

Fairness in Repeated Matching: A Maximin Perspective

Eugene Lim¹, Tzeh Yuan Neoh² and Nicholas Teh³

¹National University of Singapore, Singapore

²Harvard University, USA

³University of Oxford, UK

Abstract

We study a sequential decision-making model where a set of items is repeatedly matched to the same set of agents over multiple rounds. The objective is to determine a sequence of matchings that either maximizes the utility of the least advantaged agent at the end of all rounds (optimal) or at the end of every individual round (anytime optimal). We investigate the computational challenges associated with finding (anytime) optimal outcomes and demonstrate that these problems are generally computationally intractable. However, we provide approximation algorithms, fixed-parameter tractable algorithms, and identify several special cases whereby the problem(s) can be solved efficiently. Along the way, we also establish characterizations of Pareto-optimal/maximum matchings, which may be of independent interest to works in matching theory and house allocation.

1 Introduction

Traditional machine learning (ML) algorithms often focus on global objectives such as efficiency (e.g., maximizing accuracy or minimizing error rates in decision-making systems) or maximizing revenue/profit (e.g., maximizing click-through rates for recommendation systems), as they align closely with organizational goals and are more straightforward to quantify and optimize. However, modern approaches increasingly emphasize *fairness* as a key desideratum, as societal and regulatory demands push for more equitable and responsible ML systems.

We consider a multi-agent sequential decision-making scenario where a set of resources must be allocated among agents repeatedly over time, with the objective of achieving fairness in the assignment process. This framework encompasses applications such as dynamic spectrum allocation in wireless networks and energy distribution in smart grids [Elhachmi, 2022, Jain et al., 2022, Rony et al., 2021, Soares et al., 2024]. In the case of spectrum allocation, communication channels must be repeatedly assigned to devices, with each device requiring exclusive access to one channel in each time slot. Persistent disparities in access can degrade system efficiency, reduce user satisfaction, and undermine trust. Similarly, in many other ML-driven resource allocation systems, disparities in the distribution of resources—such as GPUs in distributed computing—can lead to unfair outcomes that compromise the perceived and actual effectiveness of the system. Numerous other applications where decisions are made dynamically—such as assigning tasks to workers in crowdsourcing platforms [Moayedikia et al., 2020], or distributing compute resources in cloud systems [Belgacem, 2022, Gupta et al., 2017, Saraswathi et al., 2015]—call for central decision-makers to ensure that no agent is persistently disadvantaged, which is critical for both fairness and long-term trust in the system.

The scenarios described above can be captured using the *repeated matching* framework—a multi-agent sequential decision-making model in which a set of goods is repeatedly matched to agents over time, and each agent is assigned exactly one good at each round. This can also be viewed as a multi-round generalization of the *bottleneck assignment problem* [Ford and Fulkerson, 1962] which is well-known in multi-agent task allocation: an application of this problem arises in *threat seduction*, where

decoys are assigned to multiple incoming threats [Shames et al., 2017]. Our problem can also be viewed as a sequential variant of the *Santa Claus problem* [Bansal and Sviridenko, 2006], which is closely related to the classic scheduling problem of *makespan minimization* on unrelated parallel machines [Lenstra et al., 1990, Bamas et al., 2024].

In particular, we focus on the *maximin* (or *egalitarian*) objective [Demko and Hill, 1988, Thomson, 1983], which aims to find a sequence of matchings that maximizes the minimum utility among agents. Maximin fairness serves as a principled trade-off between fairness and efficiency, as minimizing disparities often enhances overall system robustness and user satisfaction. Moreover, modern ML systems often involve iterative, data-driven decision-making, and maximin fairness integrates naturally with these systems by providing a fairness criterion that adapts dynamically, with its ability to handle both short-term and long-term outcomes.¹

1.1 Our Results

We study the *repeated matching* problem from the perspective of *maximin* (or *egalitarian*) fairness, a principle grounded in game theory, fair division, and matching problems. Leveraging techniques from classical matching algorithms, approximation methods, dynamic programming, and online decision-making, we analyze how to design fair repeated matching policies that ensure long-term fairness across multiple rounds.

In Section 2, we formally define the repeated matching problem and introduce the notion of (*anytime*) *optimality* in the egalitarian sense. We also introduce several tools that is central in proving some of our results.

In Section 3, we study the computation of optimal solutions. We begin by defining the decision variant of our matching problem and showing that it is NP-hard in general. Notably, this hardness holds even with only two timesteps and ternary agent valuations (i.e., when each agent’s utility for a good takes one of three possible values). Given these hardness results, we turn to the optimization variant of the problem and develop approximation algorithms that achieve an additive approximation bound independent of the number of rounds T . Crucially, this implies that as T increases—a scenario common in real-world applications—the solution produced by our algorithm converges to the optimal one. In addition, we also show that the problem is fixed-parameter tractable (FPT) with respect to the number of agents by providing a polynomial-time algorithm when the number of agents is a constant. Notably, in the process, we derive a characterization of *Pareto optimal* matchings in terms of the permutations of agents. This generalizes the previously-known result that *serial dictatorship* characterizes Pareto optimal matchings and may be of independent interest to communities working on the *house allocation* problem.

In Section 4, we shift our focus to *anytime optimal* solutions. We show that such solutions always exist for two agents, and we provide a polynomial-time algorithm for it. However, this does not extend to three or more agents—even with just two rounds, deciding if an instance admits an anytime optimal solution becomes coNP-hard. Nevertheless, we design an approximation algorithm that achieves anytime optimality with an additive bound independent of T . These results underscore the inherent difficulty of achieving anytime optimality in our setting.

In Section 5, we revisit optimality and identify three special cases admitting polynomial-time algorithms: (i) agents with binary valuations, (ii) two types of goods, and (iii) identical agent valuations. These special cases are well-motivated by the (temporal) fair division literature. For (i), we present an exact algorithm and a new characterization of Pareto optimal matchings under binary valuations. For (ii), we similarly provide an efficient exact algorithm. For (iii), despite NP-hardness in general, we show

¹This is in contrast to other *comparative* notions of fairness, such as *envy-freeness*, which has also been studied in the static matching [Aigner-Horev and Segal-Halevi, 2022, Wu and Roth, 2018, Yokoi, 2020] and the two-sided repeated matching [Gollapudi et al., 2020] setting. Maximin fairness is also more *demonstrably* fair compared to an envy-based approach.

that optimal solutions can be computed in polynomial time when the number of rounds is a multiple of the number of agents. Finally, we extend our approximation approach to anytime optimality in these cases, giving us a stronger results than in the general setting.

1.2 Related Work

We highlight several streams of research that are related to our work. We note that while there are many works on *online* matching and fair division, they are not directly relevant to our setting, as the underlying assumptions differ fundamentally. In our setting, the entire set of goods is made available in every round, whereas in online models, the set of goods may vary over time. Thus, we focus only on discussing works where meaningful implications can be drawn between their results and ours.

Repeated matching. Repeated matching was first studied by Hosseini et al. [2015], which considered ordinal preferences that could change over time. They study strategyproofness and approximate envy-freeness. However, ordinal (their model) and cardinal (our model) preferences are vastly different, both in techniques and results. Gollapudi et al. [2020] subsequently looked at a two-sided repeated matching problem (i.e., each side have preferences over the other side). They also study approximate envy-freeness as the key desiderata, albeit under some strong assumptions. In contrast, our model is on one-sided repeated matching, which is fundamentally different. Our model is most aligned with that of Caragiannis and Narang [2024]. However, they consider a slightly more general variant, whereby the value of an agent for a good in some round depends on the number of rounds in which the good has been given to the agent in the past. They study approximately envy-free notions, show an intractability result, and special cases where fairness can be guaranteed. Our model, while more specialized than theirs, has a few distinctions: (i) we have stronger negative and intractability results, (ii) the fairness concept we consider is not envy-based, and is therefore novel in this domain, and (iii) we consider a notion of fairness *at every round prefix*, something with prior work does not consider—they look at fairness at the end. Recently, Micheel and Wilczynski [2024] also studied essentially the same model (under a different name: *repeated house allocation*), but with ordinal preferences and other kinds of envy-based measures.

Repeated fair division. Igarashi et al. [2024] studied a model of repeated fair division, where a set of goods is available at each round, and every good must be allocated. This is in contrast to our model where each agent gets exactly one good. They consider the compatibility of envy-freeness and Pareto optimality, and show positive results in restricted cases. Balan et al. [2011] study a similar model, but with a focus on the *average* utility of goods received by the agents. Note that as with classical fair division, house allocation (where each agent gets exactly one good) is a special case and has considerably different results. Elkind et al. [2025a] also consider a non-repeated (but also offline) variant of this model where a single good needs to be allocate at each round.

Multi-agent sequential decision-making. Several other works in multi-agent systems bear resemblance to our model. For instance, Zhang and Shah [2014] also study the egalitarian objective multi-agent decision-making problems. However, they take a non-cooperative game-theoretic approach and do not study a matching problem. Lim et al. [2024] consider an assignment problem in the context of stochastic multi-armed bandits, with egalitarian fairness as the objective. In their setting, at each round, exactly one “arm” must be assigned to each user such that no two users are assigned to the same arm. However, the user’s utility (“reward”) in this case is stochastic, and therefore explores a different problem. Several other works [Cheng et al., 2005, Kellerer et al., 1997] consider the problem of semi-online multiprocessor scheduling, with the objective of minimizing the *makespan* (i.e., minimize the maximum time taken by any any processor). This is analogous to the egalitarian objective. However, results in this setting only hold for identical valuations (since machines are identical), and primarily apply to a

(semi-)online setting, where goods arrive one at a time (and so valuations over future goods are known not in advance), but the total valuation is known.

Santa Claus problem. Another related line of work is egalitarian fair division, also known as the *Santa Claus* problem. The standard model here is a single-shot fair division setting with an egalitarian objective, which was studied as far back as Thomson [1983], who axiomatically characterized the egalitarian solution using numerous desirable properties. Bansal and Sviridenko [2006] then initiated the study of approximation algorithms for this problem, by providing an $\mathcal{O}(\log \log m / \log \log \log m)$ approximation algorithm for the special case when agents have *restricted additive* valuations. Annamalai et al. [2015] and Davies et al. [2020] subsequently provided a 12.33- and $(4 + \varepsilon)$ -approximation algorithm for this restricted case, respectively. Numerous other works study *online* variants of this problem, but typically under various relaxations—since strong worst-case guarantees are impossible without additional assumptions. Some of these restrictions include allowing for some reordering in the allocation process [Epstein et al., 2010] or restricting the number of agents [He and Jiang, 2005, Tan and Cao, 2005, Wu et al., 2014], or allowing transfer of items after assignment [Chen and Qin, 2011].

Other sequential decision-making models. We briefly mention several other models that may appear similar to (or could be superficially framed as) repeated matching, but are in fact distinct. In the *temporal voting* model [Alouf-Heffetz et al., 2022, Bulteau et al., 2021, Chandak et al., 2024, Elkind et al., 2022, 2024a,b, 2025b,c, Phillips et al., 2025, Zech et al., 2024], the outcome is a sequence of decisions, where in each round a single project or candidate is selected. These outcomes are *public* in nature; they simultaneously benefit all agents rather than being individually allocated. While the same universal set of alternatives may exist across rounds (as in our model), the goal in temporal voting is to ensure fairness and representational balance across time in collective decisions. This differs fundamentally from repeated matching, where items are assigned *exclusively* to individual agents in each round, and fairness arises from managing trade-offs in personal allocations over time. Another related body of work looks at the *online fair division* model [Aleksandrov et al., 2015, Choo et al., 2025, Neoh et al., 2025, Zhou et al., 2023], where the repeated perspective does not apply, since a defining feature is uncertainty about future arrivals and valuations (the set of goods is not known in advance).

2 Preliminaries

Given a positive integer z , let $[z] = \{1, \dots, z\}$. We consider the problem of fairly matching a set of n agents $N = [n]$ to a set of $m \geq n$ goods $G = \{g_1, \dots, g_m\}$ over T rounds. We note that this is without loss of generality—to model the case of $m < n$, one can simply add zero-valued goods to arrive at the $m \geq n$ case and the results remain the same.

Matchings. A *matching* M is an injective map from N to G . We have $M(i) = g$ if and only if agent $i \in N$ is matched to good $g \in G$. In some instances, we also represent a matching either as a n -tuple $M = (M(1), \dots, M(n))$ or as an $n \times m$ matrix M , where $M_{ij} = 1$ if $M(i) = g_j$, and 0 otherwise. We denote the set of all sequences of matchings with length at least $t \in [T]$ as $\mathbb{S}^t$.

