price update phase. In the allocation update phase, the only change we make is in initializing and updating the set S. In particular, after each call to balance allocation, the set S is determined by the following procedure: Renumber the agents according to increasing order of their earnings. Let i be such that e_i/e_{i+1} is maximum. Then $S \leftarrow [i]$. We run the algorithm as long as $max_{i \in [n]} e_i / min_{i \in [n]} e_i \geq 1 + \varepsilon$. The analysis of convergence is a little more involved.

Note that Observation 8, Lemma 9 and Observation 10 still hold as they only rely on the fact that every time S is updated, we have $max_{i \in S} e_i < min_{i \notin S} e_i$, which is always true in the new algorithm. We only need to argue about Lemma 11. This would require more work as we may

not always have the multiplicative gap of $1 + \alpha$ between $max_{i \in S} e_i$ and $min_{i \notin S} e_i$. However, we circumvent this problem by showing that between every consecutive calls to balance the product of earnings increases by a multiplicative factor of $1 + \varepsilon^2/256n^2$.

Consider the set of iterations (say $iter_1$ ro $iter_r$) following the iteration involving a call to Balance-allocation() until the next call to Balance-allocation(). Let e be the earning vector, p be the price vector, and $S = [i]$ (after renumbering the agents in increasing order of earnings) at the beginning of $iter_1$. Since we have $e_1/e_n \geq 1 + \varepsilon$, we have $e_i/e_{i+1} \geq (1 + \varepsilon)^{1/n} \geq 1 + \varepsilon/2n$. Also note that $e_{min}/e_{max} = min_{i \in S} e_i / max_{i \in S} e_i = e_i/e_{i+1} \geq 1 + \varepsilon/2n$.

We now look into $iter_r$. Let e' be the earning vector, p' be the price vector, and S' be the set S at the beginning of $iter_r$. Note that $S' = S$, as the set S has not changed (it only changes through an invocation to Balance-allocation(), which has not happened from $iter_1$ to $iter_{r-1}$), while some of the chores from the set $[m] \setminus \Gamma(S)$ in the earlier iterations got allocated along MPB edges to agents in S. If $e'_{min}/e'_{max} = min_{i \in S} e'_i / max_{i \in S} e'_i \geq 1 + \varepsilon/4n$, then by the same analysis in Lemma 11, we can argue that the product of disutilities improve by a factor of $1 + \varepsilon^2/256n^2$.

Now, we consider the case that $e'_{min}/e'_{max} \leq 1 + \varepsilon/4n$. In this case, we show that the ratio of the product of disutilities at the beginning of $iter_r$ to that at the beginning of $iter_1$ is $1 + \varepsilon^2/16n^2$. Let J denote the set of chores that were added to the set $\Gamma(S)$ from $iter_1$ to $iter_{r-1}$. Let p_j denote the price of a chore $j \in J$ in the iteration in which it was added to $\Gamma(S)$ via MPB edges. Note that if a chore j is added to $\Gamma(S)$ via new MPB edges in $iter_\ell$, the price of chore j remained unaltered from $iter_1$ to $iter_\ell$, i.e., p_j is the price of the chore in $iter_1$ also. Let β denote the total multiplicative decrease across all price-update phases from $iter_1$ to $iter_{r-1}$, i.e., $\beta = \prod_{\ell \in [r-1]} \gamma_\ell$, where γ_ℓ is the amount by which the prices of the chores in $\Gamma(S)$ has been reduced in the price update phase of $iter_\ell$. Note for each $i \in S$, we have $e'_i \geq \beta \cdot (e_i + \delta_i)$, where $\delta_i = \sum_{j \in J} x'_{ij} p_j$ where x' is the allocation at the end of $iter_{r-1}$ (or equivalently, at the beginning of $iter_1$): This is due to the fact that if z units of chore $j \in J$ is allocated to agent i in $iter_\ell$, then the price of this fractional amount of chore j gets scaled down by $\prod_{\ell+1 \leq q \leq r-1} \gamma_q \geq \beta$ until $iter_{r-1}$ in the earning of agent i. Similarly, for each $i \notin S$, we have $e'_i = e_i - \delta_i$ where $\delta_i = \sum_{j \in J} x_{ij} p_j$ where x is the allocation at $iter_1$: This is due to the fact that the only decrease in the earnings of the agents outside S comes from loosing the consumption of the chores in J. Note that $\sum_{i \in S} \delta_i = \sum_{i \in S} \sum_{j \in J} x'_{ij} p_j = \sum_{j \in J} p_j$ as by the end of $iter_{r-1}$ all chores in J are allocated to agents in S and similarly $\sum_{i \notin S} \sum_{j \in J} x_{ij} p_j = \sum_{j \in J} p_j$ as initially (at the beginning of $iter_1$), all chores in J were allocated to agents in $[n] \setminus S$. Therefore, we have $\sum_{i \in S} \delta_i = \sum_{i \notin S} \delta_i$.

Let $i' \in S$ be such that $e'_{max} = e'_{i'}$ and $\tilde{i} \in [n] \setminus S$ be such that $e'_{min} = e'_{\tilde{i}}$. Similar to the proof of Lemma 11, we give a lower-bound on $\Delta = \sum_{i \in [n]} \delta_i$ by giving a lower-bound on $\delta_{i'} + \delta_{\tilde{i}}$. Since we have $e'_{min}/e'_{max} \leq 1 + \varepsilon/4n$, we can conclude that

$$\frac{e_{min} - \delta_{\tilde{i}}}{\beta(e_{max} + \delta_{i'})} \leq 1 + \frac{\varepsilon}{4n}$$
$$\implies e_{min} - \delta_{\tilde{i}} \leq \beta(e_{max} + \delta_{i'}) + \frac{\varepsilon}{4n}(\beta)(e_{max} + \delta_{i'})$$
$$\implies \beta\delta_{i'} + \delta_{\tilde{i}} \geq e_{min} - \beta e_{max} - \frac{\varepsilon}{4n}(\beta(e_{max} + \delta_{i'}))$$

Since $e_{min} \geq e'_{min} \geq e'_{max} \geq \beta(e_{max} + \delta_{i'})$, we have ,

$$\beta\delta_{i'} + \delta_{\tilde{i}} \geq e_{min} - \beta e_{max} - \frac{\varepsilon}{4n}(e_{min})$$

$$\implies \beta\delta_{i'} + \delta_{\tilde{i}} \geq (1 - \frac{\varepsilon}{4n})e_{min} - \beta e_{max}$$

$$\implies \delta_{i'} + \delta_{\tilde{i}} \geq (1 - \frac{\varepsilon}{4n})e_{min} - e_{max} \qquad \text{(as } \beta < 1\text{)}$$

$$\implies \delta_{i'} + \delta_{\tilde{i}} \geq (1 - \frac{\varepsilon}{4n})e_{min} - (1 + \frac{\varepsilon}{2n})^{-1}e_{min} \qquad \text{(as } e_{min} \geq (1 + \frac{\varepsilon}{2n})e_{max}\text{)}$$

$$\implies \delta_{i'} + \delta_{\tilde{i}} \geq ((1 - \frac{\varepsilon}{4n}) - (1 + \frac{\varepsilon}{2n})^{-1})e_{min}$$

$$\implies \delta_{i'} + \delta_{\tilde{i}} \geq \left(\frac{(1 - \frac{\varepsilon}{4n})(1 + \frac{\varepsilon}{2n}) - 1}{1 + \frac{\varepsilon}{2n}} \right) e_{min}$$

$$\implies \delta_{i'} + \delta_{\tilde{i}} \geq \frac{\varepsilon}{16n} \cdot e_{min}.$$

Therefore, $\Delta \geq (\varepsilon/16n) \cdot e_{min}$. We are now ready to show that ratio of the product of disutilities at the beginning of $iter_r$ to that at the beginning of $iter_1$ is $1 + \varepsilon^2/256n^2$. Let MPB_i denote the MPB value of agent i in $iter_1$ and MPB'_i denote the MPB value of agent i in $iter_{r-1}$. Note that $MPB'_i = (1/\beta) \cdot MPB_i$ for all $i \in S$ and $MPB'_i = MPB_i$ for all $i \notin S$. Therefore, we have the product of disutilities in $iter_{r-1}$ to that in $iter_1$ as

$$\frac{\prod_{i \in S} MPB'_i \cdot \prod_{i \notin S} MPB'_i \cdot \prod_{i \in [n]} e'_i}{\prod_{i \in S} MPB_i \cdot \prod_{i \notin S} MPB_i \cdot \prod_{i \in [n]} e_i}$$

$$= \frac{(1/\beta)^{|S|} \cdot \prod_{i \in [n]} e'_i}{\prod_{i \in [n]} e_i}$$

$$\geq \frac{(1/\beta)^{|S|} \cdot \prod_{i \in S} \beta(e_i + \delta_i) \cdot \prod_{i \notin S}(e_i - \delta_i)}{\prod_{i \in [n]} e_i}$$

$$= \frac{\prod_{i \in S}(e_i + \delta_i) \cdot \prod_{i \notin S}(e_i - \delta_i)}{\prod_{i \in [n]} e_i}$$

$$\geq \frac{(e_{max} + \Delta)(e_{min} - \Delta)}{e_{max}e_{min}} \qquad \text{(by Claim 12)}.$$

From here on, we can continue the same analysis as in the proof of Lemma 11 and get the desired bound.

Therefore, there can be at most $\mathcal{O}(\log_{1+\varepsilon^2/256n^2}((nD_{max})^n)) \in \mathcal{O}(n^3 \log(nD_{max})/\varepsilon^2)$ calls to Balance-allocation(). Between any two consecutive calls to Balance-allocation(), we can have at most $\mathcal{O}(m)$ iterations (as the size of $\Gamma(S)$ strictly increases in each of these iterations). Therefore, there are at most $\mathcal{O}(n^3 m \log(nD_{max})/\varepsilon^2)$ many iterations that do not involve a call to Balance-allocation(). Each call to Balance-allocation() can be implemented in $\mathcal{O}(nm(n+m))$ time and each iteration that does not involve a call to Balance-allocation() can be implemented in $\mathcal{O}(nm)$ time. Therefore, the desired running time is $\mathcal{O}(\frac{n^3 \log(nD_{max})}{\varepsilon^2} \cdot nm(n+m) \log(nm) + \frac{n^3 m \log(nD_{max})}{\varepsilon^2} \cdot nm) \in \mathcal{O}(n^4 m^2 \log(nD_{max}) \log(nm)/\varepsilon^2)$.