Valuations. Let $u_i(g)$ denote the non-negative value that agent $i \in N$ receives when matched to good $g \in G$. The *valuation profile* of a matching M is the n -tuple $(u_1(M(1)), \dots, u_n(M(n)))$. Given a sequence of T matching $S = (M^1, \dots, M^T)$, the value that agent i receives under S up to round $t \in [T]$ is the sum of the values received up to that round, that is, $v_i^t(S) := \sum_{s=1}^t u_i(M^s(i))$.

Instances. An instance of the *egalitarian repeated matching* problem is a tuple $\mathcal{I} = (N, G, T, \{u_i\}_{i \in N})$. The egalitarian (or maximin) objective seeks to maximize the value received by the worst-off agents. Let $t \in [T]$. We define the *bottleneck agents* of a sequence $S \in \mathbb{S}^t$ at round t as the set of agents who received the lowest value under S up to that round. We further define the *bottleneck value* as the value received by the bottleneck agents, that is, $b^t(S) := \min_{i \in N} v_i^t(S)$.

Objective. Motivated by the egalitarian objective, we denote the maximum bottleneck value at round t as $\text{OPT}(t) := \max\{b^t(S) \mid S \in \mathbb{S}^t\}$. In this work, we consider two notions of optimality²: one that ensures the best outcome at a specific round, and another that ensures the best outcome at every round up to a given round. Both concepts of this nature (fairness at the end or at the end of each prefix) have been studied in temporal/repeated fair division [Elkind et al., 2025a, Igarashi et al., 2024] and repeated matching [Caragiannis and Narang, 2024].

We first introduce the weaker notion of optimality,³ which is defined by mandating fairness at the end of a particular round $t \in [T]$. More formally, we say that a sequence $S \in \mathbb{S}^t$ is *optimal* at round $t \in [T]$ if $b^t(S) = \text{OPT}(t)$.

Note that this property does not require optimality to hold at any previous rounds s , for $s < t$. However, for any round $t \in [T]$, if we require optimality at every round $s \leq t$, then we get a stronger notion of optimality. More formally, we say that a sequence $S \in \mathbb{S}^t$ is *anytime optimal* up to round $t \in [T]$ if $b^s(S) = \text{OPT}(s)$ for all rounds $s \in [t]$.

Observe that while anytime optimality is significantly stronger than standard optimality, positive results for anytime optimality do not necessarily extend to the well-studied online setting. This is because, in the online setting, goods typically arrive one at a time, and valuations over these goods can be arbitrary—potentially over an unlimited set.

Efficiency. We also consider *Pareto optimality*, a notion of economic efficiency commonly studied in the social choice literature. Formally, a matching M is said to *weakly Pareto dominates* another matching M_0 if all agents $i \in N$ receive at least as much value under M as M_0 , that is, $u_i(M(i)) \geq u_i(M_0(i))$. A matching M is said to *strongly Pareto dominates* M_0 if M weakly Pareto dominates M_0 and there exist some agent $i \in N$ with $u_i(M(i)) > u_i(M_0(i))$. A matching M is *Pareto optimal* when no matching strongly Pareto dominates M .

2.1 Allocations and Bistochastic Matrices

Working with sequences of matchings can be challenging due to the constraints imposed by each matching. It would be helpful if we could ignore these constraints in our analysis and focus solely on the frequency with which each good is allocated to each agent. We refer to such an abstraction as an allocation. An *allocation* $A = (A_1, \dots, A_n)$ is a collection of multiset, where A_i is the multiset of goods that are allocated to agent $i \in N$. We can represent an allocation as a matrix A where A_{ij} is the number of times good $g_j \in G$ appears in A_i . The value that agent i receives under A is defined as

$$v_i(A) := \sum_{g \in A_i} u_i(g) = \sum_{g_j \in G} A_{ij} u_i(g_j).$$

Lemma 2.1 states that an allocation can be transformed into a polynomial-length sequence of unique matching. Hence, when a proof is phrased in terms of allocations instead of a sequence, no generality is

²For simplicity, we refer to *optimality* as shorthand for the egalitarian welfare-maximizing optimal solution.

³Note that our problem with optimality as an objective can be reformulated as a single-shot fair division problem with T copies of each good and an added constraint that each agent receives exactly T goods. While mathematically equivalent, this formulation is unintuitive in the classical setting, non-standard, and remains unexplored (with no known algorithms designed for it) in the literature. Furthermore, the *sequential* perspective is necessary for defining and motivating anytime-optimality and enabling potential extensions, neither of which can be naturally accommodated in a single-shot optimization framework.

lost. Accordingly, we will often reason with allocations in our proofs, invoking the lemma whenever an explicit sequence of matchings is required.

Lemma 2.1. *Suppose $A \in \mathbb{R}^{n \times m}$ is an allocation with*

$$\sum_{i \in N} A_{ij} \leq T \quad \text{and} \quad \sum_{g_j \in G} A_{ij} \leq T.$$

Then, there exist a sequence of matchings S consisting of $d \leq m^2 - m + 1$ unique matchings that satisfy $v_i^T(S) \geq v_i(A)$. This can be computed in polynomial time.

Several proofs of our results, including the preceding lemma, represent an allocation as a bistochastic matrix. A *bistochastic matrix* is a non-negative square matrix whose rows and columns each sum to 1, and a *scaled integer bistochastic matrix* is its integer counterpart, with non-negative integer entries and the sum of each row and column is a common integer. We defer an extended discussion of the mathematical preliminaries (along with all other omitted proofs in this paper) to the appendix.

3 Finding Optimal Sequences

We begin by focusing on optimality in this section. We first show that finding an optimal sequence of matchings is computationally intractable. We then show an relationship between a multiplicative approximation to our problem and the popular Santa Claus problem. Since computing exact solutions is intractable for large instances, we propose an approximation algorithm to find a near-optimal sequence efficiently. We also complement the hardness result by introducing a fixed-parameter tractable (FPT) algorithm that finds an optimal sequence when n or m is a constant, thereby providing an efficient algorithm for practical applications.

We assume that the reader is familiar with basic notions of classic complexity theory [Papadimitriou, 2007] and parameterized complexity [Flum and Grohe, 2006, Niedermeier, 2006].

3.1 Hardness Results

Consider the decision problem associated with the egalitarian repeated matching problem, as follows.

EGALITARIAN REPEATED MATCHING (ERM)

Input: An instance $(N, G, T, \{u_i\}_{i \in N})$ and a target κ .

Question: Is there a sequence $S \in \mathbb{S}^T$ with $b^T(S) \geq \kappa$?

We show that ERM is NP-complete by reducing from a known NP-hard problem, 3-OCC-3-SAT (defined in the proof). This result also implies that ERM is APX-hard—that is, there exists no polynomial-time approximation scheme (PTAS) for the problem. Our result is as follows.

Theorem 3.1. *ERM is NP-complete (and APX-hard) even when $u_i(g) \in \{0, 0.5, 1\}$ for all $i \in N$ and $g \in G$, for any $T \geq 2$.*

An implication of ERM not having a PTAS is that only constant-factor multiplicative approximations may be possible (though its existence is not guaranteed). We define this formally: for any $c \in [1, \infty)$, we say that an algorithm is *c-approximate* (or simply *c-approx*) if the sequence $S \in \mathbb{S}^t$ returned by the algorithm satisfy $b^t(S) \geq \text{OPT}/c$ for all $t \in [T]$. When $c = 1$, we have an exact algorithm. A natural question is whether ERM admits a *c-approx* algorithm, for some constant $c \in [1, \infty)$. Interestingly, we show that the existence of a *c-approx* algorithm for ERM would imply the existence of

a c -approx algorithm for the single-shot egalitarian fair division problem (i.e., the Santa Claus problem with additive valuations⁴).

Proposition 3.2. *For any $c \in [1, \infty)$, there is a c -approx algorithm for ERM only if there is a c -approx algorithm for the Santa Claus problem with additive valuations.*

The result above implies that finding even a constant-factor multiplicative approximation algorithm for ERM is likely to be very challenging. This is because, despite the Santa Claus problem being a well-studied and long-standing problem, no constant-factor approximation is currently known for the version with general additive valuations. A constant-factor approximation is only known in the restricted additive case.⁵

3.2 Approximation Algorithm

Given the results above, we focus on whether we can achieve an *additive* approximation with respect to optimality instead. We now describe an approximation algorithm that achieves an additive approximation bound independent of the number of rounds T . Crucially, this implies that as T increases, the approximate solution converges rapidly to the optimal one. The setting when the number of rounds is large can be observed in applications where the matching process runs continuously over extended periods—such as dynamic spectrum allocation (where the system operates continuously, often measured in (milli)seconds), leading to an immense number of allocation rounds.

Without loss of generality, we can assume that $n = m$; otherwise, we can simply create $m - n$ dummy agents with $u_i(g_j) = \max_{i' \in N} \max_{g'_j \in G} u_{i'}(g'_j)$ for all dummy agents i and goods g . Then, consider the following linear program:

$$\begin{aligned}
 & \underset{b, B}{\text{maximize}} && b && \text{(P1)} \\
 & \text{subject to} && \sum_{g_j \in G} B_{ij} u_i(g_j) \geq b, && \forall i \in N, \\
 & && \sum_{g_j \in G} B_{ij} = 1, && \forall i \in N, \\
 & && \sum_{i \in N} B_{ij} = 1, && \forall g_j \in G, \\
 & && B_{ij} \geq 0, && \forall i \in N, \forall g_j \in G.
 \end{aligned}$$

Note that the solution to (P1) is a bistochastic matrix B . Our approximation algorithm uses Birkhoff’s algorithm to decompose B into a convex combination of matchings. The number of times each matchings are included in the sequence is then determined by the convex coefficients (see Algorithm 1).

Then, we prove the following result.

Theorem 3.3. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$, the sequence $S \in \mathbb{S}^T$ returned by Algorithm 1 satisfy*

$$b^T(S) \geq \text{OPT}(T) - m \cdot \max_{i \in N} \max_{g \in G} u_i(g).$$

Proof. Consider the allocation A in which $A_{ij} = \lfloor TB_{ij} \rfloor$ for all $i \in N$ and $g_j \in G$. Note that for each $g_j \in G$, we have

$$\sum_{i \in N} A_{ij} = \sum_{i \in N} \lfloor TB_{ij} \rfloor \leq \sum_{i \in N} TB_{ij} = T,$$

⁴We specify “additive valuations” explicitly as some works (e.g., Davies et al. [2020]) consider a more restricted variant of the Santa Claus problem with restricted additive valuations.

⁵The current best known approximation factor is $(4 + \varepsilon)$, for a small $\varepsilon > 0$ in this restricted case [Davies et al., 2020].

Algorithm 1 Approximation algorithm for finding an optimal sequence of matchings

Input: An instance $\mathcal{I} = (N, G, T, \{u_i\}_{i \in N})$

- 1: let B be the solution to linear program (P1)
 - 2: decompose B into $\alpha_1 M_1 + \dots + \alpha_d M_d$ using Algorithm 4
 - 3: let S be an empty sequence
 - 4: add $\lfloor T\alpha_k \rfloor$ copies of M_k in S for each $k \in [d]$
 - 5: add any matchings into S so that $|S| = T$
 - 6: **return** S
-

and similarly, for each $i \in N$, we have

$$\sum_{g_j \in G} A_{ij} = \sum_{g_j \in G} \lfloor TB_{ij} \rfloor \leq \sum_{g_j \in G} TB_{ij} = T.$$

By Lemma 2.1, there exist a sequence S over T rounds composed of at most $O(m^2)$ unique matchings such that $v_i^T(S) \geq v_i(A)$. Then, for any agent $i \in N$, we have

$$\begin{aligned} v_i^T(S) &\geq v_i(A) \geq \sum_{g_j \in G} u_i(g_j) \lfloor TB_{ij} \rfloor \\ &\geq \sum_{g_j \in G} u_i(g_j) \cdot (TB_{ij} - 1) \\ &= \sum_{g_j \in G} TB_{ij} u_i(g_j) - \sum_{g_j \in G} u_i(g_j) \\ &\geq Tb - m \cdot \max_{g_j \in G} u_i(g_j) \\ &\geq \text{OPT}(T) - m \cdot \max_{g_j \in G} u_i(g_j). \end{aligned}$$

Let $k \in N$ be a bottleneck agent of sequence S at round T so that $b^T(S) = v_k^T(S)$. Then, we have

$$b^T(S) \geq \text{OPT}(T) - m \cdot \max_{g_j \in G} u_k(g_j) \geq \text{OPT}(T) - m \cdot \max_{i \in N} \max_{g_j \in G} u_i(g_j). \quad \square$$

Note that although the maximum valuation can be arbitrarily large, they are typically bounded in practice. Consequently, such a bound remains informative and relevant. Instance-dependent additive bounds of this type are well-established in the literature, particularly in the context of stochastic bandits [Lattimore and Szepesvári, 2020, Lim et al., 2024] and online fair division [Benadè et al., 2018, Hajiaghayi et al., 2022].

3.3 Fixed-Parameter Tractable (FPT) Algorithm

Next, we consider another approach to dealing with computational intractability. We show that the problem is *fixed parameter tractable* (FPT) when the number of agents is a fixed parameter, i.e., there exists an algorithm that can compute an optimal sequence in polynomial-time when n is a constant. This provides a practical solution for small-group matching. Our result is as follows.

Theorem 3.4. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$, ERM is FPT with respect to n .*

The proof of Theorem 3.4 relies on our newly established characterizations of Pareto-optimal and maximum matchings in terms of permutations of agents. These results may be of independent interest to researchers in matching and house allocation.

In particular, let $\pi : N \rightarrow [n]$ be a permutation of the agents. A matching M_* is said to be π -*optimal* if there exists no matching M such that

- Some agent $i \in N$ satisfies $u_i(M(i)) > u_i(M_*(i))$; and
- For every such agent i , it holds that for all agents $i' \in N$ with $\pi(i') < \pi(i)$, we have $u_{i'}(M(i')) \geq u_{i'}(M_*(i'))$.

Then, we obtain the following lemma.

Lemma 3.5. *A matching M is Pareto optimal if and only if it is π -optimal for some permutation π .*

In the context of house allocation without indifferences, it is well-established that *serial dictatorship* characterizes Pareto-optimal allocations [Abdulkadiroğlu and Sönmez, 1998]. However, when agents are allowed to express indifferences between houses, the allocations produced by serial dictatorship are not guaranteed to be Pareto optimal [Abraham et al., 2004]. Therefore, our definition of π -optimal can be interpreted as an extension of serial dictatorship that ensures Pareto optimality even in the presence of indifferences.

We describe how this characterization leads to an FPT algorithm in Section D.

4 Anytime Optimality

In this section, we consider the problem of *anytime optimality*, a stronger notion that requires optimality at every round prefix. We show that an anytime optimal sequence always exists when $n = 2$, but determining whether such a sequence exists for $n \geq 3$ is **coNP**-hard. The setting of $n = 2$ is a widely studied and is an important special case in related literature [Elkind et al., 2025a, Gollapudi et al., 2020, Igarashi et al., 2024]. Our results are as follows.

Theorem 4.1. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$ with $n = 2$, there always exist an anytime optimal sequence of matchings, and we can find it in polynomial time.*

However, we show that this positive result does not extend to the case when $n \geq 3$, for all $T \geq 2$, with the following impossibility result.

Proposition 4.2. *An anytime optimal sequence might not exist for any problem instance $(N, G, T, \{u_i\}_{i \in N})$ with $n \geq 3$ and $T \geq 2$.*

Proof. Consider the following instance with $m = n = 3$. For each $i \in N$ and $g_j \in G$, let $u_i(g_j) = U_{ij}$, where

$$U = \begin{bmatrix} 5 & 2 & 1 \\ 3 & 3 & 2 \\ 2 & 5 & 1 \end{bmatrix}.$$

Note that $\text{OPT}(1) = 2$ and $\text{OPT}(2) = 6$. Furthermore, the only way to achieve $\text{OPT}(2)$ is by choosing $M_1 = (1, 2, 3)$ and $M_2 = (3, 1, 2)$ in any order. As such, the bottleneck value at $t = 1$ is 1, which is not anytime optimal. $\square$

The above implies that we cannot hope for anytime optimality in most cases. However, given a problem instance, one may still wish to obtain an anytime optimal result *if it exists*. Unfortunately, we show that even determining whether an instance admits an anytime optimal solution is computationally intractable, with the following result.

Theorem 4.3. *Given instance $\mathcal{I} = (N, G, T, \{u_i\}_{i \in N})$, the problem of deciding if $\mathcal{I}$ admits an anytime optimal sequence is **coNP**-hard.*

Finally, we complement the above hardness result with an approximation algorithm that achieves an additive approximation bound independent of the number rounds T . Again, this means that as T increases, the approximate solution converges rapidly to the optimal one.

Algorithm 2 Approximate algorithm for anytime optimal sequence

Input: An instance $\mathcal{I} = (N, G, T, \{u_i\}_{i \in N})$

- 1: let B be the solution to (P1)
 - 2: decompose B into $\alpha_1 M_1 + \dots + \alpha_d M_d$ using Algorithm 4
 - 3: initialize $n_k = 0$ for all $k \in [d]$
 - 4: **for** $t = 1, \dots, T$ **do**
 - 5: choose matching $M^t = \arg \min_{M_k} (n_k + 1) / \alpha_k$
 - 6: update $n_k \leftarrow n_k + 1$
 - 7: **end for**
 - 8: **return** $\{M_1, \dots, M_T\}$
-

Theorem 4.4. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$, there always exist a sequence of matchings that is approximate anytime optimal. Furthermore, Algorithm 2 outputs a sequence of matchings S , in polynomial time, that satisfy*

$$b^t(S) \geq \text{OPT}(t) - 5m \cdot \max_{i \in N} \max_{g \in G} u(g), \quad \forall t \in [T].$$

Proof sketch. Let n_{kt} be the value of n_k after round t . After each round $t \in [T]$, we claim that our choice of matching M^t maintains the invariant $n_{kt} \geq \alpha_k \cdot t - 1$ for all $k \in [d]$. Intuitively, this says that by any round t , each matching M_k has been selected for roughly its intended α_k fraction of the rounds. Thus, we will get a result similar to that of Theorem 3.3. More specifically, we can show that $v_i^t(S) \geq \text{OPT}(t) - d \cdot \max_{g \in G} u_i(g)$ for all $i \in N$. Observe that since (P1) has $m^2 + 5m$ inequality constraints and $m^2 + 1$ variables, $m^2 + 1$ constraints will be tight at a vertex solution, meaning there are at most $5m$ non-zero entries in B , which implies that $d \leq 5m$. $\square$

5 Special Cases

In this section, we shift our focus back to optimality⁶ and consider three special cases: (1) when agents have binary valuations, (2) when there are only two types of goods, and (3) when agents share identical valuations. For each of the first two cases, we provide an algorithm that computes an optimal sequence of matchings in polynomial time. A key technique that we used here is to reduce the problem to one of *circulation with demand* and leveraging the Ford-Fulkerson algorithm to compute a feasible circulation. We then show for the third case that the problem is hard even for optimality, address a special case where we it can be solved in polynomial time, and provide an approximate anytime optimal algorithm for it.

5.1 Binary Valuations

The first setting we consider is when agents have *binary valuations*, i.e. $u_i : G \rightarrow \{0, 1\}$ for all agents $i \in N$. This is an important and well-studied subclass of valuations (sometimes referred to as *binary additive* valuations). Numerous fair division [Aleksandrov et al., 2015, Amanatidis et al., 2021, Bouveret and Lemaître, 2016, Freeman et al., 2019, Halpern et al., 2020, Hosseini et al., 2020, Suksompong and Teh, 2022] and matching [Bogomolnaia and Moulin, 2004, Gollapudi et al., 2020] papers consider this setting. Binary valuations can also be viewed as *approval votes*, which have long been studied in the voting literature [Brams and Fishburn, 2007, Kilgour, 2010], and permit very simple elicitation.

⁶Unfortunately, anytime optimality is a strong condition with relatively strong negative results (as with many similar problems in the online setting). We leave the existence (or impossibility) of obtaining anytime optimality in special cases as an interesting direction for future work.

Notably, under binary valuations, maximizing egalitarian welfare is equivalent to maximizing *Nash welfare* (i.e., the geometric mean), which is an extremely popular concept in fair division, and has many desirable properties [Halpern et al., 2020, Suksompong and Teh, 2022].

We first establish the following lemma.

Lemma 5.1. *Let G' be the goods in a maximum matching. Then, for any matching M , there is a matching M_* that weakly Pareto dominates M and that the goods matched by M_* is a subset of G' .*

The above lemma basically provides another characterization, this time, of maximum matchings under binary valuations in terms of Pareto optimality. To the best of our knowledge, this result is also novel in the context of house allocation, which may be of independent interest. This lemma is used to prove the following result.

Theorem 5.2. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$ with binary valuations, we can find an optimal sequence of matchings in polynomial time.*

Note that the NP-hardness result of Theorem 3.1 implies that we cannot strengthen the positive results to the setting where agents have *ternary valuations* (or three-valued instances) [Fitzsimmons et al., 2025].

5.2 Two Types of Goods

Next, we consider the setting with two *types* of goods: each good can be divided into two groups, and each agent values all goods in a particular group equally. This preference restriction is also commonly studied in (temporal) fair division [Aziz et al., 2023, Elkind et al., 2025a, Garg et al., 2024]. Formally, let $G_0, G_1 \subseteq G$ be a partition of the set of goods such that $G_0 \cap G_1 = \emptyset$, $G_0 \cup G_1 = G$, and for all agent $i \in N$ and all goods $g, g' \in G_r$ for some $r \in \{0, 1\}$, we have $u_i(g) = u_i(g')$. Then, our result is as follows.

Theorem 5.3. *Given an instance $(N, G_1 \cup G_2, T, \{u_i\}_{i \in N})$ with two types of goods, we can find an optimal sequence of matchings in polynomial time.*

5.3 Identical Valuations

The last special case we consider here is one where agents have *identical valuation* functions, i.e., $u_i = u_{i'}$ for all agents $i, i' \in N$. The setting with identical valuations is also well-studied in the repeated fair division/matching [Caragiannis and Narang, 2024, Igarashi et al., 2024] and standard fair division [Barman and Sundaram, 2020, Mutzari et al., 2023, Plaut and Roughgarden, 2020] literature. Moreover, works on semi-online multiprocessor scheduling with the makespan minimization objective (analogous to the egalitarian objective) [Cheng et al., 2005, Kellerer et al., 1997] focus on identical valuations as well (since machines are identical in that setting).

We show that even under this restricted setting of identical valuations, the problem of finding an optimal sequence is generally still NP-hard.

Theorem 5.4. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$ with identical valuations, finding an optimal sequence of matchings is NP-complete.*

However, when T is a multiple of n , we shown that the problem can be solved in polynomial time, with the following two results. We note that the case when T is a multiple of n is also a popular special case studied in repeated matching/fair division [Caragiannis and Narang, 2024, Igarashi et al., 2024]

Theorem 5.5. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$ with identical valuations and $T = kn$ for some $k \in \mathbb{Z}$, we can find an optimal sequence of matchings in polynomial time.*

Finally, we complement the above with an approximation algorithm that achieves (even anytime) optimality up to an additive approximation factor of $\max_{g \in G} u(g)$.⁷ This gives us a stronger result compared to the general case, which is also only for optimality (as in Theorem 3.3).

Theorem 5.6. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$ with identical valuations, we can find, in polynomial time, a sequence of matchings S that satisfy*

$$b^t(S) \geq \text{OPT}(t) - \Delta, \quad \forall t \in [T],$$

where Δ is the difference in value between the most valuable good and the n -th most valuable good.⁸

6 Conclusion

In this work, we introduced and studied a model of repeated matching with goal of obtaining egalitarian optimality. We investigated the computational complexity of achieving optimality and anytime optimality, and identified several settings where these problems can be solved efficiently, together with accompanying algorithms. Specifically, for optimality, we provided an approximation algorithm independent of T , and FPT algorithms with respect to n or m . For anytime optimality, we provided an approximation algorithm that complements the hardness and impossibility result even in simple cases. We also showed two special cases (binary valuations, two types of goods) where optimality can be achieved, and a final special case (identical valuations) where approximate anytime optimality can be achieved.

Directions for future work include considering other special cases that admit efficient optimal solutions, such as bi-valued utilities (where each agent values each good at either 1 or some integer $p > 1$) or identical rankings. It would also be interesting to study concepts that interpolate optimality and anytime optimality (e.g., optimality at every τ timesteps). In two of our special cases, we mentioned the equivalence between egalitarian and Nash welfare. It would be interesting to identify the conditions under which these two objectives are equivalent in this setting.