Theorem 22. *There exists a $\mathcal{O}((n^4 m^2 \log(nD_{max}) \log(nm))/\varepsilon^2)$ time combinatorial algorithm that determines a $(1 - \varepsilon)$-CEEI when agents have arbitrary disutility values.*

Observe that if we do not stop our algorithm, until earning of all the agents are the same, then we reach an exact CE. Since each balanced flow computation finds the Nash-welfare maximizing

29

allocation for the given MPB configuration, it follows that the MPB configuration cannot repeat. This together with Nash-welfare as the potential function puts the problem in PLS. This algorithm and argument extends to the Fisher model as well, where agents may have different earning requirements – if the earning requirement of agent i is η_i then in every iteration sort them in the decreasing order of $\frac{e_i}{\eta_i}$.

Corollary 23. *The problem of finding competitive equilibrium in the Fisher model is in PLS.*

5 PPAD-Hardness for the Exchange Model

In this section, we show that finding (approximate) CE is intractable in the exchange model with chores and linear valuations. It is well-known that under exchange model a CE may not exist, and recently [CGMM22] showed existence under certain mild sufficiency conditions, namely conditions SC_1 and SC_2 of Theorem 4. We show that even for the instances satisfying these conditions, it is PPAD-hard to find a $(1 - 1/poly(n))$-approximate CE (Definition 3). In particular, we will show that any polynomial time algorithm that determines a $(1 - 1/poly(n))$-approximate CE on instances that satisfy conditions SC_1 and SC_2 of Theorem 4, will yield an algorithm to find a $1/n$-approximate Nash equilibrium in a *normalized polymatrix game*. The latter is known to be PPAD-hard [CPY17]. Next we recall the normalized polymatrix game problem:

Problem. *(Normalized Polymatrix Game) [CPY17]*
Given: *A $2n \times 2n$ rational matrix $\mathbf{M}$ with every entry in $[0, 1]$ and $\mathbf{M}_{i,2j-1} + \mathbf{M}_{i,2j} = 1$ for all $i \in [2n]$ and $j \in [n]$.*
Find: *An approximate Nash equilibrium strategy vector $x \in \mathbb{R}^{2n}_{\geq 0}$ such that $x_{2i-1} + x_{2i} = 1$ and*

$$x^T \cdot \mathbf{M}_{*,2i-1} > x^T \cdot \mathbf{M}_{*,2i} + \frac{1}{n} \implies x_{2i} = 0.$$
$$x^T \cdot \mathbf{M}_{*,2i} > x^T \cdot \mathbf{M}_{*,2i-1} + \frac{1}{n} \implies x_{2i-1} = 0.$$

where $M_{*,k}$ represents the kth column of the matrix $\mathbf{M}$.

From the next section onward, we elaborate our construction and proof of reduction: We first introduce all agents and chores. Thereafter, we define the disutility matrix and endowment matrix and show that our instance satisfies the sufficiency conditions of Theorem 4, and therefore admits a CE. Then, we show that our instance and the prices at $(1 - 1/poly(n))$-approximate CE exhibits the five properties of *pairwise equal endowments, (approximate) fixed earning, (approximate) price equality, price regulation* and *reverse ratio amplification* (as discussed in Section 2.3), and thus in polynomial-time we can construct the equilibrium strategy vector x for I from any $(1 - \frac{1}{poly(n)})$-approximate CE in $E(I)$. The reader is highly encouraged to read Section 2.3 before reading the elaborate version of the proof to get the idea of the overall proof sketch.

5.1 Agent and Chore Sets

We define the set of $K = 2c \cdot \lceil \log(n) \rceil$ many sets of chores, where $c = 3$ (observe crucially that K is even),

$$B_k = \left\{ \cup_{i \in [2n]} b_i^k \right\} \qquad\qquad \text{for all } k \in [K],$$

and K many sets of agents

$$A_k = \begin{cases} \{a_i^1 \mid i \in [2n]\} \cup \{a_i' \mid i \in [2n]\} & \text{when } k = 1, \\ \{a_i^k \mid i \in [2n]\} \cup \{\overline{a}_i^k \mid i \in [n]\} & \text{when } 2 \leq k \leq K - 1, \\ \{a_{i,j}^K \mid i,j \in [2n]\} \cup \{\overline{a}_i^K \mid i \in [n]\} & \text{when } k = K. \end{cases}$$

We remark that the sets A_1, A_K of agents and sets B_1, B_K of chores are to enforce the fixed earning, price equality and price regulation properties as mentioned in sketch of the reduction in Section 2.3, while the sets A_k of agents and B_k of chores for all $2 \leq k \leq K - 1$ are to primarily enforce reverse ratio amplification property as mentioned in Section 2.3. We now define the disutility matrix and the endowment matrix of the instance.

Disutility Matrix and the Disutility Graph. The disutility graph for our instance will be a disjoint union of complete bipartite graphs and the entries in our disutility matrix will be to enforce price-regulation and reverse ratio-amplification properties. We now describe the disutility matrix: We define only the disutility values in the matrix that are finite (the disutility of all agent-chore pair not mentioned should be assumed to be ∞). For all $k \in [K]$, for each pair of chores b_{2i-1}^k and b_{2i}^k, there are a set of agents that have finite disutility towards them and have infinite disutility towards all other chores; Additionally, these agents also happen to be either in A_k or A_{k-1} (indices are modulo K). We now outline these agents and their disutilities for every $k \in [K]$. To define the finite entries in the disutility matrix, we introduce the scalars $\frac{1}{n^{3c}} = \alpha_1, \alpha_2, \ldots, \alpha_K$ such that each $\alpha_{i+1} = \frac{3}{2} \cdot \alpha_i$ for all $i \in [K-1]$. Before we define the disutility matrix, we make an obvious claim about the scalars α_i for all $i \in [K]$, which will be useful later,

Claim 24. *We have $n^c \cdot \alpha_1 < \alpha_K \leq \frac{1}{n^c}$.*

Proof. We first show the lower bound. We have $\alpha_K = (\frac{3}{2})^{K-1} \cdot \alpha_1 = (\frac{3}{2})^{2c\lceil \log(n) \rceil - 1} \cdot \alpha_1 > 2^{c\log(n)} \cdot \alpha_1 = n^c \cdot \alpha_1$. Similarly, for the upper bound, we have, $\alpha_K = (\frac{3}{2})^{K-1} \cdot \alpha_1 = (\frac{3}{2})^{2c\lceil \log(n) \rceil - 1} \cdot \alpha_1 < 2^{2c\log(n)} \cdot \alpha_1 = n^{2c} \cdot \alpha_1 = \frac{1}{n^c}$ (as $\alpha_1 = \frac{1}{n^{3c}}$). $\qquad\square$

We now define the disutility matrix:

- $k = 1$: For each $i \in [n]$, we first define the disutilities of the agents that have finite disutility for chores b_{2i-1}^1 and b_{2i}^1. For each $i \in [n]$ we have,

$$\begin{aligned} d(a_{i',2i-1}^K, b_{2i-1}^1) &= (1 - \alpha_1) && \text{and} & d(a_{i',2i-1}^K, b_{2i}^1) &= (1 + \alpha_1) && \text{for all } i' \in [2n] \\ d(a_{i',2i}^K, b_{2i-1}^1) &= (1 + \alpha_1) && \text{and} & d(a_{i',2i}^K, b_{2i}^1) &= (1 - \alpha_1) && \text{for all } i' \in [2n] \\ d(a_{2i-1}', b_{2i-1}^1) &= (1 - \alpha_1) && \text{and} & d(a_{2i-1}', b_{2i}^1) &= (1 + \alpha_1) \\ d(a_{2i}', b_{2i-1}^1) &= (1 + \alpha_1) && \text{and} & d(a_{2i}', b_{2i}^1) &= (1 - \alpha_1). \end{aligned}$$

Therefore, for each $i \in [n]$, we have a component $\mathcal{D}_i^1$ in the disutility graph which is a complete bipartite graph comprising of agents $\{a_{i',2i-1}^K \mid i' \in [2n]\} \bigcup \{a_{i',2i}^K \mid i' \in [2n]\} \bigcup \{a_{2i-1}', a_{2i}'\}$ and chores $\{b_{2i-1}^1, b_{2i}^1\}$ (see Figure 2 (left subfigure) for an illustration).

- $2 \leq k \leq K$: For each $i \in [n]$ we have,

$$\begin{aligned} d(a_{2i-1}^{k-1}, b_{2i-1}^k) &= (1 - \alpha_k) && \text{and} & d(a_{2i-1}^{k-1}, b_{2i}^k) &= (1 + \alpha_k) \\ d(a_{2i}^{k-1}, b_{2i-1}^k) &= (1 + \alpha_k) && \text{and} & d(a_{2i}^{k-1}, b_{2i}^k) &= (1 - \alpha_k) \\ d(\overline{a}_i^k, b_{2i-1}^k) &= (1 - \alpha_k) && \text{and} & d(\overline{a}_i^k, b_{2i}^k) &= (1 - \alpha_k) \ . \end{aligned}$$

31

Therefore, for every k such that $2 \leq k \leq K$, for each $i \in [n]$, we have a connected component $\mathcal{D}_i^k$ in the disutility graph which is a complete bipartite graph comprising of agents $\left\{a_{2i-1}^{k-1}, a_{2i}^{k-1}, \bar{a}_i^k\right\}$ and chores $\left\{b_{2i-1}^k, b_{2i}^k\right\}$ (see Figure 2 (right subfigure) for an illustration).