References

- Atila Abdulkadiroğlu and Tayfun Sönmez. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):689–701, 1998.
- Atila Abdulkadiroğlu and Tayfun Sönmez. House allocation with existing tenants. *Journal of Economic Theory*, 88(2):233–260, 1999.
- David J Abraham, Katarína Cechlárová, David F Manlove, and Kurt Mehlhorn. Pareto optimality in house allocation problems. In *Proceedings of the 15th International Symposium on Algorithms and Computation (ISAAC)*, pages 3–15, 2004.
- Elad Aigner-Horev and Erel Segal-Halevi. Envy-free matchings in bipartite graphs and their applications to fair division. *Information Sciences*, 587:164–187, 2022.
- Martin Aleksandrov, Haris Aziz, Serge Gaspers, and Toby Walsh. Online fair division: Analysing a food bank problem. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2540–2546, 2015.
- Shiri Alouf-Heffetz, Laurent Bulteau, Edith Elkind, Nimrod Talmon, and Nicholas Teh. Better collective decisions via uncertainty reduction. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 24–30, 2022.

⁷We denote agents’ identical utility function as u . Then, $v_i^t(S) := \sum_{s=1}^t u(M^s(i))$ for all $t \in [T]$.

⁸This is equivalent to the concept of *gap* in the bandits literature.

- Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, Alexandros Hollender, and Alexandros A. Voudouris. Maximum Nash welfare and other stories about EFX. *Theoretical Computer Science*, 863: 69–85, 2021.
- Chidambaram Annamalai, Christos Kalaitzis, and Ola Svensson. Combinatorial algorithm for restricted max-min fair allocation. In *Proceedings of the 2015 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1357–1372, 2015.
- Haris Aziz, Jeremy Lindsay, Angus Ritossa, and Mashbat Suzuki. Fair allocation of two types of chores. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 143–151, 2023.
- Gabriel Balan, Dana Richards, and Sean Luke. Long-term fairness with bounded worst-case losses. *Autonomous Agents and Multi-Agent Systems*, 22:43–63, 2011.
- Étienne Bamas, Alexander Lindermayr, Nicole Megow, Lars Rohwedder, and Jens Schlöter. Santa claus meets makespan and matroids: Algorithms and reductions. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2829–2860, 2024.
- Nikhil Bansal and Maxim Sviridenko. The santa claus problem. In *Proceedings of the 38th ACM Symposium on Theory of Computing (STOC)*, pages 31–40, 2006.
- Siddharth Barman and Ranjani G. Sundaram. Uniform welfare guarantees under identical subadditive valuations. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 46–52, 2020.
- Ali Belgacem. Dynamic resource allocation in cloud computing: Analysis and taxonomies. *Computing*, 104(3):681–710, 2022.
- Gerdus Benadè, Aleksandr M. Kazachkov, Ariel D. Procaccia, and Christos-Alexandros Psomas. How to make envy vanish over time. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*, pages 593–610, 2018.
- Garrett Birkhoff. Three observations on linear algebra. *Univ. Nac. Tucumán. Revista A*, 5:147–151, 1946.
- Anna Bogomolnaia and Hervé Moulin. Random matching under dichotomous preferences. *Econometrica*, 72(1):257–279, 2004.
- Sylvain Bouveret and Michel Lemaître. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Autonomous Agents and Multiagent Systems*, 30(2):259–290, 2016.
- Steven J. Brams and Peter C. Fishburn. *Approval Voting*. Springer, 2007.
- Laurent Bulteau, Noam Hazon, Rutvik Page, Ariel Rosenfeld, and Nimrod Talmon. Justified representation for perpetual voting. *IEEE Access*, 9:96598–96612, 2021.
- Ioannis Caragiannis and Shivika Narang. Repeatedly matching items to agents fairly and efficiently. *Theoretical Computer Science*, 981:114246, 2024.
- Nikhil Chandak, Shashwat Goel, and Dominik Peters. Proportional aggregation of preferences for sequential decision making. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 9573–9581, 2024.

- Xufeng Chen and Sen Qin. On-line machine covering on two machines with local migration. *Computers & Mathematics with Applications*, 62(5):2336–2341, 2011.
- Tai Chiu Edwin Cheng, Hans Kellerer, and Vladimir Kotov. Semi-on-line multiprocessor scheduling with given total processing time. *Theoretical Computer Science*, 337(1):134–146, 2005.
- Davin Choo, Yan Hao Ling, Warut Suksompong, Nicholas Teh, and Jian Zhang. Envy-free house allocation with minimum subsidy. *Operations Research Letters*, page 107103, 2024.
- Davin Choo, Winston Fu, Derek Khu, Tzeh Yuan Neoh, Tze-Yang Poon, and Nicholas Teh. Approximate proportionality in online fair division. *arXiv preprint arXiv:2508.03253*, 2025.
- Sami Davies, Thomas Rothvoss, and Yihao Zhang. A tale of santa claus, hypergraphs and matroids. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2748–2757, 2020.
- Stephen Demko and Theodore P. Hill. Equitable distribution of indivisible objects. *Mathematical Social Sciences*, 16(2):145–158, 1988.
- Jamal Elhachmi. Distributed reinforcement learning for dynamic spectrum allocation in cognitive radio-based internet of things. *IET Networks*, 11(6):207–220, 2022.
- Edith Elkind, Sonja Kraicz, and Nicholas Teh. Fairness in temporal slot assignment. In *Proceedings of the 15th International Symposium on Algorithmic Game Theory (SAGT)*, pages 490–507, 2022.
- Edith Elkind, Tzeh Yuan Neoh, and Nicholas Teh. Temporal elections: Welfare, strategyproofness, and proportionality. In *Proceedings of the 27th European Conference on Artificial Intelligence (ECAI)*, pages 3292–3299, 2024a.
- Edith Elkind, Svetlana Obraztsova, and Nicholas Teh. Temporal fairness in multiwinner voting. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 22633–22640, 2024b.
- Edith Elkind, Alexander Lam, Mohamad Latifian, Tzeh Yuan Neoh, and Nicholas Teh. Temporal fair division of indivisible items. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 676–685, 2025a.
- Edith Elkind, Tzeh Yuan Neoh, and Nicholas Teh. Not in my backyard! Temporal voting over public chores. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3814–3820, 2025b.
- Edith Elkind, Svetlana Obraztsova, Jannik Peters, and Nicholas Teh. Verifying proportionality in temporal voting. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI)*, pages 13805–13813, 2025c.
- Leah Epstein, Asaf Levin, and Rob van Stee. Max-min online allocations with a reordering buffer. In *Proceedings of the 37th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 336–347, 2010.
- Zack Fitzsimmons, Vignesh Viswanathan, and Yair Zick. On the hardness of fair allocation under ternary valuations. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2025.
- Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006.

- Lester R. Ford and Delbert R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.
- Rupert Freeman, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. Equitable allocations of indivisible goods. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 280–286, 2019.
- Jiarui Gan, Warut Suksompong, and Alexandros A. Voudouris. Envy-freeness in house allocation problems. *Mathematical Social Sciences*, 101:104–106, 2019.
- Jugal Garg, Aniket Murhekar, and John Qin. Weighted EF1 and PO allocations with few types of agents or chores. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2799–2806, 2024.
- Sreenivas Gollapudi, Kostas Kollias, and Benjamin Plaut. Almost envy-free repeated matching in two-sided markets. In *Proceedings of the 16th International Conference on Web and Internet Economics (WINE)*, pages 3–16, 2020.
- Priya Gupta, Makrand Samvatsar, and Upendra Singh. Cloud computing through dynamic resource allocation scheme. In *Proceedings of the 2017 International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 544–548, 2017.
- MohammadTaghi Hajiaghayi, MohammadReza Khani, Debmalya Panigrahi, and Max Springer. Online algorithms for the Santa Claus problem. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 30732–30743, 2022.
- Daniel Halpern, Ariel D. Procaccia, Alexandros Psomas, and Nisarg Shah. Fair division with binary valuations: One rule to rule them all. In *Proceedings of the 16th Conference on Web and Internet Economics (WINE)*, pages 370–383, 2020.
- Yong He and Yiwei Jiang. Optimal semi-online preemptive algorithms for machine covering on two uniform machines. *Theoretical Computer Science*, 339(2):293–314, 2005.
- Hadi Hosseini, Kate Larson, and Robin Cohen. Matching with dynamic ordinal preferences. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, pages 936–943, 2015.
- Hadi Hosseini, Sujoy Sikdar, Rohit Vaish, Hejun Wang, and Lirong Xia. Fair division through information withholding. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 2014–2021, 2020.
- Aanund Hylland and Richard Zeckhauser. The efficient allocation of individuals to positions. *Journal of Political Economy*, 87(2):293–314, 1979.
- Ayumi Igarashi, Martin Lackner, Oliviero Nardi, and Arianna Novaro. Repeated fair allocation of indivisible items. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 9781–9789, 2024.
- Khushboo Jain, Meera Dhabu, Omprakash Kakde, and Nitesh Funde. Completely fair energy scheduling mechanism in a smart distributed multi-microgrid system. *Journal of King Saud University - Computer and Information Sciences*, 34(9):7819–7829, 2022.
- Hans Kellerer, Vladimir Kotov, Maria Grazia Speranza, and Zsolt Tuza. Semi on-line algorithms for the partition problem. *Operations Research Letters*, 21(5):235–242, 1997.
- D. Marc Kilgour. Approval balloting for multi-winner elections. In Jean-François Laslier and M. Remzi Sanver, editors, *Handbook on Approval Voting*, chapter 6, pages 105–124. Springer, 2010.

- Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.
- Hendrik W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.
- Jan Karel Lenstra, David B. Shmoys, and Éva Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46:259–271, 1990.
- Eugene Lim, Vincent Y. F. Tan, and Harold Soh. Stochastic bandits for egalitarian assignment. *Transactions on Machine Learning Research*, 2024.
- Karl Jochen Micheel and Anaëlle Wilczynski. Fairness in repeated house allocation. In *Proceedings of the 27th European Conference on Artificial Intelligence (ECAI)*, pages 3549–3556, 2024.
- Alireza Moayedikia, Hadi Ghaderi, and William Yeoh. Optimizing microtask assignment on crowd-sourcing platforms using markov chain monte carlo. *Decision Support Systems*, 139:113404, 2020.
- Dolev Mutzari, Yonatan Aumann, and Sarit Kraus. Resilient fair allocation of indivisible goods. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 2688–2690, 2023.
- Tzeh Yuan Neoh, Jannik Peters, and Nicholas Teh. Online fair division with additional information. *arXiv preprint arXiv:2505.24503*, 2025.
- Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
- Christos H. Papadimitriou. *Computational complexity*. Academic Internet Publ., 2007. ISBN 978-1-4288-1409-7.
- Bradley Phillips, Edith Elkind, Nicholas Teh, and Tomasz Wąs. Strengthening proportionality in temporal voting. *arXiv preprint arXiv:2505.22513*, 2025.
- Benjamin Plaut and Tim Roughgarden. Almost envy-freeness with general valuations. *SIAM Journal on Discrete Mathematics*, 34(2):1039–1068, 2020.
- Rakibul Islam Rony, Elena Lopez-Aguilera, and Eduard Garcia-Villegas. Dynamic spectrum allocation following machine learning-based traffic predictions in 5g. *IEEE Access*, 9:143458–143472, 2021.
- A.T. Saraswathi, Y.R.A. Kalaashri, and S. Padmavathi. Dynamic resource allocation scheme in cloud computing. *Procedia Computer Science*, 47:30–36, 2015.
- Iman Shames, Anna Dostovalova, Jijoong Kim, and Hatem Hmam. Task allocation and motion control for threat-seduction decoys. In *Proceedings of the 2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 4509–4514, 2017.
- João Soares, Fernando Lezama, Ricardo Faia, Steffen Limmer, Manuel Dietrich, Tobias Rodemann, Sergio Ramos, and Zita Vale. Review on fairness in local energy systems. *Applied Energy*, 374:123933, 2024.
- Warut Suksompong and Nicholas Teh. On maximum weighted Nash welfare for binary valuations. *Mathematical Social Sciences*, 117:101–108, 2022.
- Zhiyi Tan and Shunjuan Cao. Semi-online machine covering on two uniform machines with known total size. *Computing*, 78:369–378, 2005.

- William Thomson. Problems of fair division and the egalitarian solution. *Journal of Economic Theory*, 31(2):211–226, 1983.
- Qingyun Wu and Alvin E. Roth. The lattice of envy-free matchings. *Games and Economic Behavior*, 109:201–211, 2018.
- Yong Wu, T.C.E. Cheng, and Min Ji. Optimal algorithms for semi-online machine covering on two hierarchical machines. *Theoretical Computer Science*, 531:37–46, 2014.
- Yu Yokoi. Envy-free matchings with lower quotas. *Algorithmica*, 82(2):188–211, 2020.
- Valentin Zech, Niclas Boehmer, Edith Elkind, and Nicholas Teh. Multiwinner temporal voting with aversion to change. In *Proceedings of the 27th European Conference on Artificial Intelligence (ECAI)*, pages 3236–3243, 2024.
- Chongjie Zhang and Julie A Shah. Fairness in multi-agent sequential decision-making. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 2636–2644, 2014.
- Lin Zhou. On a conjecture by Gale about one-sided matching problems. *Journal of Economic Theory*, 52(1):123–135, 1990.
- Shengwei Zhou, Rufan Bai, and Xiaowei Wu. Multi-agent online scheduling: MMS allocations for indivisible items. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 42506–42516, 2023.

Appendix

A Allocations and Bistochastic Matrices

We devote this section to establishing Lemma 2.1.

We begin by showing that any allocation A satisfying the conditions of Lemma 2.1 can be transformed into a scaled integer bistochastic matrix via Algorithm 3.

Algorithm 3 Convert allocation to a scaled integer BM

Input: Allocation A and integer T

- 1: let B be a copy of A
- 2: append B with $m - n$ rows of zeros
- 3: **while** some row i and column j does not sum to T **do**
- 4: increment B'_{ij} by

$$T - T \cdot \max \left\{ \sum_{i'=1}^m B_{i'j}, \sum_{j'=1}^m B_{ij'} \right\}$$

- 5: **end while**
 - 6: **return** B
-

Lemma A.1. Suppose $A \in \mathbb{R}^{n \times m}$ is an allocation with

$$\sum_{i \in N} A_{ij} \leq T \quad \text{and} \quad \sum_{g_j \in G} A_{ij} \leq T.$$

There exist a scaled integer bistochastic matrix $B \in \mathbb{R}^{m \times m}$ such that the sum of each row and column is T , and for each agent $i \in N$, $v_i(B) \geq v_i(A)$. This can be computed by Algorithm 3 in polynomial time.

Proof. Observe that Algorithm 3 must exit its loop after at most $2m$ iterations. This is because after each iteration, at least one more row or column will sum to T , and no row or column will sum to greater than T . Then, after at most $2m$ iterations, every row and column must sum to T . Since each iteration can be completed in polynomial time, the loop will also terminate in polynomial time. Furthermore, by the loop condition, we know that once the algorithm exits the loop, B will be a scaled integer bistochastic matrix. $\square$

Given a bistochastic matrix B , the *Birkhoff-von Neumann Theorem* [Birkhoff, 1946] states that B can be written as a convex combination of $d \leq m^2 - m + 1$ matchings: $B = \alpha_1 M_1 + \dots + \alpha_d M_d$, where $\alpha_1, \dots, \alpha_d$ are non-negative coefficients that sums to 1, and $M_1, \dots, M_d$ are matchings. This decomposition can be computed in polynomial time using *Birkhoff's algorithm*, and we describe the procedure for computing the coefficients and matchings in Algorithm 4. Lemma A.2 further extend the Birkhoff-von Neumann Theorem to scaled integer bistochastic matrices.

Lemma A.2. If $B \in \mathbb{R}^{m \times m}$ is a scaled integer bistochastic matrix, then we can decompose B as a linear combination of $d \leq m^2 - m + 1$ matchings $M_1, \dots, M_d$ with positive integer coefficients $\alpha_1, \dots, \alpha_d$: $B = \alpha_1 M_1 + \dots + \alpha_d M_d$. This can be computed by Algorithm 4 in polynomial time.

Proof. This decomposition can be achieved by Birkhoff's algorithm, and its correctness follows analogously to the proof of correctness for Birkhoff's algorithm. As such, we will focus only on showing

Algorithm 4 Birkhoff's algorithm

Input: (Scaled integer) bistochastic matrix $B \in \mathbb{R}^{m \times m}$

- 1: let B' be a copy of B
 - 2: initialize $k = 1$
 - 3: **while** there are non-zero entries in B' **do**
 - 4: construct a bipartite graph $G = ([m], [m], E)$ such that $(i, j) \in E$ if and only if $B'_{ij} > 0$
 - 5: find a perfect matching $E' \subseteq E$ of G
 - 6: let $\alpha_k = \min\{B'_{ij} \mid (i, j) \in E'\}$
 - 7: let M_k be the permutation matrix corresponding to the perfect matching E'
 - 8: update $B' \leftarrow B' - \alpha_k M_k$
 - 9: update $k \leftarrow k + 1$
 - 10: **end while**
 - 11: **return** $\{\alpha_1, \dots, \alpha_d\}$ and $\{M_1, \dots, M_d\}$
-

that $\alpha_1 \dots, \alpha_d$ are positive integers. We claim that at the start of each iteration of the loop, the entries in A' can only be non-negative integers. This is trivially true in the first iteration. Suppose this is true for the k -th iteration. Since the coefficient α_k is the minimum entry of A' that corresponds to the perfect matching E' , α_k is a positive integer. Furthermore, after we update A' by subtracting α_k from the entries that correspond to the matching E' , they must remain as non-negative integers. Thus, at the start of iteration $k + 1$, the entries in A' can only be non-negative integers. Since the coefficient α_k is just some entry of A' at iteration $k \in [d]$, it is a positive integer. $\square$

Lemma A.1 and Lemma A.2 together imply that an allocation A can be transformed into a sequence of matchings S by first converting A into a bistochastic matrix using Algorithm 3, then next applying Algorithm 4 to convert the bistochastic matrix into a sequence. The resulting sequence satisfies the inequality $v_i^T(S) \geq v_i(A)$.

Proof of Lemma 2.1. The claim follows directly from Lemma A.1 and Lemma A.2. $\square$

B Hardness Results for Optimality (Section 3.1)

Theorem 3.1. ERM is *NP-complete* (and *APX-hard*) even when $u_i(g) \in \{0, 0.5, 1\}$ for all $i \in N$ and $g \in G$, for any $T \geq 2$.

Proof. We will utilize the following decision problem 3-OCC-3-SAT that is known to be *NP-hard*.

3-OCCURRENCES 3-SATISFIABILITY (3-OCC-3-SAT)

Input: A boolean formula Φ with p variables $x_1, \dots, x_p$ and q clauses $c_1, \dots, c_q$. For each $i \in [p]$, the literal x_i appears twice and the literal $\bar{x}_i$ appears once.

Question: Is there an assignment for the variables such that Φ evaluates to TRUE?

We first prove that ERM is *NP-hard* when $T = 2$ and $u_i(g) \in \{0, 0.5, 1\}$ for all $i \in N$ and $g \in G$. Given a 3-OCC-3-SAT instance Φ , we will reduce it to an ERM instance $(\mathcal{I}, \kappa)$ with $\mathcal{I} = (N, G, T, \{u_i\}_{i \in N})$, $T = 2$, and $\kappa = 1$. We construct $\mathcal{I}$ as follows: For each $i \in [p]$, create three agents who are labeled, by an abuse of notation, as x_{i1}, x_{i2}, x_{i3} and three goods g_{i1}, g_{i2}, g_{i3} . The base valuation

of these agents are given by

	g_{i1}	g_{i2}	g_{i3}
x_{i1}	0.5	0.5	0.0
x_{i2}	0.0	0.0	1.0
x_{i3}	0.5	0.5	1.0

and 0 otherwise. Then, for each $j \in [q]$, create one agent with label c_j . The base valuation of c_j is given by $u_{c_j}(g) = 1$ if literal x_i is in clause c_j , and $g = g_{i1}$ or $g = g_{i2}$; or if literal $\bar{x}_i$ is in clause c_j and $g = g_{i3}$; and $u_{c_j}(g) = 0$ otherwise. We create another $3p + q$ dummy goods, each with zero value for all agents.

Suppose Φ is a YES instance and let $x_1, \dots, x_p$ be a satisfying assignment. Consider the sequence S constructed as follows: For each $i \in [p]$, if x_i is TRUE, we match agents x_{i1}, x_{i2}, x_{i3} according to

	$t = 1$			$t = 2$		
	g_{i1}	g_{i2}	g_{i3}	g_{i1}	g_{i2}	g_{i3}
x_{i1}	0	1	0	1	0	0
x_{i2}	0	0	1	0	0	0
x_{i3}	0	0	0	0	0	1

where we match an agent to a dummy good if its row contains all zeros. Since there are exactly two clauses, say c_a and c_b , with literal x_i , we can, if necessary, match agent c_a to good g_{i1} at $t = 1$ and match agent c_b to good g_{i2} at $t = 2$. Likewise, if x_i is FALSE, then we match according to

	$t = 1$			$t = 2$		
	g_{i1}	g_{i2}	g_{i3}	g_{i1}	g_{i2}	g_{i3}
x_{i1}	0	1	0	1	0	0
x_{i2}	0	0	1	0	0	0
x_{i3}	1	0	0	0	1	0

Since there are exactly one clause, say c_a , with literal $\bar{x}_i$, we can, if necessary, match agent c_a to good g_{i3} at $t = 2$. It is easy to verify that each agent $i \in N$ has $v_i^2(S) \geq 1$. Thus, $(\mathcal{I}, \kappa)$ is a YES instance.

Suppose $(\mathcal{I}, \kappa)$ is a YES instance and let S be a solution to this instance. For each $j \in [q]$, if the literal x_i appears in clause c_j and agent c_j is matched to either good g_{i1} or g_{i2} , then we set x_i to TRUE to satisfy the clause. Similarly, if the literal $\bar{x}_i$ appears in clause c_j and agent c_j is matched to good g_{i3} , then we set x_i to FALSE to satisfy the clause. Since $v_{c_j}^2(S) \geq 1$, at least one literal in clause c_j must have been set to satisfy the clause.

This procedure might be ambiguous because two clauses might assign different values to the same variable. We claim that this will not happen. Suppose, for sake of contradiction, that there exist two clauses $c_a \neq c_b$ such that the literal x_i appears in clause c_a while the literal $\bar{x}_i$ appears in c_b , and that agent c_a is matched to, without loss of generality, good g_{i1} while agent c_b is matched to good g_{i3} . Then, agent x_{i2} must be matched to good g_{i3} once, leaving only one copy of good g_{i1} and two copies of good g_{i2} to be allocated to agent x_{i1} and x_{i3} . However, there is no way to achieve $b^2(S) \geq 1$ with this configuration, which leads to a contradiction.

We now prove the case for $T \geq 3$. We will perform a reduction from an instance of the decision problem $(\mathcal{I}', \kappa')$ where $\mathcal{I}' = (N', G', T', \{u'_i\}_{i \in N'})$ with $T' = 2$.

Given an instance $(\mathcal{I}', \kappa')$, we will reduce it to an instance $(\mathcal{I}, \kappa)$, where $\kappa = \kappa' + C$ with

$$C = T \cdot \max_{i \in N'} \max_{g_j \in G'} u'_i(g_j)$$

and $\mathcal{I} = (N, G, T, \{u_i\}_{i \in N})$ with $n = m = 2n' + m'$ and

$$u_i(g_j) = \begin{cases} u'_i(g_j), & \text{for } i \in \{1, \dots, n'\} \text{ and } j \in \{1, \dots, m'\}, \\ C/(T-2), & \text{for } i \in \{1, \dots, n'\} \text{ and } j = i + n' + m', \\ \kappa/(T-2), & \text{for } i \in \{n' + 1, \dots, n' + m'\} \text{ and } j = i - n', \\ \kappa/2, & \text{for } i \in \{n' + m' + 1, \dots, 2n' + m'\} \text{ and } j = i, \\ 0, & \text{otherwise.} \end{cases}$$

Suppose $(\mathcal{I}', \kappa')$ is a YES instance and let $S' \in \mathbb{S}^2$ be a solution to this instance. Let A' be the allocation associated to S' . Consider the allocation A with

$$A_{ij} = \begin{cases} A'_{ij}, & \text{for } i \in \{1, \dots, n'\} \text{ and } j \in \{1, \dots, m'\}, \\ T-2, & \text{for } i \in \{1, \dots, n'\} \text{ and } j = i + n' + m', \\ T-2, & \text{for } i \in \{n' + 1, \dots, n' + m'\} \text{ and } j = i - n', \\ 2, & \text{for } i \in \{n' + m' + 1, \dots, 2n' + m'\} \text{ and } j = i, \\ 0, & \text{otherwise.} \end{cases}$$

Since the sum of each rows and each columns of A' is at most 2, it is straightforward to verify that the sum of each rows and each columns of A is at most T . Under A , agent $i \in \{1, \dots, n'\}$ receives

$$v_i(A) = v'_i(A') + (T-2) \cdot \frac{C}{T-2} \geq \kappa' + C = \kappa,$$

agent $i \in \{n' + 1, \dots, n' + m'\}$ receives

$$v_i(A) = (T-2) \cdot \frac{\kappa}{T-2} = \kappa,$$

and agent $i \in \{n' + m' + 1, \dots, 2n' + m'\}$ receives

$$v_i(A) = 2 \cdot \frac{\kappa}{2} = \kappa.$$

By Lemma 2.1, there exist a sequence $S \in \mathbb{S}^T$ such that $v_i^T(S) \geq v_i(A) \geq \kappa$. Thus, $(\mathcal{I}, \kappa)$ is a YES instance.