It is clear that the disutility graph is a disjoint union of complete bipartite graphs, namely, the union of $\mathcal{D}_i^k$ for all $i \in [n]$ and $k \in [K]$. Therefore,

$$E(I) \text{ satisfies condition } SC_2 \text{ of Theorem 4.}$$
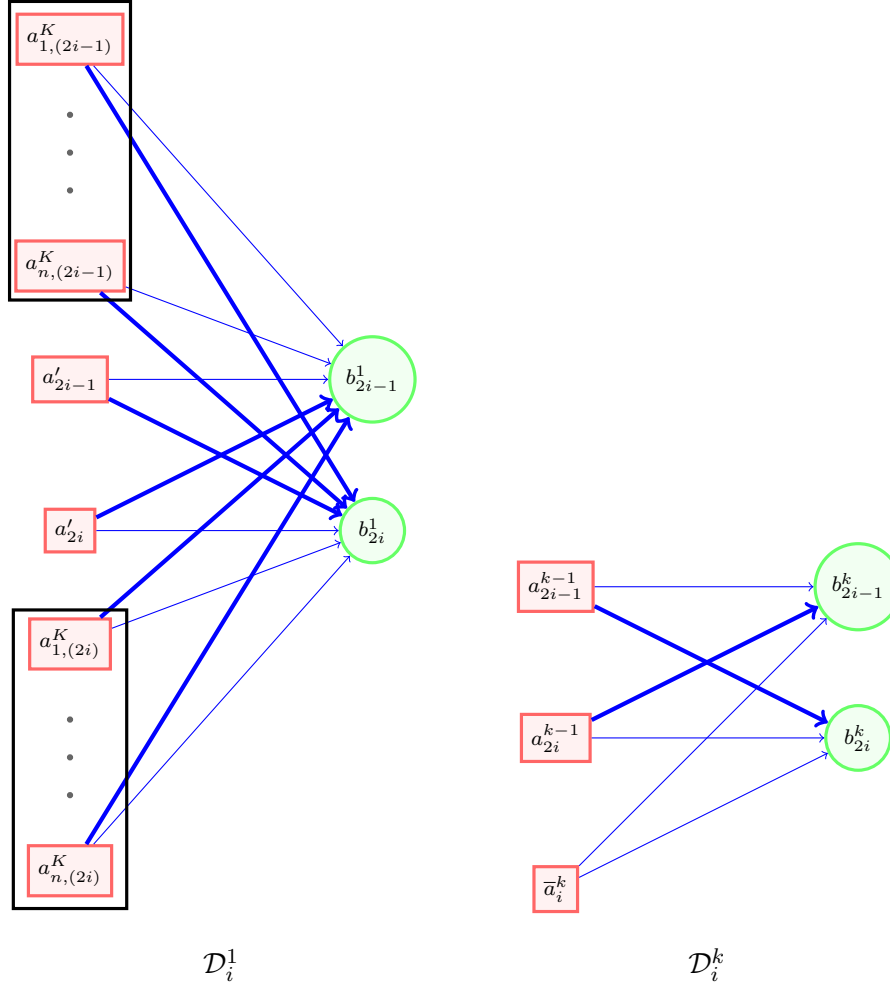
Figure 2: Illustration of the disutility graph corresponding to the disutility matrix: On the left, we have the component $\mathcal{D}_i^1$, and on the right we have $\mathcal{D}_i^k$ when $2 \leq k \leq K$. The edges are colored in order to also encode the disutility matrix. The thin blue edges from agents to chores depict a disutility of $(1 - \alpha_1)$ for $\mathcal{D}_i^1$ (left), and $(1 - \alpha_k)$ for $\mathcal{D}_i^k$ when $2 \leq k \leq K$ (right). Similarly, the thick blue edges from agents to chores depict a disutility of $(1 + \alpha_1)$ for $\mathcal{D}_i^1$ (left) and $(1 + \alpha_k)$ for $\mathcal{D}_i^k$ (right).

Endowment Matrix.   All agents in A_k have endowments of chores only in B_k for all $k \in [K]$. We only mention the non-zero agent-chore endowments (all agent-chore endowments, if not mentioned, are zero).

- $k = 1$: For each $i \in [2n]$ we have,

$$w(a_i^1, b_i^1) = n.$$

Also, for each $i \in [n]$ we have

$$w(a_{2i-1}', b_{2i-1}^1) = w(a_{2i-1}', b_{2i}^1) = \frac{1}{2} \cdot (1 - \alpha_K) \cdot (2n - \sum_{j \in [2n]} \mathbf{M}_{j,2i-1})$$

$$w(a_{2i}', b_{2i-1}^1) = w(a_{2i}', b_{2i}^1) = \frac{1}{2} \cdot (1 - \alpha_K) \cdot (2n - \sum_{j \in [2n]} \mathbf{M}_{j,2i}).$$

- $2 \leq k \leq K - 1$: For each $i \in [n]$, we have,

$$w(a_{2i-1}^k, b_{2i-1}^k) = n \qquad \text{and} \qquad w(a_{2i}^k, b_{2i}^k) = n$$
$$w(\overline{a}_i^k, b_{2i-1}^k) = \delta_k \qquad \text{and} \qquad w(\overline{a}_i^k, b_{2i}^k) = \delta_k \ ,$$

where $\delta_k = \frac{n \cdot \alpha_k}{2}$. The reason behind the exact choice of the value of δ_k will become explicit when we show that our instance satisfies the reverse ratio amplification property in Section 5.2. As of now, the reader is encouraged to think of it just as a small scalar.

- $k = K$: For each $i \in [n]$ we have,

$$w(a_{2i-1,j}^K, b_{2i-1}^K) = \mathbf{M}_{2i-1,j} \qquad \text{and} \qquad w(a_{2i,j}^K, b_{2i}^K) = \mathbf{M}_{2i,j} \qquad \text{for all } j \in [2n]$$
$$w(\overline{a}_i^K, b_{2i-1}^K) = \delta_K \qquad \text{and} \qquad w(\overline{a}_i^K, b_{2i}^K) = \delta_K \ ,$$

where $\delta_K = \frac{n \cdot \alpha_K}{2}$ (the reason behind the choice of value will become explicit in Section 5.2).

Strongly Connected Economy Graph. We now show that the economy graph G of our instance is strongly connected. For ease of explanation, we introduce the notion of *economy graph of components* $W = ([d], E_W)$, where there is an edge from $i \in [d]$ to $j \in [d]$, if and only if, there is an agent $a \in \mathcal{D}_i$ that has a positive endowment of some chore in $b \in \mathcal{D}_j$. We now make a claim that strong connectivity of W implies strong connectivity of the economy graph G.

Claim 25. *If W is strongly connected then G is also strongly connected.*

Proof. Consider any two agents a and a'. Let $a \in \mathcal{D}_i$ and $a' \in \mathcal{D}_j$.[19] Consider any chore b that agent a has a positive endowment of and let $\mathcal{D}_{i'}$ be the component in the disutility graph that contains b.[20] Then since $\mathcal{D}_{i'}$ is a biclique in our instance, every agent in $\mathcal{D}_{i'}$ has finite disutility for the chore b. Therefore, every agent in $\mathcal{D}_{i'}$ is reachable from a with an edge in the economy graph G. Now, since W is strongly connected, there is a path $\ell_1 \to \ell_2 \to \cdots \to \ell_k$ from $\ell_1 = i'$ to $\ell_k = j$. Let a_{ℓ_r} be the agent in the component $\mathcal{D}_{\ell_r}$, that has a positive endowment of some chore in the component $\mathcal{D}_{\ell_{r+1}}$ for all $r \in [k-1]$. Again, since each $\mathcal{D}_{\ell_r}$ is a biclique, every agent in $\mathcal{D}_{\ell_r}$ has a finite disutility for every chore in $\mathcal{D}_{\ell_r}$. Thus, there is an edge in the economy graph G from a_{ℓ_r} to every agent in $\mathcal{D}_{\ell_r}$, in particular there is an edge between a_{ℓ_r} and $a_{\ell_{r+1}}$ in G. Thus, we have a path $a \to a_{\ell_1} \to \cdots \to a_{\ell_{k-1}} \to a'$ in G. Therefore, if W is strongly connected, then there is a path between any two agents in G, implying that G is also strongly connected. $\square$

---

[19]Note that j could also be equal to i.
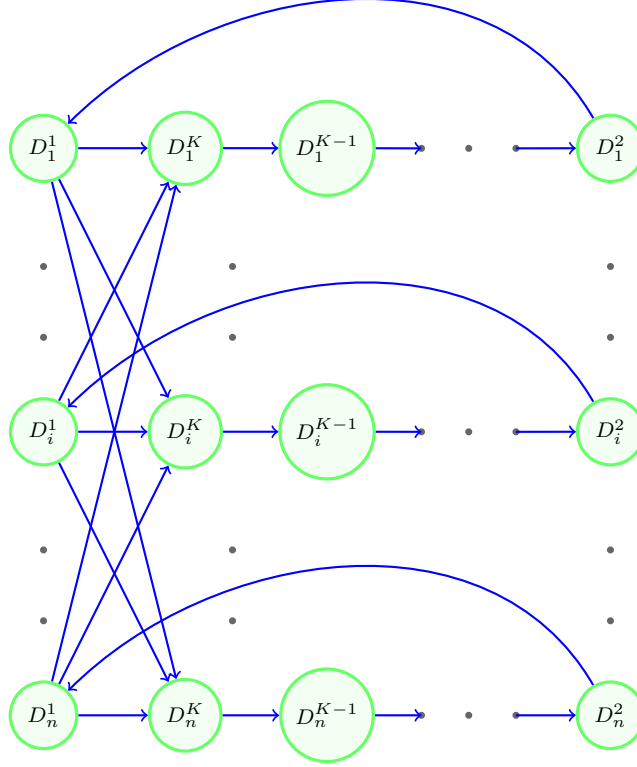[20]Again, i' could also be equal to i.