Suppose $(\mathcal{I}, \kappa)$ is a YES instance and let $S \in \mathbb{S}^T$ be a solution to this instance. Let A be the allocation associated to S and A' be the first n' rows and m' columns of A . Observe that since agents $i \in \{n' + 1, \dots, n' + m'\}$ have value at least κ , $A_{ij} \geq T-2$ for $j = i - n'$. This implies that the sum of each columns of A' is at most 2. By a similar argument, we have $A_{ij} \geq 2$ for all agents $i \in \{n' + m' + 1, \dots, 2n' + m'\}$ and $j = i$. Furthermore, since agent $i \in \{1, \dots, n'\}$ can never reach a value of C by only allocating goods in $\{g_1, \dots, g_{n'+m'}\}$ to them, they must be allocated to $g_{i+n'+m'}$ for at least $T-2$ times. Since these are the same goods that must be allocated at least twice to agents in $\{n' + m' + 1, \dots, 2n' + m'\}$, we have $A_{ij} = T-2$ for $j = i + n' + m'$. This implies that the sum of each row in A' is also at most 2. Under A' , agent $i \in \{1, \dots, n'\}$ receives

$$v'_i(A') = v_i(A) - (T-2) \cdot \frac{C}{T-2} \geq \kappa - C = \kappa'.$$

By Lemma 2.1, there exist a sequence $S' \in \mathbb{S}^2$ such that $v_i'^2(S') \geq v'_i(A') \geq \kappa'$. Thus, $(\mathcal{I}', \kappa')$ is a YES instance.

We now show that ERM is APX-hard for $T = 2$ by showing that if there exists a $(2 - \epsilon)$ -approx algorithm for ERM (for any small $\epsilon > 0$), then we can determine if there is a sequence S for the instance

$\mathcal{I}$ in our reduction such that $b^T(S) \geq 1$. Suppose that $\text{OPT} \geq 1$. A $(2 - \epsilon)$ -approx algorithm would find a sequence S such that $b^T(S) \geq \text{OPT}/(2 - \epsilon) \geq 1/(2 - \epsilon) > 0.5$. Now, note that as $u_i(g) \in \{0, 0.5, 1\}$, for all $i \in N$ and $g \in G$, if $b^T(S) > 0.5$ for any sequence S , then $b^T(S) \geq 1$. Hence, a $(2 - \epsilon)$ -approx algorithm allows us to determine if there exists a sequence S for the instance $\mathcal{I}$ in our reduction such that $b^T(S) \geq 1$.

Lastly, we note that we can easily extend this APX-hardness result from $T = 2$ to any $T \geq 3$ by adding dummy agents. Specifically, for each good, introduce $T - 2$ dummy agents that only have non-zero utility for that good and zero for other goods. Hence, to satisfy these dummy agents (so they have non-zero utility), only two “instances” of any good can be given to non-dummy agents. Formally, let the set of dummy agents be $\{a_{j,k} \mid j \in [m], k \in [T - 2]\}$. For this dummy set of agents, set

$$u_{a_{j,k}}(g_{j'}) = \begin{cases} 2 \cdot \max_{g \in G} u_1(g), & \text{if } j = j', \\ 0, & \text{otherwise,} \end{cases}$$

for all $j, j' \in [m]$ and $k \in [T - 2]$. □

Proposition 3.2. *For any $c \in [1, \infty)$, there is a c -approx algorithm for ERM only if there is a c -approx algorithm for the Santa Claus problem with additive valuations.*

Proof. Let $\Phi = (N, G, (v_i)_{i \in N})$ be an instance of the Santa Claus problem with additive valuations. Define OPT to be the maximum egalitarian welfare across all allocations:

$$\text{OPT} = \max_A \min_{i \in [n]} v_i(A_i).$$

Now, we will construct an instance $\mathcal{I}$ of ERM such that $\text{OPT}(T)$ for $\mathcal{I}$ is equal to OPT and that for every sequence S such that $b^T(S) > 0$, S can be easily mapped to an allocation of goods $A = (A_1, \dots, A_n)$ for Φ such that $\min_i v_i(A_i) = b^T(S)$.

Our instance of ERM has m timesteps, m goods and $n + m \times (m - 1)$ agents. We split the agents into two sets $N_1 = \{a_1, \dots, a_n\}$ (representing the agents in Φ) and $N_2 = \{a_{j,k} \mid j \in [m], k \in [m - 1]\}$ (representing dummy agents). We construct the utilities as follows: for agents $a_i \in N_1$, let $u_{a_i}(g_j) = v_i(g_j)$, and for agents $a_{j,k} \in N_2$, let

$$u_{a_{j,k}}(g_{j'}) = \begin{cases} v_1(G), & \text{if } j' = j \\ 0, & \text{otherwise.} \end{cases}$$

We now show that for all $\kappa > 0$ there is a sequence S for $\mathcal{I}$ with bottleneck value $b^T(S) \geq \kappa$ if and only if there is an allocation A for Φ such that $\min_{i \in [n]} v_i(A_i) \geq \kappa$.

($\Leftarrow$) Suppose there is an allocation A for Φ such that $\min_{i \in [n]} v_i(A_i) \geq \kappa$. Then, we can construct allocation A' for $\mathcal{I}$ as follows: for $a_i \in N_1$, $A'_{a_i} = A_i$ and for $a_{j,k} \in N_2$, $A'_{a_{j,k}} = \{j\}$. We note that $u_{a_{j,k}}(\{j\}) \geq v_1(G) > \kappa$. We further note that all goods are allocated exactly m times and all agents are allocated at most m goods. Hence, by Lemma 2.1, there exist a sequence $S \in \mathbb{S}^T$ such that $v_i^T(S) \geq v_i(A') \geq \kappa$.

($\Rightarrow$) Suppose there is a sequence S for $\mathcal{I}$ with bottleneck value $b^T(S) \geq \kappa$. Then, let A' be the allocation associated with the sequence. We now construct an allocation A for Φ as follows: for $i \in [n]$, $A_i = A'_{a_i}$. This construction ensures that $v_i(A_i) = u_{a_i}(A'_{a_i}) \geq \kappa$. We are now only left to prove that for all goods $g_j \in [m]$, g_j was allocated at most once to agents in N_1 (i.e., $\sum_{a_i \in N_1} A'_{a_i, j} \leq 1$). As $b^T(S) > 0$, all agents receive at least one good that they have non-negative utility for. Thus, for all $a_{j,k} \in N_2$, they must be allocated the good g_j at least once. Hence, as g_j must be allocated at least $m - 1$ times to agents in N_2 , g_j was allocated only once to agents in N_1 .

Hence, $\text{OPT}(T)$ for $\mathcal{I}$ is equal to OPT , and for every sequence S such that $b^T(S) > 0$ can be easily mapped to an allocation of goods A for Φ such that $\min_{i \in [n]} v_i(A_i) \geq b^T(S)$. Thus, there is a c -approx algorithm for ERM only if there is a c -approx algorithm for the Santa Claus problem with additive valuations. $\square$

C Approximate Algorithm for Optimality (Section 3.2)

Theorem 3.3. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$, the sequence $S \in \mathbb{S}^T$ returned by Algorithm 1 satisfy*

$$b^T(S) \geq \text{OPT}(T) - m \cdot \max_{i \in N} \max_{g \in G} u_i(g).$$

Proof. Consider the allocation A in which $A_{ij} = \lfloor TB_{ij} \rfloor$ for all $i \in N$ and $g_j \in G$. Note that for each $g_j \in G$, we have

$$\sum_{i \in N} A_{ij} = \sum_{i \in N} \lfloor TB_{ij} \rfloor \leq \sum_{i \in N} TB_{ij} = T,$$

and similarly, for each $i \in N$, we have

$$\sum_{g_j \in G} A_{ij} = \sum_{g_j \in G} \lfloor TB_{ij} \rfloor \leq \sum_{g_j \in G} TB_{ij} = T.$$

By Lemma 2.1, there exist a sequence S over T rounds composed of at most $O(m^2)$ unique matchings such that $v_i^T(S) \geq v_i(A)$. Then, for any agent $i \in N$, we have

$$\begin{aligned} v_i^T(S) &\geq v_i(A) \geq \sum_{g_j \in G} u_i(g_j) \lfloor TB_{ij} \rfloor \\ &\geq \sum_{g_j \in G} u_i(g_j) \cdot (TB_{ij} - 1) \\ &= \sum_{g_j \in G} TB_{ij} u_i(g_j) - \sum_{g_j \in G} u_i(g_j) \\ &\geq Tb - m \cdot \max_{g_j \in G} u_i(g_j) \\ &\geq \text{OPT}(T) - m \cdot \max_{g_j \in G} u_i(g_j). \end{aligned}$$

Let $k \in N$ be a bottleneck agent of sequence S at round T so that $b^T(S) = v_k^T(S)$. Then, we have

$$\begin{aligned} b^T(S) &\geq \text{OPT}(T) - m \cdot \max_{g_j \in G} u_k(g_j) \\ &\geq \text{OPT}(T) - m \cdot \max_{i \in N} \max_{g_j \in G} u_i(g_j). \end{aligned} \quad \square$$

D FPT Algorithms for Optimality (Section 3.3)

In this section, we consider another approach to dealing with the computational intractability. Our goal is to develop a *fixed parameter tractable* (FPT) when the number of agents is a fixed parameter, i.e., there exists an algorithm that can compute an optimal sequence in polynomial-time when n is a constant. This provides a practical solution for small-group matching (e.g., crowdsourcing platforms divide workers into subgroups tailored to specific categories of tasks).

To build up to this result, let us first consider the easier case where the number of goods m is the fixed parameter.

Theorem D.1. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$, ERM is FPT with respect to m .*

Proof. Since the order of the matchings in a sequence does not affect the values at round T , our goal is to determine the number of times each matching should be chosen to achieve the highest bottleneck value. Let $\mathcal{M}$ be the set of all possible matchings. For each $M \in \mathcal{M}$, let X_M be the number of times M should be chosen in the sequence. We can formulate the optimization problem as the following integer linear program:

$$\begin{aligned}
& \underset{b, X}{\text{maximize}} && b \\
& \text{subject to} && \sum_{M \in \mathcal{M}} X_M u_i(M(i)) \geq b, \quad \forall i \in N, \\
& && \sum_{M \in \mathcal{M}} X_M = T, \\
& && X_M \geq 0, \quad \forall M \in \mathcal{M}.
\end{aligned} \tag{P2}$$

Since the number of variables for (P2) is at most $m! + 1$, we obtain the FPT result using Lenstra's theorem Lenstra [1983]. $\square$

In order to extend the result for that the algorithm is FPT with respect to n , we need to reduce the number of variables in (P2) to a function of n and not m . This reduction is accomplished through two observations. First, it is always possible to construct an optimal sequence consisting solely of Pareto optimal matchings, by replacing any non-Pareto optimal matching with one that strongly Pareto dominates it. Second, there are at most $n!$ unique (up to its valuation profile) Pareto optimal matchings. Together, these observations allow us to consider a smaller set of matchings $\mathcal{M}$ in (P2), with size at most $n!$, thereby achieving our desired result. We now prove the second observation by characterizing Pareto optimal matchings in terms of permutations of agents. We note that this result may be of independent interest, especially on the topic of *house allocation* [Abdulkadiroğlu and Sönmez, 1998, Abraham et al., 2004, Abdulkadiroğlu and Sönmez, 1999, Choo et al., 2024, Gan et al., 2019, Hylland and Zeckhauser, 1979, Zhou, 1990].

Lemma 3.5. *A matching M is Pareto optimal if and only if it is π -optimal for some permutation π .*

Proof. If a matching M is π -optimal for some permutation π , then M is clearly Pareto optimal: no agent can improve without harming someone with higher priority in π . Now suppose for contradiction that M is Pareto optimal but not π -optimal for any permutation π .

We define the envy⁺ graph of M as a directed graph where each vertex corresponds to an agent. There is an edge from agent i to i' if there exists a sequence of $p \geq 2$ distinct agents $(i_1, \dots, i_p)$, with $i_1 = i$ and $i_p = i'$, such that agent i_1 strictly envies i_2 , meaning $u_{i_1}(M(i_1)) < u_{i_1}(M(i_2))$, and for $r = 2, \dots, p-1$, agent i_r weakly envies agent i_{r+1} , that is, $u_{i_r}(M(i_r)) \leq u_{i_r}(M(i_{r+1}))$.