Figure 3: Illustration of the strong connectivity of the economy graph of components of our instance. Observe that all nodes are reachable from any $\mathcal{D}_i^1$ ($i \in [n]$). Also, from any arbitrary $\mathcal{D}_{i'}^{k'}$, the node $\mathcal{D}_{i'}^1$ is reachable and since every node is reachable from $\mathcal{D}_{i'}^1$, every node is also reachable from $\mathcal{D}_{i'}^{k'}$ as well. Therefore, the economy graph of components is strongly connected.

From here on, we show that W is strongly connected. Observe that the disutility graph consists of connected components $\mathcal{D}_i^k$ for $k \in [K]$ and $i \in [n]$. Also observe that every component $\mathcal{D}_i^k$ in the disutility graph comprises of exactly two chores b_{2i-1}^k and b_{2i}^k. Therefore, to show that there exists an edge from component $\mathcal{D}_{i'}^{k'}$ to $\mathcal{D}_i^k$ in W, it suffices to show that $\mathcal{D}_{i'}^{k'}$ contains agents that own parts of chores b_{2i-1}^k and b_{2i}^k. We now outline the edges in our exchange graph (see Figure 3):

- For all $i \in [n]$, and $2 \leq k \leq K$ there is an edge in W from $\mathcal{D}_i^k$ to $\mathcal{D}_i^{k-1}$: $\mathcal{D}_i^k$ contains the agents a_{2i-1}^{k-1} and a_{2i}^{k-1} that own parts of chores b_{2i-1}^{k-1} and b_{2i}^{k-1} respectively (see Figure 3).

- For all $i \in [n]$, there is an edge in W from $\mathcal{D}_i^1$ to D_j^K for all $j \in [n]$: Consider any $j \in [n]$. Observe that the component $\mathcal{D}_i^1$ contains the agents $a_{2j-1,2i}^K$ and $a_{2j,2i}^K$ and the agents $a_{2j-1,2i}^K$ and $a_{2j,2i}^K$ own parts of chores b_{2j-1}^K and b_{2j}^K respectively (see Figure 3).

Observe that all nodes are reachable from any $\mathcal{D}_i^1$ ($i \in [n]$). Also, from any arbitrary $\mathcal{D}_{i'}^{k'}$, the node $\mathcal{D}_{i'}^1$ is reachable and since every node is reachable from $\mathcal{D}_{i'}^1$, every node is also reachable from $\mathcal{D}_{i'}^{k'}$ as well. Therefore, the economy graph of components W, is strongly connected. Therefore, by Claim 25 we have that,

$$E(I) \text{ satisfies the condition } SC_1 \text{ of Theorem 4.}$$

34

Thus, $E(I)$ satisfies conditions of Theorem 4 and therefore admits a CE and thereby a $(1-\epsilon)$-approximate CE for any $\epsilon \geq 0$ (see Definition 3). Let $p(b_i^k)$ denote the price of chore b_i^k at an $(1-\epsilon)$-approximate CE for

$$\epsilon = \frac{\alpha_1}{200 \cdot n} < \frac{1}{n^{3c+1}} \Rightarrow \epsilon = O(1/n^{10}).$$

We now prove that our instance satisfies the required properties of *pairwise equal endowments, (approximate) price equality, (approximate) fixed earning, price regulation* and *reverse ratio amplification*.

5.2 $E(I)$ Satisfies All the Properties (Approximately)

Pairwise Equal Endowments. Here, we show that for all $i \in [n]$ and for all $k \in [K]$ the total endowment of b_{2i-1}^k equals the total endowment of b_{2i}^k and the total endowments of each chore in $E(I)$ is $\mathcal{O}(n)$.

Lemma 26. *For all $i \in [2n]$, the total endowments of chores b_{2i-1}^k and b_{2i}^k is*

1. *$n + n \cdot (1 - \alpha_K)$, if $k = 1$. In particular, a'_{2i-1} and a'_{2i} together, own $n \cdot (1 - \alpha_K)$ units of chores b_{2i-1}^k and b_{2i}^k each.*

2. *$n + \delta_k$, if $2 \leq k \leq K$.*

Proof. When $k = 1$, the only agents that have positive endowments of b_{2i}^1 are a_{2i}^1 (has an endowment of n) , a'_{2i} (has an endowment of $\frac{1}{2} \cdot (1 - \alpha_K) \cdot (2n - \sum_{j \in [2n]} \mathbf{M}_{j,2i})$) and a'_{2i-1}(has an endowment of $\frac{1}{2} \cdot (1 - \alpha_K) \cdot (2n - \sum_{j \in [2n]} \mathbf{M}_{j,2i-1})$). Therefore, the total endowment of b_{2i}^1 from the agents a'_{2i} and a'_{2i-1} is

$$= \frac{1}{2} \cdot (1 - \alpha_K) \cdot (2n - \sum_{j \in [2n]} \mathbf{M}_{j,2i}) + \frac{1}{2} \cdot (1 - \alpha_K) \cdot (2n - \sum_{j \in [2n]} \mathbf{M}_{j,2i-1})$$

$$= \frac{1}{2} \cdot (1 - \alpha_K) \cdot (4n - \sum_{j \in [2n]} (\mathbf{M}_{j,2i} + \mathbf{M}_{j,2i-1})).$$

Recall that $\mathbf{M}_{j,2i} + \mathbf{M}_{j,2i-1} = 1$. Therefore, the total endowment of b_{2i}^1 from the agents a'_{2i} and a'_{2i-1} is

$$= \frac{1}{2} \cdot (1 - \alpha_K) \cdot (4n - 2n)$$

$$= (1 - \alpha_K) \cdot n.$$

Therefore, the total endowment of chore b_{2i}^1 is $n + n \cdot (1 - \alpha_K)$. A similar argument will show that the total endowment of chore b_{2i-1}^1 is also $n + n \cdot (1 - \alpha_K)$ and that agents a'_{2i-1} and a'_{2i} *together*, own $n \cdot (1 - \alpha_K)$ units of it.

When $2 \leq k \leq K - 1$, the only agents that have positive endowments of b_{2i}^k are a_{2i}^k (has an endowment of n) and $\bar{a}_i^k$ (has an endowment of δ_k). Therefore, the total endowment is $n + \delta_k$. A similar argument will show that the total endowment of chore b_{2i-1}^k is also $n + \delta_k$.

When $k = K$, the only agents that have positive endowments of b_{2i}^K are the agents $a_{2i,j}^K$ (has an endowment of $\mathbf{M}_{2i,j}$) for all $j \in [2n]$ and the agent $\bar{a}_i^K$ (has an endowment of δ_K). Therefore, the

total endowment of chore b_{2i}^K is

$$= \sum_{j \in [2n]} \mathbf{M}_{2i,j} + \delta_K$$

$$= \sum_{j \in [n]} (\mathbf{M}_{2i,2j-1} + \mathbf{M}_{2i,2j}) + \delta_K$$

$$= \sum_{j \in [n]} 1 + \delta_K$$

$$= n + \delta_K$$

A similar argument will show that the total endowment of chore b_{2i-1}^K is also $n + \delta_K$. □

Price Equality. Here we will show that the sum of prices of chores b_{2i-1}^k and b_{2i}^k are almost same for all $i \in [n]$ and $k \in [K]$. Let us define

$$\pi_i^k = p(b_{2i-1}^k) + p(b_{2i}^k), \quad \forall i \in [n], k \in [K] .$$

Here on we will use the properties of approximate CE (Definition 3). Recall that, at an $(1-\epsilon)$-approximate CE, we have (i) complete allocation each chore, (ii) every agent only consumes her minimum pain-per-buck chores, and (iii) every agent i earns total price of her endowment up to $(1 \pm \epsilon)$ factor. And that, for two quantities x and y, by $x = (1 \pm \epsilon)y$ we mean $(1-\epsilon)y \le x \le (1+\epsilon)y$. In addition, now on

$$\text{by } x = (1 \pm \epsilon)^d y \text{ we mean } (1-\epsilon)^d y \le x \le (1+\epsilon)^d y.$$

Lemma 27. *For all $i, i' \in [n]$ and for all $k, k' \in [K]$, we have $\pi_{i'}^{k'} = (1 \pm O(n\epsilon))\pi_i^k$.*

Proof. Since $\epsilon < 1/n$ it suffices to show that $\pi_{i'}^{k'} = (1 + \pm \epsilon))^{O(n)}\pi_i^k$. We show this in two steps: First we show that for each $i \in [n]$, we have $\pi(k+1)_i = (1 \pm \epsilon)^2 \pi_i^k$ for all $k < K$, implying that $\pi_i^{k'} = (1 \pm \epsilon))^{O(n)}\pi_i^k$ for all $k, k' \in [K]$.

Then we show that $(1 - \epsilon)^2(\sum_{j \in [n]} \pi_j^K) \le n\pi_i^1 = (1 \pm \epsilon)^2(\sum_{j \in [n]} \pi_j^K)$ for all $i \in [n]$. This together with the above imply that $\pi_{i'}^1 \le (1 \pm \epsilon))^{O(n)}\pi_i^1$ for all $i, j \in [n]$. Putting these together proves the lemma.

We first show $\pi(k+1)_i \le (1\pm\epsilon)^2 \pi_i^k$ for all $i \le n, k < K$, wlog let $i = 1$. Observe that the agents $a_1^k, a_2^k, \bar{a}_1^{(k+1)}$ and chores $b_1^{(k+1)}, b_2^{(k+1)}$ form the connected component $\mathcal{D}_1^{(k+1)}$ in the disutility graph. That means these three agents earn all their money by consuming only these two chores, and no one else consumes these chores. The total money supply of these two chores is $(n + \delta_{(k+1)})\pi_1^{(k+1)}$.

As per Definition 3, at $(1 - \epsilon)$-approximate CE, an agents earn as much as the total price of their endowments up to $(1 \pm \epsilon)$ factor. Now since $\bar{a}_1^{(k+1)}$ owns $\delta_{(k+1)}$ units of both $b_1^{(k+1)}$ and $b_2^{(k+1)}$ only, her total cost is $\delta_{(k+1)}\pi_1^{(k+1)}$. Total price of chores owned by agents a_1^k and a_2^k is $n\pi_1^k$. At equilibrium demand should be equals supply, implying that,

$$(1 - \epsilon)(\delta_{(k+1)}\pi_1^{(k+1)} + n\pi_1^k) \le (n + \delta_{(k+1)})\pi_1^{(k+1)} \le (1 + \epsilon)(\delta_{(k+1)}\pi_1^{(k+1)} + n\pi_1^k)$$

$$\Rightarrow \quad (1 - \epsilon)n\pi_1^k - \epsilon\delta_{(k+1)}\pi_1^{(k+1)} \le n\pi_1^{(k+1)} \le (1 + \epsilon)n\pi_1^k + \epsilon\delta_{(k+1)}\pi_1^{(k+1)}$$

$$\Rightarrow \quad \tfrac{(1-\epsilon)}{(1+\epsilon\delta_{(k+1)}/n)}\pi_1^k \le \pi_1^{(k+1)} \le \tfrac{(1+\epsilon)}{(1-\epsilon\delta_{(k+1)}/n)}\pi_1^k$$

$$\Rightarrow \quad (1 - \epsilon)^2\pi_1^k \le \pi_1^{(k+1)} \le (1 + \epsilon)^2\pi_1^k$$

where the last implication follows from the fact that for $\epsilon < 1/n$ we have $(1 - \epsilon) < \frac{1}{(1+\epsilon\delta_{(k+1)}/n)}$ and $(1 + \epsilon) < \frac{1}{(1-\epsilon\delta_{(k+1)}/n)}$.

We now show that $n\pi_i^1 = (1\pm\epsilon)^2 \sum_{j\in[n]} \pi_j^K$: This time, we look into the connected component $\mathcal{D}_i^1$ of the disutility graph. We can claim that the agents $\{a_{j,2i-1}^K \mid j \in [2n]\}$, $\{a_{j,2i}^K \mid j \in [2n]\}$ and the agents a_{2i-1}' and a_{2i}' earn all of their money at an approximate CE from chores b_{2i-1}^1 and b_{2i}^1, and these are the only agents who consume these two chores. Total money supply of these two chores is $(n + n(1 - \alpha_K))\pi_i^1$ (by Lemma 26)

Observe that the total endowment of agents a_{2i-1}' and a_{2i}' is $n(1-\alpha_K)$ units of chores b_{2i-1}^1 and b_{2i}^1 only, and hence together they must earn $n(1-\alpha_K)\pi_i^1$ up to $(1\pm\epsilon)$. For agents $\{a_{j,2i-1}^K \mid j \in [2n]\}$ and $\{a_{j,2i}^K \mid j \in [2n]\}$, recall that each $a_{\ell,\ell'}^K$ owns $\mathbf{M}_{\ell,\ell'}$ units of b_ℓ^K. Therefore, together they need to earn the following amount up to $(1 \pm \epsilon)$ factor.

$$\sum_{j\in[2n]} \mathbf{M}_{j,2i} \cdot p(b_j^K) + \sum_{j\in[2n]} \mathbf{M}_{j,2i-1} \cdot p(b_j^K)$$
$$= \sum_{j\in[2n]} (\mathbf{M}_{j,2i} + \mathbf{M}_{j,2i-1}) \cdot p(b_j^K)$$
$$= \sum_{j\in[2n]} p(b_j^K) \quad (\text{using } \mathbf{M}_{j,2i-1} + \mathbf{M}_{j,2i} = 1)$$
$$= \sum_{j\in[n]} \pi_j^K$$

Again, equating supply with demand we get,

$$(1 - \epsilon)(n(1-\alpha_K)\pi_i^1 + \sum_{j\in[n]} \pi_j^K) \le (n + n(1 - \alpha_K))\pi_i^1 \le (1+\epsilon)(n(1-\alpha_K)\pi_i^1 + \sum_{j\in[n]} \pi_j^K)$$
$$\Rightarrow \quad (1 - \epsilon)\sum_{j\in[n]} \pi_j^K - \epsilon(n(1-\alpha_K)\pi_i^1 \le n\pi_i^1 \le (1+\epsilon)\sum_{j\in[n]} \pi_j^K + \epsilon(n(1-\alpha_K)\pi_i^1$$
$$\Rightarrow \quad \frac{(1-\epsilon)}{(1+\epsilon(1-\alpha_K))}\sum_{j\in[n]} \pi_j^K \le n\pi_i^1 \le \frac{(1+\epsilon)}{(1-\epsilon(1-\alpha_K))}\sum_{j\in[n]} \pi_j^K$$
$$\Rightarrow \quad (1 - \epsilon)^2 \sum_{j\in[n]} \pi_j^K < n\pi_i^1 < (1 + \epsilon)^2 \sum_{j\in[n]} \pi_j^K$$

where the last implication follows from the fact that for $\epsilon, \alpha_K < 1/n$ we have $(1-\epsilon) < \frac{1}{(1+\epsilon(1-\alpha_K))}$ and $(1 + \epsilon) > \frac{1}{(1-\epsilon(1-\alpha_K))}$. □

(Approximately) Fixed Earning. Here, we show that in every CE, the earning of each agent a_i' for $i \in [2n]$ is fixed up to $(1 \pm O(n)\epsilon)$ factor. Note that (approximate) CE prices are scale invariant.

Lemma 28. *Let* $\min_{i\in[n],k\le K} \pi_k^i = 2$. *Then for all* $i \in [2n]$, *we have that the earning of agent* a_i' *is* $(1 \pm \epsilon O(n))(1 - \alpha_K) \cdot (2n - \sum_{j\in[2n]} \mathbf{M}_{j,i})$.

Proof. Let $i = 2i'$. Then agent $a_{2i'}'$ owns $\frac{1}{2} \cdot (1 - \alpha_K) \cdot (2n - \sum_{j\in[2n]} \mathbf{M}_{j,2i'})$ units of both chores $b_{2i'-1}^1$ and $b_{2i'}^1$. Since the earning of any agent at an $(1 - \epsilon)$-approximate CE equals the sum of prices of chores she owns up to $(1 \pm \epsilon)$ factor, we have that the earning of agent $2i'$ is

$$= \frac{(1 \pm \epsilon)}{2} \cdot (1 - \alpha_K) \cdot (2n - \sum_{j\in[2n]} \mathbf{M}_{j,2i'}) \cdot (p(b_{2i'-1}^1) + p(b_{2i'}^1))$$

$$= \frac{(1 \pm \epsilon)}{2} \cdot (1 - \alpha_K) \cdot (2n - \sum_{j\in[2n]} \mathbf{M}_{j,2i'}) \cdot \pi_{i'}^1$$

$$= \frac{(1 \pm \epsilon)}{2} \cdot (1 - \alpha_K) \cdot (2n - \sum_{j\in[2n]} \mathbf{M}_{j,2i'}) \cdot 2(1 \pm O(n)\epsilon) \qquad \text{(by Lemma 27)}.$$

$$= (1 \pm O(n)\epsilon)(1 - \alpha_K) \cdot (2n - \sum_{j\in[2n]} \mathbf{M}_{j,2i'}).$$

37

Similarly, when $i = 2i' - 1$ we can show that the total earning of agent $a'_{2i'-1}$ is $(1 - \alpha_K) \cdot (2n - \sum_{j \in [2n]} \mathbf{M}_{j,2i'-1})$. Thus the total earning of any agent a'_i in a CE is $(1 \pm O(n)\epsilon)(1 - \alpha_K) \cdot (2n - \sum_{j \in [2n]} \mathbf{M}_{j,i})$. □

Price Regulation. Here, we show that for all $k \in [K]$ and $i \in [2n]$ the ratio of the prices of chores b^k_{2i-1} and b^k_{2i} is bounded.

Lemma 29. *For all $k \in [K]$ and for all $i \in [n]$, we have $\frac{1-\alpha_k}{1+\alpha_k} \leq \frac{p(b^k_{2i-1})}{p(b^k_{2i})} \leq \frac{1+\alpha_k}{1-\alpha_k}$.*

Proof. We prove the lower bound ($\frac{1-\alpha_k}{1+\alpha_k} \leq \frac{p(b^k_{2i-1})}{p(b^k_{2i})}$) by contradiction. The proof for the upper bound is symmetric. So assume that $\frac{1-\alpha_k}{1+\alpha_k} > \frac{p(b^k_{2i-1})}{p(b^k_{2i})}$. In that case, none of the agents in the connected component $\mathcal{D}^k_i$ will do any part of chore b^k_{2i-1} (as the disutility to price ratio of b^k_{2i-1} will be strictly more than that of b^k_{2i}). Since all the other agents have a disutility of ∞ for b^k_{2i-1}, it will remain unallocated. Therefore, the current prices for chores are not the prices corresponding to a CE, which is a contradiction. □

Reverse Ratio Amplification. Lastly, we show the property that when the price of chore b^k_i is at a limit, then the price of chore b^{k+1}_i is at the opposite limit, i.e., when $p(b^k_i) = 1 + \alpha_k$, then we have $p(b^{k+1}_i) = 1 - \alpha_{k+1}$ and similarly when $p(b^k_i) = 1 - \alpha_k$, then we have $p(b^{k+1}_i) = 1 + \alpha_{k+1}$.