If this graph contains a cycle, we could perform a cyclic exchange among the agents to strictly improve at least one agent's value without hurting others, contradicting the Pareto optimality of M . Hence, the envy⁺ graph must be acyclic. We can therefore define π to be the topological sort of this graph in reverse dependency order, so that agents earlier in the ordering are not envied by those that come later. In particular, if $\pi(i) < \pi(i')$, then (i, i') is not an edge in the envy⁺ graph.

Let M_* be a π -optimal matching. We claim that there exists an agent $a \in N$ such that (1) $u_a(M(a)) < u_a(M_*(a))$, and (2) for all agent $i \in N$ with $\pi(i) < \pi(a)$, we have $u_i(M(i)) \leq u_i(M_*(i))$. To see this, recall that by definition of π -optimality applied to M_* , any matching, including M , must satisfy one of the following:

- For all agent $i \in N$, $u_i(M(i)) \leq u_i(M_*(i))$; or

- There exists an agent $i \in N$ such that $u_i(M(i)) > u_i(M_*(i))$, but there exists another agent $i' \in N$ with $\pi(i') < \pi(i)$ such that $u_{i'}(M(i')) < u_{i'}(M_*(i'))$.

In the first case, if all agents are indifferent between M and M_* , then M is itself π -optimal, contradicting our assumption. Hence, there must exist some agent $a \in N$ who is strictly better off under M_* , that is, $u_a(M(a)) < u_a(M_*(a))$, thus proving property (1). Since $u_i(M(i)) \leq u_i(M_*(i))$ for all i , property (2) follows immediately.

In the second case, let $i \in N$ be the agent with the highest priority under π such that $u_i(M(i)) > u_i(M_*(i))$. By definition, there must exist an agent $i' \in N$ with $\pi(i') < \pi(i)$ such that $u_{i'}(M(i')) < u_{i'}(M_*(i'))$. Setting $a = i'$ proves property (1). Since all agent with higher priority than a also outrank i , and i is the highest-priority agent property who strictly prefers M over M_* , property (2) follows. This completes the claim, allowing us to proceed with the main argument.

To analyze the structure of M_* relative to M , we define the envy path graph as a directed graph whose vertices correspond to agents. There is an edge from agent i to agent $i' \neq i$ if, under M_* , agent i receives the good that agent i' was assigned in M , that is, $M_*(i) = M(i')$.

Consider the traversal of the envy path graph starting from agent a . Since each good is matched to at most one agent in both M and M_* , each vertex in the graph has at most one incoming and one outgoing edge. As a result, the traversal either enters a cycle that includes agent a , or eventually terminates at a vertex with no outgoing edge, forming a simple path. In the former case, we obtain a cycle $(a = a_1, \dots, a_q, a_1)$; and in the latter, we obtain a path $(a = a_1, \dots, a_q)$, where it is possible that $q = 1$ if there is no outgoing edge from a .

We now show, by induction on the position $r = 1, \dots, q - 1$ along the traversal path, that (i) $u_{a_r}(M(a_r)) \leq u_{a_r}(M(a_{r+1}))$, and (ii) $\pi(a_{r+1}) < \pi(a)$. Intuitively, this means that each agent along the path or cycle weakly prefers the good assigned to the next agent, and all agents in the sequence have higher priority than a under π .

For the base case $r = 1$, observe that agent $a = a_1$ strictly prefers their assignment in M_* over M , and by construction, $M_*(a_1) = M(a_2)$. Therefore, we have

$$u_{a_1}(M(a_1)) = u_a(M(a)) < u_a(M_*(a)) = u_{a_1}(M_*(a_1)) = u_{a_1}(M(a_2)).$$

This confirms the first property, showing that agent a_1 envies a_2 in M . As such, there is an edge from a_1 to a_2 in the envy⁺ graph. Since π is defined as a reverse topological ordering of this graph, we conclude that $\pi(a_2) < \pi(a_1) = \pi(a)$, thereby establishing the second property.

For the inductive step, assume that both properties hold for all indices up to $r = k - 1$. Since $\pi(a_k) < \pi(a)$ and M_* is a π -optimal matching, it follows that

$$u_{a_k}(M(a_k)) \leq u_{a_k}(M_*(a_k)) = u_{a_k}(M(a_{k+1})),$$

which verifies the first property for $r = k$. To verify the second property, recall that $u_{a_1}(M(a_1)) < u_{a_1}(M(a_2))$, and for all $r = 2, \dots, k$, $u_{a_r}(M(a_r)) \leq u_{a_r}(M(a_{r+1}))$. By definition, there must be an edge between a_1 and a_{k+1} in the envy⁺ graph, which implies that $\pi(a_{k+1}) < \pi(a_1) = \pi(a)$, completing the inductive argument.

Thus, we return to the two cases of the traversal: either a cycle $(a = a_1, \dots, a_q, a_1)$ or a path $(a = a_1, \dots, a_q)$. If the cyclic case, we perform a cyclic exchange where each agent a_i receives $M(a_{i+1})$ for all $i \in [q - 1]$, and agent a_q receives $M(a_1)$. Since each edge in the envy path graph represents weak preference, every agent in the cycle weakly prefers their new assignment, and agent $a = a_1$ strictly prefers $M(a_2)$ over $M(a_1)$. This yields a matching that strictly Pareto dominates M , contradicting its assumed optimality.

In the path case, $M_*(a_q)$ must be a good that is unassigned in M . We construct a new matching by assigning each agent a_i to $M(a_{i+1})$ for $i \in [q - 1]$, and assigning agent a_q to $M_*(a_q)$. Again, each a_i weakly prefers their new good, and $a = a_1$ strictly prefers $M(a_2)$ over $M(a_1)$. Furthermore, since

$\pi(a_q) < \pi(a)$, the π -optimality of M_* implies that $u_{a_q}(M(a_q)) \leq u_{a_q}(M_*(a_q))$, so agent a_q weakly prefers their new good as well. This matching also strictly Pareto dominates M , again contradicting its assumed optimality. Thus, M is π -optimal for some π . $\square$

Now, note that in an optimal sequence, we can swap a matching with one that has an identical valuation profile without affecting the sequence's optimality. Thus, it is sufficient to compute just one π -optimal matching for all permutations π and consider only sequences construct from these matching. To ensure the resulting algorithm is FPT with respect to n , we need to show that it is efficient in all other parameters to compute a π -optimal matching for each π .

Lemma D.2. *Given a permutation of agents π , we can find a π -optimal matching in polynomial time.*

Proof. Denote the rank $r(i, g_j)$ of good g_j for agent i to be the number of goods (inclusive of g_j) that are valued at most as highly as g_j . More formally,

$$r(i, g_j) = \sum_{g \in G} \mathbb{I}[u_i(g) \leq u_i(g_j)]$$

where $\mathbb{I}[\cdot]$ is the indicator function, which equals 1 if the condition inside is true and 0 otherwise. Observe that if two goods have the same value $u_i(g_j) = u_i(g)$, then they have the same rank, that is, $r(i, g_j) = r(i, g)$.

Construct a complete bipartite graph $H = (N, G, E)$ where the weight of $(i, g) \in E$ is

$$w(i, g) = \begin{cases} r(i, g) \cdot m^{n-\pi(i)}, & \text{if } i \leq n, \\ 0, & \text{otherwise.} \end{cases}$$

Consider the maximum weight matching M in H . We claim that M is π -optimal.

Suppose that M is not π -optimal. Then there must exists some M_0 with an agent $i \in N$ that has $u_i(M_0(i)) > u_i(M(i))$, which implies $r(i, M_0(i)) > r(i, M(i))$. Moreover, all agents $i' \in N$ with $\pi(i') < \pi(i)$ must also satisfy $u_{i'}(M_0(i')) \geq u_{i'}(M(i'))$, which implies $r(i', M_0(i')) \geq r(i', M(i'))$. Partition N into

$$\begin{aligned} N_1 &= \{i' \in N \mid \pi(i') \leq \pi(i)\}, \\ N_2 &= \{i' \in N \mid \pi(i') > \pi(i)\}. \end{aligned}$$

Then, we have

$$\sum_{i' \in N_1} w(i', M_0(i')) - w(i', M(i')) = \sum_{i' \in N_1} (r(i', M_0(i')) - r(i', M(i'))) \cdot m^{n-\pi(i')} \geq m^{n-\pi(i)}.$$

Furthermore, we also have

$$\begin{aligned} \sum_{i' \in N_2} w(i', M(i')) - w(i', M_0(i')) &= \sum_{i' \in N_2} (r(i', M(i')) - r(i', M_0(i'))) \cdot m^{n-\pi(i')} \\ &\leq (m-1) \cdot m^{n-(\pi(i)+1)} + \dots + (m-1) \cdot m^{n-n} \\ &= m^{n-\pi(i)} - 1. \end{aligned}$$

Thus, the weight of matching M_0 is greater than the weight of the matching M and we have a contradiction. $\square$

Then, we propose Algorithm 5 that gives us our desired result, as follows.

Theorem 3.4. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$, ERM is FPT with respect to n .*

Algorithm 5 FPT algorithm for optimal sequence

Input: an instance $\mathcal{I} = (N, G, T, \{u_i\}_{i \in N})$

- 1: let $\mathcal{M}$ be an empty set
 - 2: **for all** permutation π **do**
 - 3: compute π -optimal sequence M using Lemma D.2
 - 4: add M to $\mathcal{M}$
 - 5: **end for**
 - 6: construct integer linear program (P2) based on $\mathcal{M}$
 - 7: let X be the solution to (P2)
 - 8: construct a sequence S with X_M copies of $M \in \mathcal{M}$
 - 9: **return** S
-

Proof. The correctness of Algorithm 5 follows immediately from our discussion. We will now show that it is FPT with respect to n . Let $\mathcal{M}$ be the set of π -optimal matching for all permutations π . By Lemma D.2, the loop in Algorithm 5 computes $\mathcal{M}$ in time $\text{poly}(n, m) \cdot n!$. Furthermore, since the number of variables for (P2) is at most $n! + 1$, we obtain the FPT result using Lenstra's theorem [1983]. $\square$

E Omitted Proofs in Section 4

Theorem 4.1. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$ with $n = 2$, there always exist an anytime optimal sequence of matchings, and we can find it in polynomial time.*

Proof. We begin by proving that such a sequence always exists, and then demonstrate how to construct it in polynomial time.

Proof of existence. Consider the case of $m = 2$. We will prove by induction in T . The statement is obviously true for $T = 1$. Suppose the statement is true for $T - 1$. Let $\mathbb{S}_{1:T-1} = \{S \in \mathbb{S}^T \mid b^t(S) = \text{OPT}(t), \forall t < T\}$ be the set of sequences that are anytime optimal up to round $T - 1$ and $\mathbb{S}_T = \{S \in \mathbb{S}^T \mid b^T(S) = \text{OPT}(T)\}$ be the set of sequences that are optimal at round T . By the inductive hypothesis, $\mathbb{S}_{1:T-1}$ is nonempty. We want to show that $\mathbb{S}_{1:T-1} \cap \mathbb{S}_T \neq \emptyset$.

Suppose, for sake of contradiction, that $\mathbb{S}_{1:T-1} \cap \mathbb{S}_T = \emptyset$. For each pair of sequences $S_1 = (M_1^1, \dots, M_1^T) \in \mathbb{S}_{1:T-1}$ and $S_2 = (M_2^1, \dots, M_2^T) \in \mathbb{S}_T$, there is some round $s = \min\{t \in [T] \mid M_1^t \neq M_2^t\}$ such that the sequences first deviate. Choose S_1 and S_2 such that the first deviated round s is maximized.

We claim that for all rounds $t \in \{s, \dots, T\}$, the matching $M_2^t = M_2^s$. If there is a round t in which $M_2^t \neq M_2^s$, then we can swap these two matchings to obtain a new sequence of matchings S . Since S is a rearrangement of S_2 , we know $S \in \mathbb{S}_T$. Furthermore, since there are only two types of matching for $n = m = 2$, the s -th matching for S is M_1^s . This implies that the first deviated round between $S_1 \in \mathbb{S}_{1:T-1}$ and $S \in \mathbb{S}_T$ is greater than s , which contradicts the assumption that s is the maximum first deviated round.

Consider the sequence of matchings

$$S_0 = (M_2^1, \dots, M_2^{s-1}, M_1^s, M_2^{s+1}, \dots, M_2^T)$$

that is constructed by exchanging the matching of S_2 at round s to M_1^s . Without loss of generality, we assume that agent 1 is the bottleneck agent for S_2 at round s .

If $\min\{u_1(g_1), u_1(g_2)\} > \max\{u_2(g_1), u_2(g_2)\}$, then by choosing the good that maximizes the value for agent 2 for all rounds, we will obtain an anytime optimal sequence of matchings up till round T . As such, we only have to consider $\min\{u_1(g_1), u_1(g_2)\} \leq \max\{u_2(g_1), u_2(g_2)\}$.