Lemma 30. *For all $1 \leq k < K$ and $i \in [n]$, we have that,*

1. *if $\frac{p(b^k_{2i-1})}{p(b^k_{2i})} = \frac{1-\alpha_k}{1+\alpha_k}$, then $\frac{p(b^{k+1}_{2i-1})}{p(b^{k+1}_{2i})} = \frac{1+\alpha_{k+1}}{1-\alpha_{k+1}}$, and*

2. *if $\frac{p(b^k_{2i-1})}{p(b^k_{2i})} = \frac{1+\alpha_k}{1-\alpha_k}$, then $\frac{p(b^{k+1}_{2i-1})}{p(b^{k+1}_{2i})} = \frac{1-\alpha_{k+1}}{1+\alpha_{k+1}}$.*

Proof. We just show the proof of part 1. The proof for part 2 is symmetric. Let us assume that $\frac{p(b^k_{2i-1})}{p(b^k_{2i})} = \frac{1-\alpha_k}{1+\alpha_k}$, and that $p(b^k_{2i-1}) + p(b^k_{2i}) = 2$ (using scale invariance of CE prices). Then, $p(b^k_{2i-1}) = 1 - \alpha_k$ and $p(b^k_{2i}) = 1 + \alpha_k$. From the proof of Lemma 27, we have that $p(b^{k+1}_{2i-1}) + p(b^{k+1}_{2i}) = 2(1 \pm \epsilon)^2$ and by Lemma 29 we have $\frac{1-\alpha_{k+1}}{1+\alpha_{k+1}} \leq \frac{p(b^{k+1}_{2i-1})}{p(b^{k+1}_{2i})} \leq \frac{1+\alpha_{k+1}}{1-\alpha_{k+1}}$. Therefore, $(1 - \alpha_{k+1})(1 - \epsilon)^2 \leq p(b^{k+1}_{2i-1}), p(b^{k+1}_{2i}) \leq (1 + \alpha_{k+1})(1 + \epsilon)^2$.

Observe that agent a^k_{2i} owns n units of chore b^k_{2i} and has finite disutility only for the chores b^{k+1}_{2i-1} and b^{k+1}_{2i} (a^k_{2i} belongs in the connected component $\mathcal{D}^{k+1}_i$). Since at a $(1 - \epsilon)$-approximate CE, the total earning of agent a^k_{2i} equals the sum of prices of chores she owns up to $(1 \pm \epsilon)$ factor, we have that a^k_{2i} earns $(1 \pm \epsilon)n \cdot p(b^k_{2i}) = (1 \pm \epsilon)n(1 + \alpha_k)$ amount of money from chores b^{k+1}_{2i-1} and b^{k+1}_{2i}. We claim that it suffices to show that a^k_{2i} earns some of her money from the chore b^{k+1}_{2i-1}: This would immediately imply that $\frac{d(a^k_{2i}, b^{k+1}_{2i-1})}{p(b^{k+1}_{2i-1})} \leq \frac{d(a^k_{2i}, b^{k+1}_{2i})}{p(b^{k+1}_{2i})}$, further implying that $\frac{p(b^{k+1}_{2i-1})}{p(b^{k+1}_{2i})} \geq \frac{1+\alpha_{k+1}}{1-\alpha_{k+1}}$. However, by Lemma 29, we have that $\frac{p(b^{k+1}_{2i-1})}{p(b^{k+1}_{2i})} \leq \frac{1+\alpha_{k+1}}{1-\alpha_{k+1}}$, and thus we can conclude that $\frac{p(b^{k+1}_{2i-1})}{p(b^{k+1}_{2i})} = \frac{1+\alpha_{k+1}}{1-\alpha_{k+1}}$.

For the rest of the proof, we show that a^k_{2i} earns positive amount of money from chore b^{k+1}_{2i-1}. We prove this by contradiction. So let us assume that a^k_{2i} earns all her money of at least $(1-\epsilon)n \cdot (1+\alpha_k)$, only from chore b^{k+1}_{2i}. We will now show that the current prices of chores are not the prices corresponding to a CE by distinguishing between two cases. Recall that $\delta_k = \frac{n}{2}\alpha_k$ for all $k \in [K]$.

- $p(b_{2i}^{k+1}) > p(b_{2i-1}^{k+1})$: Observe that in this case, agent $\bar{a}_i^{k+1}$ will also earn all of her money from b_{2i}^{k+1} only (as the disutility to price ratio of b_{2i}^{k+1} is strictly smaller than that of b_{2i-1}^{k+1}). Therefore, we have that the total money agents a_{2i}^k and $\bar{a}_i^{k+1}$ earn from b_{2i}^{k+1} is,

$$\geq (1-\epsilon)(\pi_i^{k+1}\delta_{k+1} + n\cdot(1+\alpha_k))$$

$$\geq (1-\epsilon)(2(1-2\epsilon)\frac{n}{2}\alpha_{k+1} + n\cdot(1+\frac{2}{3}\cdot\alpha_{k+1})$$

$$(\text{as } \delta_{k+1} = \frac{n}{2}\alpha_{k+1},\ \alpha_{k+1} = \frac{3}{2}\cdot\alpha_k,\ \text{and } \pi_i^{k+1} \geq 2(1-2\epsilon))$$

$$= (1-\epsilon)\cdot(n(1+\alpha_{k+1}) + \frac{2n}{3}\alpha_{k+1} - 2\epsilon n\alpha_{k+1})$$

$$= \frac{(1-\epsilon)}{(1+3\epsilon)}(1+3\epsilon)[(n+\frac{n}{2}\alpha_{k+1})(1+\alpha_{k+1}) - \frac{n}{2}\alpha_{k+1}\frac{2n}{3}\alpha_{k+1} - 2\epsilon n\alpha_{k+1}]$$

$$= (1-\frac{4\epsilon}{(1+3\epsilon)})(1+3\epsilon)[(n+\delta_{k+1})(1+\alpha_{k+1}) - \frac{n}{2}\alpha_{k+1} + \frac{2n}{3}\alpha_{k+1} - 2\epsilon n\alpha_{k+1}]$$

$$> (1+3\epsilon)(n+\delta_{k+1})(1+\alpha_{k+1}) + (1+3\epsilon)n\alpha_{k+1}\left(\frac{2}{3} - \frac{1+\alpha_{k+1}}{2} - 2\epsilon\right) - 4\epsilon(4n)$$

$$(\text{as } [(n+\delta_{k+1})(1+\alpha_{k+1}) - \frac{n}{2}\alpha_{k+1} + \frac{2n}{3}\alpha_{k+1} - 2\epsilon n\alpha_{k+1}] < 4n)$$

$$> (1+\epsilon)^2(n+\delta_{k+1})(1+\alpha_{k+1}) + n(\alpha_{k+1}(2/3-1/2-1/n) - 16\epsilon)$$

$$> (1+\epsilon)^2(n+\delta_{k+1})(1+\alpha_{k+1})\quad (\text{as } \epsilon < \frac{\alpha_{k+1}}{192})$$

  However, total (money supply) price of b_{2i}^{k+1} is at most $(n+\delta_{k+1})p(b_{2i}^{k+1}) \leq (n+\delta_{k+1})(1+\epsilon)^2(1+\alpha_{k+1})$ since it's total endowment is $n+\delta_{k+1}$ by Lemma 26. This contradicts demand equals supply for chore b_{2i}^{k+1}.

- $p(b_{2i}^{k+1}) \leq p(b_{2i-1}^{k+1})$: In this case, $p(b_{2i}^{k+1}) \leq (1+\epsilon)^2$. Since the total endowment of b_{2i}^{k+1} is $n+\delta_{k+1}$ by Lemma 26, the total (money supply) price of chore b_{2i}^{k+1} is $(1+\epsilon)^2(n+\delta_{k+1})$. Next we show that this is strictly less than the demand from agent a_{2i}^k. Agent a_{2i}^k owns n units of b_{2i}^k and hence earns at least $(1-\epsilon)n(1+\alpha_k)$ at $(1-\epsilon)$-approximate CE.

$$(1-\epsilon)n(1+\alpha_k) = \frac{(1-\epsilon)}{(1+3\epsilon)}(1+3\epsilon)n(1+\frac{2}{3}\alpha_{k+1})$$

$$= (1-\frac{4\epsilon}{(1+3\epsilon)})(1+3\epsilon)(n+\frac{n}{2}\alpha_{k+1}+\frac{n}{6}\alpha_{k+1})$$

$$> (1+3\epsilon)(n+\delta_{k+1})+\frac{n}{6}\alpha_{k+1}) - 4\epsilon(2n)$$

$$> (1+\epsilon)^2(n+\delta_{k+1})\quad (\text{as } \epsilon < \frac{\alpha_{k+1}}{48})\qquad\square$$

Since K is even, a repeated application of Lemma 30 will yield the following lemma,

Lemma 31. *We have,*

1. *if* $\frac{p(b_{2i-1}^1)}{p(b_{2i}^1)} = \frac{1-\alpha_1}{1+\alpha_1}$, *then* $\frac{p(b_{2i-1}^K)}{p(b_{2i}^K)} = \frac{1+\alpha_K}{1-\alpha_K}$, *and*

2. *if* $\frac{p(b_{2i-1}^1)}{p(b_{2i}^1)} = \frac{1+\alpha_1}{1-\alpha_1}$, *then* $\frac{p(b_{2i-1}^K)}{p(b_{2i}^K)} = \frac{1-\alpha_K}{1+\alpha_K}$.

Now that we have shown that our instance satisfies the desired properties of pairwise equal endowments, (approximate) fixed earning, (approximate) price equality, price regulation and reverse

ratio amplification, we are ready to outline how to determine the equilibrium strategy vector x for the instance I of the polymatrix game, given the $(1-\epsilon)$-approximate CE prices of the instance $E(I)$ of chore division:

$$\forall i \in [n], \ x_{2i-1} = \frac{2p(b_{2i-1}^K) - (1-\alpha_K)\pi_i^K}{2 \cdot \pi_i^K \cdot \alpha_K} \qquad x_{2i} = \frac{2p(b_{2i}^K) - (1-\alpha_K)\pi_i^K}{2 \cdot \pi_i^K \cdot \alpha_K}$$

It is clear that given the prices of chores at a CE, the equilibrium strategy vector can be obtained in linear time. We will now show that x is the desired equilibrium strategy vector for instance I of the polymatrix game.

Lemma 32. $x = \langle x_1, x_2, \ldots, x_{2n} \rangle$ *is an equilibrium strategy vector for the polymatrix game instance* I.

Proof. First, observe that since $\pi_i^K = p(b_{2i-1}^K) + p(b_{2i}^K)$ and our instance satisfies price regulation (Lemma 29) we have that for all $i \in [n]$, $(1-\alpha_K)\frac{\pi_i^K}{2} \le p(b_{2i-1}^K), p(b_{2i}^K) \le (1+\alpha_K)\frac{\pi_i^K}{2}$. Therefore, for all $i \in [2n]$ $x_i \ge 0$. Furthermore, for all $i \in [n]$ we have $x_{2i-1} + x_{2i} = \frac{2p(b_{2i-1}^K)+p(b_{2i}^K)-2(1-\alpha_K)\pi_i^K}{2 \cdot \pi_i^K \cdot \alpha_K} = \frac{2\pi_i^K \alpha_K}{2\pi_i^K \alpha_K} = 1$.

Now we will show that if $x^T \cdot \mathbf{M}_{*,2i} > x^T \cdot \mathbf{M}_{*,2i-1} + \frac{1}{n}$, then $x_{2i-1} = 0$. The proof for the other symmetric condition will be similar. So let us assume that $x^T \cdot \mathbf{M}_{*,2i} > x^T \cdot \mathbf{M}_{*,2i-1} + \frac{1}{n}$. Note that if $p(b_{2i-1}^K) = (1-\alpha_K)\frac{\pi_i^K}{2}$, then $x_{2i-1} = 0$. Hence, by Lemma 31 it suffices to show that $\frac{p(b_{2i-1}^1)}{p(b_{2i}^1)} = \frac{1+\alpha_1}{1-\alpha_1}$. Wlog, assume that $\pi_i^1 = p(b_{2i-1}^1) + p(b_{2i}^1) = 2$, implying $(1-\alpha_1) \le p(b_{2i-1}^1), p(b_{2i}^1) \le (1+\alpha_1)$. Next we will show that if $\frac{p(b_{2i-1}^1)}{p(b_{2i}^1)} < \frac{1+\alpha_1}{1-\alpha_1}$, then $p(b_{2i}^1) > (1+\alpha_1)$, which is a contradiction. This will complete the proof.

By Lemma 27 we have $2(1 - O(n)\epsilon) \le \pi_i^K \le 2(1 + O(n)\epsilon)$ for all $i \in [n]$. This implies, $p(b_i^K) = (1 \pm O(n)\epsilon)(2x_i\alpha_K + (1-\alpha_K))$ for all $i \in [2n]$. Observe that the agents that have a disutility of $1-\alpha_1$ towards chore b_{2i}^K are $\left\{ a_{j,2i}^K \mid j \in [2n] \right\} \cup a_{2i}'$. Observe that at the given prices the sum of prices of chores owned by these agents is,

$$= \sum_{j\in[2n]} \mathbf{M}_{j,2i} \cdot p(b_j^K) + (1-\alpha_K) \cdot (2n - \sum_{j\in[2n]} \mathbf{M}_{j,2i}) \qquad \text{(by Lemma 28)}$$

$$\ge \sum_{j\in[2n]} \mathbf{M}_{j,2i} \cdot (1 - O(n)\epsilon)(2\alpha_K \cdot x_j + (1-\alpha_K)) + (1-\alpha_K) \cdot (2n - \sum_{j\in[2n]} \mathbf{M}_{j,2i}) \qquad \text{(substituting } p(b_j^K))$$

$$= (1 - O(n)\epsilon) \sum_{j\in[2n]} 2\alpha_K \cdot x_j \cdot \mathbf{M}_{j,2i} + (1-\alpha_K)(1 - O(n)\epsilon) \cdot \sum_{j\in[2n]} \mathbf{M}_{j,2i} + (1-\alpha_K) \cdot (2n - \sum_{j\in[2n]} \mathbf{M}_{j,2i})$$

$$\ge (1 - O(n)\epsilon)2\alpha_K x^T \cdot \mathbf{M}_{*,2i} + 2n \cdot (1-\alpha_K) - \epsilon O(n^2).$$

Similarly, the total earning of the agents that have a disutility of $1-\alpha_1$ towards b_{2i-1}^1 is *at most* $(1+O(n)\epsilon)2\alpha_K x^T \cdot \mathbf{M}_{*,2i-1} + 2n \cdot (1-\alpha_K) + \epsilon O(n^2)$. Observe that the agents with disutility $1-\alpha_1$ towards b_{2i}^1 can earn all of their money only from the chores b_{2i}^1 or b_{2i-1}^1 (as these are the only chores towards which they have finite disutility). Also note that both chores b_{2i-1}^1 and b_{2i}^1 have the same total endowment which is $n + n \cdot (1 - \alpha_K)$ by Lemma 26(part 1). Now if, $\frac{p(b_{2i-1}^1)}{p(b_{2i}^1)} < \frac{1+\alpha_1}{1-\alpha_1}$, then agents with disutility $1-\alpha_1$ towards b_{2i}^1 earn all of their money entirely from b_{2i}^1 (they earn nothing from b_{2i-1}^1). Since at $(1-\epsilon)$-approximate CE they earn at least $(1-\epsilon)$ times price of their

endowments, we have

$$p(b_{2i}^1) \geq (1 - \epsilon) \frac{2(1 - O(n)\epsilon)\alpha_K x^T \cdot \mathbf{M}_{*,2i} + 2n \cdot (1 - \alpha_K) - \epsilon O(n^2)}{n + n \cdot (1 - \alpha_K)} \tag{9}$$

Agents with disutility $1 - \alpha_1$ towards b_{2i-1}^1 may earn from both b_{2i}^1 and b_{2i-1}^1. Since they earn at most $(1 + \epsilon)$ times price of their endowments, we have

$$p(b_{2i-1}^1) \leq (1 + \epsilon) \frac{2(1 + O(n)\epsilon)\alpha_K x^T \cdot \mathbf{M}_{*,2i-1} + 2n \cdot (1 - \alpha_K) + \epsilon O(n^2)}{n + n \cdot (1 - \alpha_K)} \tag{10}$$

Using the above two inequalities together with $x^T \cdot \mathbf{M}_{*,2i} > x^T \cdot \mathbf{M}_{*,2i-1} + \frac{1}{n}$ we will next show that $p(b_{2i}^1) > (1 + \alpha_1)$.

$$(n + n \cdot (1 - \alpha_K))p(b_{2i}^1) \geq (1 - \epsilon)\left[2(1 - O(n)\epsilon)\alpha_K x^T \cdot \mathbf{M}_{*,2i} + 2n \cdot (1 - \alpha_K) - \epsilon O(n^2)\right]$$

$$> (1 - \epsilon)\left[2(1 - O(n)\epsilon)\alpha_K x^T \cdot \mathbf{M}_{*,2i-1} + \frac{2(1 - O(n)\epsilon)\alpha_K}{n} + 2n(1 - \alpha_K) - \epsilon O(n^2)\right]$$

$$> (1 - \epsilon)(1 - O(n)\epsilon)^2 \left[(1 - \epsilon)(n + n(1 - \alpha_K))p(b_{2i-1}^1) - 2n(1 - \alpha_K) - \epsilon O(n^2) + \frac{2\alpha_K}{n}\right.$$

$$\left. + 2n(1 - \alpha_K) - \frac{\epsilon O(n^2)}{(1 - O(n)\epsilon)^2}\right]$$

$$\text{(Using (10) and } \tfrac{1}{(1+x)} > (1 - x))$$

$$> (1 - \epsilon)^2(1 - O(n)\epsilon)\left[(n + n(1 - \alpha_K))(1 - \alpha_1) + \frac{2\alpha_K}{n} - \epsilon O(n^2)(1 + O(n)\epsilon)\right]$$

$$\text{(Using } p(b_{2i-1}^1) > (1 - \alpha_1) \text{ and } \tfrac{1}{(1-x)} < (1 + 2x))$$

$$\geq (1 - O(n)\epsilon)\left[(n + n(1 - \alpha_K))(1 - \alpha_1 + \frac{\alpha_K}{n^2}\right] + (1 - O(n)\epsilon)\left[\frac{\alpha_K}{n} - \epsilon O(n^2)(1 + O(n)\epsilon)\right]$$

$$> (1 - O(n)\epsilon)(n + n(1 - \alpha_K)(1 - \alpha_1 + n\alpha_1) \quad \text{(as } \epsilon < \frac{\alpha_1}{4} \leq \frac{\alpha_K}{4n^3})$$

$$\geq (n + n(1 - \alpha_K))(1 + \alpha_1) + (n + n(1 - \alpha_K))(1 + (n - 2)\alpha_1)$$

$$- O(n)\epsilon(n + n(1 - \alpha_K)(1 + (n - 1)\alpha_1)$$

$$> (n + n(1 - \alpha_K))(1 + \alpha_1)$$

The above implies $p(b_{2i}^1) > (1 + \alpha_1)$ as aimed, which is a contradiction to $p(b_{2i}^1) \leq (1 + \alpha_1)$. Therefore, agents with $(1 - \alpha_1)$ disutility towards b_{2i}^1 must be consuming both b_{2i}^1 and b_{2i-1}^1. This is possible only if $\frac{p(b_{2i-1}^1)}{p(b_{2i}^1)} = \frac{1+\alpha_1}{1-\alpha_1}$, implying $\frac{p(b_{2i-1}^K)}{p(b_{2i}^K)} = \frac{1-\alpha_K}{1+\alpha_K}$ by Lemma 31 and thereby $p(b_{2i-1}^K) = (1 - \alpha_K)\frac{\pi_i^K}{2}$. Replacing this in the expression for x_{2i-1} we get,

$$x_{2i-1} = \frac{2p(b_{2i}^K) - (1 - \alpha_K)\pi_i^K}{2 \cdot \pi_i^K \cdot \alpha_K}$$

$$= \frac{\pi_i^K(1 - \alpha_K) - (1 - \alpha_K)\pi_i^K}{2 \cdot \alpha_K}$$

$$= 0.$$

A very similar argument will show that when $x^T \cdot \mathbf{M}_{*,2i-1} > x^T \cdot \mathbf{M}_{*,2i} + \frac{1}{n}$, then $x_{2i} = 0$. Thus, $x = \langle x_1, x_2, \ldots, x_n \rangle$ is a desired Nash equilibrium strategy vector for the polymatrix game I. $\quad\square$

Note that, since $\epsilon = \frac{\alpha_1}{200n}$, we have $\epsilon = \frac{1}{200n^{3c+1}} = \frac{1}{poly(n)}$. Thus, this immediately implies the main result of this section, where the sufficiency conditions to guarantee existence of CE are of Theorem 4.

Theorem 33. *Finding* $(1 - \frac{1}{poly(n)})$ *approximate CE for chore division in the exchange model is PPAD-hard even for instances satisfying conditions* SC_1 *and* SC_2 *of Theorem 4.*

Proof. We bring all the points together. Normalized polymatrix game is PPAD-hard [CPY17]. Given an instance I of the normalized polymatrix game, in polynomial time we can determine the instance $E(I)$. $E(I)$ satisfies the sufficiency conditions SC_1 and SC_2 of Theorem 4 and therefore admits an exact as well as approximate CE. Given the prices at a $(1 - \frac{1}{poly(n)})$-approximate CE of $E(I)$, in polynomial time we can determine the equilibrium strategy vector for the polymatrix game by Lemma 32. Therefore, chore division is PPAD-hard even on instances that satisfy SC_1 and SC_2 of Theorem 4 sufficient to guarantee existence of CE. $\qquad\square$

References

[BCM22]   Shant Boodaghians, Bhaskar Ray Chaudhury, and Ruta Mehta. Polynomial time algorithms to find an approximate competitive equilibrium for chores. In *SODA*, pages 2285–2302. SIAM, 2022.

[BKV18]   Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Finding fair and efficient allocations. In *Proc. 19th Conf. Economics and Computation (EC)*, pages 557–574, 2018.

[BMSY17]   Anna Bogomolnaia, Hervé Moulin, Fedor Sandomirskiy, and Elena Yanovskaia. Competitive division of a mixed manna. *Econometrica*, 85(6):1847–1871, 2017.

[BS19]   Simina Branzei and Fedor Sandomirskiy. Algorithms for competitive division of chores. arXiv:1907.01766 (To appear in Mathematics of Operations Research), 2019.

[CCPY22]   Thomas Chen, Xi Chen, Binghui Peng, and Mihalis Yannakakis. Computational hardness of the Hylland-Zeckhauser scheme. In *Proc. 33rd Symp. Discrete Algorithms (SODA)*, 2022. To appear.

[CDDT09]   Xi Chen, Decheng Dai, Ye Du, and Shang-Hua Teng. Settling the complexity of Arrow-Debreu equilibria in markets with additively separable utilities. In *Proc. 50th Symp. Foundations of Computer Science (FOCS)*, pages 273–282, 2009.

[CDG+17]   Richard Cole, Nikhil Devanur, Vasilis Gkatzelis, Kamal Jain, Tung Mai, Vijay Vazirani, and Sadra Yazdanbod. Convex program duality, Fisher markets, and Nash social welfare. In *Proc. 18th Conf. Economics and Computation (EC)*, 2017.

[CGMM21]   Bhaskar Ray Chaudhury, Jugal Garg, Peter McGlaughlin, and Ruta Mehta. Competitive allocation of a mixed manna. In *Proc. 32nd Symp. Discrete Algorithms (SODA)*, 2021.

[CGMM22]   Bhaskar Ray Chaudhury, Jugal Garg, Peter McGlaughlin, and Ruta Mehta. On the existence of competitive equilibrium with chores. In *ITCS*, volume 215 of *LIPIcs*, pages 41:1–41:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[CM18]   Bhaskar Ray Chaudhury and Kurt Mehlhorn. Combinatorial algorithms for general linear Arrow-Debreu markets. In *Proc. 38th Conf. Foundations of Software Tech. and Theoret. Comp. Sci. (FSTTCS)*, volume 122, pages 26:1–26:16, 2018.

[CMPV05] Bruno Codenotti, Benton McCune, Sriram Penumatcha, and Kasturi R. Varadarajan. Market equilibrium for CES exchange economies: Existence, multiplicity, and computation. In *Proc. 25th Conf. Foundations of Software Tech. and Theoret. Comp. Sci. (FSTTCS)*, pages 505–516, 2005.

[Cor89] Bernard Cornet. Linear exchange economies. Technical report, Cahier Eco-Math, Université de Paris, 1989.

[CPY17] Xi Chen, Dimitris Paparas, and Mihalis Yannakakis. The complexity of non-monotone markets. *Journal of the ACM (JACM)*, 64(3):1–56, 2017.

[CSVY06] Bruno Codenotti, Amin Saberi, Kasturi Varadarajan, and Yinyu Ye. Leontief economies encode two-player zero-sum games. In *Proc. 17th Symp. Discrete Algorithms (SODA)*, pages 659–667, 2006.

[CT09] Xi Chen and Shang-Hua Teng. Spending is not easier than trading: On the computational equivalence of Fisher and Arrow-Debreu equilibria. In *Proc. 20th Intl. Symp. Algorithms and Computation (ISAAC)*, pages 647–656, 2009.

[DGM16] Ran Duan, Jugal Garg, and Kurt Mehlhorn. An improved combinatorial polynomial algorithm for the linear Arrow-Debreu market. In *Proc. 27th Symp. Discrete Algorithms (SODA)*, pages 90–106, 2016.

[DGV16] Nikhil Devanur, Jugal Garg, and László Végh. A rational convex program for linear Arrow-Debreu markets. *ACM Trans. Econom. Comput.*, 5(1):6:1–6:13, 2016.

[DM15] Ran Duan and Kurt Mehlhorn. A combinatorial polynomial algorithm for the linear Arrow-Debreu market. *Inf. Comput.*, 243:112–132, 2015.

[DPSV08] Nikhil Devanur, Christos Papadimitriou, Amin Saberi, and Vijay Vazirani. Market equilibrium via a primal–dual algorithm for a convex program. *J. ACM*, 55(5), 2008.

[Eav76] B. Curtis Eaves. A finite algorithm for the linear exchange model. *J. Math. Econom.*, 3:197–203, 1976.

[EG59] Edmund Eisenberg and David Gale. Consensus of subjective probabilities: The Pari-Mutuel method. *Ann. Math. Stat.*, 30(1):165–168, 1959.

[Eis61] Edmund Eisenberg. Aggregation of utility functions. *Management Sci.*, 7(4):337–350, 1961.

[EPS22] Soroush Ebadian, Dominik Peters, and Nisarg Shah. How to fairly allocate easy and difficult chores. In *Proc. 21st Conf. Auton. Agents and Multi-Agent Systems (AAMAS)*, 2022. To appear.

[FGHS21] John Fearnley, Paul W. Goldberg, Alexandros Hollender, and Rahul Savani. The complexity of gradient descent: CLS = PPAD ∩ PLS. In *Proc. 53rd Symp. Theory of Computing (STOC)*, pages 46–59, 2021.

[Fuj80] Satoru Fujishige. Lexicographically optimal base of a polymatroid with respect to a weight vector. *Mathematics of Operations Research*, 5(2):186–196, 1980.

[Gal76] David Gale. The linear exchange model. *Journal of Mathematical Economics*, 3(2):205–209, l976.

[GHV21]   Jugal Garg, Edin Husić, and László Végh. Auction algorithms for market equilibrium with weak gross substitute demands. In *Proc. 38th Symp. Theoret. Aspects of Computer Science (STACS)*, volume 187, pages 33:1–33:19, 2021.

[GL91]   Donald Goldfarb and Shucheng Liu. An $O(n^3L)$ primal interior point algorithm for convex quadratic programming. *Math. Program.*, 49:325–340, 1991.

[GM20]   Jugal Garg and Peter McGlaughlin. Computing competitive equilibria with mixed manna. In *Proc. 19th Conf. Auton. Agents and Multi-Agent Systems (AAMAS)*, pages 420–428, 2020.

[GMQ22]   Jugal Garg, Aniket Murhekar, and John Qin. Fair and efficient allocations of chores under bivalued preferences. In *Proc. 36th Conf. Artif. Intell. (AAAI)*, 2022. To appear.

[GMVY17]   Jugal Garg, Ruta Mehta, Vijay V. Vazirani, and Sadra Yazdanbod. Settling the complexity of Leontief and PLC exchange markets under exact and approximate equilibria. In *STOC*, pages 890–901, 2017.

[GP14]   Jonathan R. Goldman and Ariel D. Procaccia. Spliddit: unleashing fair division algorithms. *SIGecom Exchanges*, 13(2):41–46, 2014.

[GV19]   Jugal Garg and László A Végh. A strongly polynomial algorithm for linear exchange markets. In *Proc. 51st Symp. Theory of Computing (STOC)*, 2019.

[HZ79]   Aanund Hylland and Richard Zeckhauser. The efficient allocation of individuals to positions. *J. Political Economy*, 87(2):293–314, 1979.

[Jai07]   Kamal Jain. A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities. *SIAM J. Comput.*, 37(1):306–318, 2007.

[NP83]   E I Nenakov and M E Primak. One algorithm for finding solutions of the Arrow-Debreu model. *Kibernetica*, 3:127–128, 1983.

[Orl10]   James Orlin. Improved algorithms for computing Fisher's market clearing prices. In *Proc. 42nd Symp. Theory of Computing (STOC)*, pages 291–300, 2010.

[Shm09]   Vadim Shmyrev. An algorithm for finding equilibrium in the linear exchange model with fixed budgets. *J. Appl. Indust. Math.*, 3(4):505–518, 2009.

[Vég12]   László Végh. Strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives. In *Proc. 44th Symp. Theory of Computing (STOC)*, pages 27–40, 2012.

[Ye08]   Yinyu Ye. A path to the Arrow-Debreu competitive market equilibrium. *Math. Prog.*, 111(1-2):315–348, 2008.