Recall that S_0 is optimal at round s . Since agent 1 is the bottleneck agent for S_2 at round s , we have $v_1^s(S_2) \leq v_1^s(S_0)$, which implies that $\min\{u_1(g_1), u_1(g_2)\} = u_1(M_2^s(1))$. This also implies that $\max\{u_2(g_1), u_2(g_2)\} = u_2(M_2^s(2))$; otherwise, M_4^s will weakly Pareto dominates M_2^s , and choosing M_4^s at every round produces an anytime optimal sequence of matchings up till round T . Thus, $v_1(M_2^s(1)) \leq v_2(M_2^s(2))$.

To reach a contradiction, we want to show that both $v_1^T(S_0)$ and $v_2^T(S_0)$ is at least $v_1^T(S_2)$, since we will have $b^T(S_0) \geq b^T(S_2) = \text{OPT}(T)$, which implies that $S_0 \in \mathbb{S}_T$. Since the first round of deviation between $S_1 \in \mathbb{S}_{1:T-1}$ and $S_0 \in \mathbb{S}_T$ is greater than s , this leads to a contradiction.

It is straightforward to show that $v_1^T(S_0) \geq v_1^T(S_2)$:

$$v_1^T(S_0) - v_1^T(S_2) = v_1^s(S_0) - v_1^s(S_2) \geq 0.$$

Let us now show that $v_2^T(S_0) \geq v_1^T(S_2)$. Since S_0 is optimal at round s and agent 1 is the bottleneck agent for S_2 at round s , we have $v_2^s(S_0) \geq v_1^s(S_2)$. Further recall that $u_1(M_2^s(1)) \leq u_2(M_2^s(2))$ and $M_2^t = M_2^s$ for all rounds $t \in \{s, \dots, T\}$. As such, we have

$$\begin{aligned} v_2^T(S_0) - v_1^T(S_2) &= v_2^s(S_0) + (T - s) \cdot u_2(M_2^s(2)) - v_1^s(S_2) - (T - s) \cdot u_1(M_2^s(1)) \\ &\geq (T - s) \cdot u_2(M_2^s(2)) - (T - s) \cdot u_1(M_2^s(1)) \\ &= (T - s) \cdot (u_2(M_2^s(2)) - u_1(M_2^s(1))) \\ &\geq 0. \end{aligned}$$

Since both $v_1^T(S_0)$ and $v_2^T(S_0)$ is at least $v_1^T(S_2)$, we reached a contradiction.

To extend the proof to general m , if each agent's most valued good differs, then it is optimal to match each agent to their most valued good in every round. As such, we only need to consider the case where both agents have the same most valued good.

Suppose that the most valued good of both agents is g_{j_0} , and let the next most valued good of agent i be g_{j_i} . Observe that every possible matchings are weakly dominated by either $M_1 = (g_{j_1}, g_{j_0})$ or $M_2 = (g_{j_0}, g_{j_2})$. As such, we only need to consider sequences that consist of M_1 and M_2 . This is equivalent to the case of $m = 2$ where the valuation $u'_i(g'_1) = u_i(g_{j_0})$ and $u'_i(g'_2) = u_i(g_{j_i})$ for $i \in \{1, 2\}$. The existence of an anytime optimal sequence follows from immediately from the proof for $n = 2$.

Proof of efficient constructibility. We first consider the trivial cases. If the agents have different most preferred goods, then we can just assign each agent to their most preferred good for all rounds. Otherwise, let g_0 be their common most preferred good and g_i be the second most preferred good for agent $i \in \{1, 2\}$. If there exist an agent $i \in \{1, 2\}$ with $u_i(g_0) = u_i(g_i)$, then we can match g_i to agent i and g_0 to the other agent for all rounds. Furthermore, if $u_1(g_0) \leq u_2(g_2)$, then we can just choose the matching (g_0, g_2) for all rounds. Similarly, if $u_2(g_0) \leq u_1(g_1)$, then we can just choose the matching (g_1, g_0) for all rounds.

Suppose our instance is not one of the trivial cases. Let $M_1 = (g_0, g_2)$ and $M_2 = (g_1, g_0)$. Note that all matching is weakly Pareto dominated by either M_1 or M_2 . As such, there must exist an anytime optimal sequence that contains only M_1 and M_2 . We now construct the sequence S greedily and iteratively. Consider the following loop invariant that must be satisfied before the start of iteration $t \in [T]$:

The sequence $S \in \mathbb{S}^{t-1}$ is anytime optimal up till round $t - 1$ and there exist an extension of the sequence such that it is anytime optimal up till round $t' \geq t$.

This is satisfied before the start of iteration $t = 1$ because S is empty (hence vacuously anytime optimal) and there exist an extension that is anytime optimal.

Suppose the loop invariant is satisfied before the start of iteration $t \in [T]$. We want to extend the sequence with a matching such that the loop invariant is satisfied before the start of iteration $t + 1$. By the loop invariant, we know that there exist an extension of the sequence (using only M_1 and M_2) that is anytime optimal up till round $t' \geq t$. Let $S_1 = S \cup M_1$ and $S_2 = S \cup M_2$. If $b^t(S_1) > b^t(S_2)$, then extending S to S_1 ensures the loop invariant holds for $t + 1$. This holds similar for the case of $b^t(S_1) < b^t(S_2)$.

Suppose that $b^t(S_1) = b^t(S_2)$. Since $u_i(g_0) > u_1(g_i)$ for both $i \in \{1, 2\}$, we have $v_1^t(S_1) > v_1^t(S_2)$ and $v_2^t(S_1) < v_2^t(S_2)$. Note that agent 1 (resp. agent 2) is the unique bottleneck agent for S_2 (resp. S_1). To see this, suppose agent 2 is a bottleneck agent for S_2 , that is, $b^t(S_2) = v_2^t(S_2) \leq v_1^t(S_2)$. Then, we have $v_2^t(S_1) < v_2^t(S_2) \leq v_1^t(S_2) < v_1^t(S_1)$, which implies that agent 2 is the bottleneck agent for S_1 , that is, $b^t(S_1) = v_2^t(S_1)$. This leads to a contradiction because we have

$$b^t(S_1) = v_2^t(S_1) < v_2^t(S_2) = b^t(S_2).$$

A similar argument can be used to prove the respective case. These results imply that $v_2^t(S_1) = b^t(S_1) = b^t(S_2) = v_1^t(S_2)$.

We will now show that it does not matter if we match g_0 to any agent in round t because we will have to match g_0 to the other agent in round $t + 1$. Suppose, for sake of contradiction, and without loss of generality, that it is optimal to match g_1 to agent 1 for round t and $t + 1$. Then, we have

$$\begin{aligned} & \min\{v_1^{t-1}(S) + 2u_1(g_1), v_2^{t-1}(S) + 2u_2(g_0)\} \\ &= \min\{v_1^t(S_2) + u_1(g_1), v_2^t(S_2) + u_2(g_0)\} \\ &\leq v_1^t(S_2) + u_1(g_1) \\ &= v_2^t(S_1) + u_1(g_1) \\ &\leq v_2^t(S_1) + \max\{u_1(g_1), u_2(g_2)\} \\ &\leq v_2^t(S_1) + \min\{u_1(g_0), u_2(g_0)\} \\ &= \min\{v_1^{t-1}(S) + u_1(g_0) + u_1(g_1), v_2^{t-1}(S) + u_2(g_0) + u_2(g_2)\}, \end{aligned}$$

which contradicts optimality at round $t + 1$. A similar argument can be used to show that it is not optimal to match g_2 to agent 2 for round t and $t + 1$. Hence, the only extension left is to either match g_0 to any agent in round t and to the other agent in round $t + 1$. $\square$

Proposition 4.2. *An anytime optimal sequence might not exist for any problem instance $(N, G, T, \{u_i\}_{i \in N})$ with $n \geq 3$ and $T \geq 2$.*

Proof. Consider the following instance with $m = n = 3$. For each $i \in N$ and $g_j \in G$, let $u_i(g_j) = U_{ij}$, where

$$U = \begin{bmatrix} 5 & 2 & 1 \\ 3 & 3 & 2 \\ 2 & 5 & 1 \end{bmatrix}.$$

Note that $\text{OPT}(1) = 2$ and $\text{OPT}(2) = 6$. Furthermore, the only way to achieve $\text{OPT}(2)$ is by choosing $M_1 = (1, 2, 3)$ and $M_2 = (3, 1, 2)$ in any order. As such, the bottleneck value at $t = 1$ is 1, which is not anytime optimal. $\square$

Theorem 4.3. *Given instance $\mathcal{I} = (N, G, T, \{u_i\}_{i \in N})$, the problem of deciding if $\mathcal{I}$ admits an anytime optimal sequence is **coNP-hard**.*

Proof. To prove that our problem is **coNP-hard**, we will show that the complement of our problem is **NP-hard** by reducing from the 3-PARTITION problem. In the 3-PARTITION problem, we are given a

multiset $R = \{a_1, \dots, a_{3d}\}$ and we need to decide if there exists a partition of R into d triplets such that the sum of all triplets equals to

$$\gamma = \frac{1}{d} \sum_{k=1}^{3d} a_k.$$

Given a 3-PARTITION instance R , let $\epsilon < \min\{\gamma/8 - 1/4, 1/6\}$. We will reduce R to an instance $\mathcal{I} = (N, G, T, \{u_i\}_{i \in N})$ with $n = m = 5d + 3$, $T = 3$, and $u_i(g_j) = U_{ij}$, where

$$U = \begin{array}{c|ccc|c} \begin{array}{c} \text{---} a \text{---} \\ \vdots \\ \text{---} a \text{---} \end{array} & 2\epsilon & \frac{\gamma-1}{3} & 0 & d \\ \hline \begin{array}{c} \gamma/2 \text{---} \\ \text{---} 2\epsilon \\ \text{---} 2\epsilon \end{array} & 0 & 0 & 2\epsilon & 3d \\ \hline 0 & \frac{\gamma-1}{3} & \frac{\gamma}{3} & 0 & d \\ \hline 2\epsilon & 0 & 0 & Z & 3 \\ \hline 3d & d & d & 3 & \end{array} \quad Z = \begin{bmatrix} \epsilon & \epsilon & \gamma - 2\epsilon \\ \epsilon & \epsilon & \gamma - 2\epsilon \\ \epsilon & \epsilon & \gamma - 2\epsilon \end{bmatrix}$$

Observe that $\text{OPT}(1) \leq 2\epsilon$ and $\text{OPT}(2) \leq 4\epsilon$ since at least one agents in $\{5d+1, 5d+2, 5d+3\}$ cannot be matched to good g_{5d+3} in the first two rounds. We also have $\text{OPT}(1) \geq 2\epsilon$ and $\text{OPT}(2) \geq 4\epsilon$. This can be achieved by considering the sequence that, for all rounds, matches agent $i \in \{1, \dots, d\}$ to g_{3d+i} , agent $d+1, d+2, d+3$ to $g_{5d+1}, g_{5d+2}, g_{5d+3}$ respectively, agent $i \in \{d+4, \dots, 4d\}$ to g_{i-d} , agent $i \in \{4d+1, \dots, 5d\}$ to g_i , and agent $5d+1, 5d+2, 5d+3$ to g_1, g_2, g_3 respectively. Thus, we have $\text{OPT}(1) = 2\epsilon$ and $\text{OPT}(2) = 4\epsilon$.

Suppose R is a YES instance and $R_1, \dots, R_d$ is a solution to the instance. We claim that $\text{OPT}(3) \geq \gamma$. To see this, consider the allocation A in which

$$A_{ij} = \begin{cases} 1, & \text{if } i \in \{1, \dots, d\} \text{ and } a_j \in R_i, \\ 2, & \text{if } i \in \{d+1, \dots, 4d\} \text{ and } j = i-d, \\ 1, & \text{if } i \in \{d+1, \dots, 4d\} \text{ and } j = 3d + \lceil (i-d)/3 \rceil, \\ 3, & \text{if } i \in \{4d+1, \dots, 5d\} \text{ and } j = i, \\ 1, & \text{if } i \in \{5d+1, 5d+2, 5d+3\} \text{ and } j \in \{5d+1, 5d+2, 5d+3\} \\ 0, & \text{otherwise.} \end{cases}$$

By Lemma 2.1, we can convert A into a sequence S of three matchings where each $i \in N$ satisfy $v_i^3(S) \geq \gamma$.

For sake of contradiction, suppose $\mathcal{I}$ is a YES instance and let S be an anytime optimal sequence. Since S is anytime optimal, it is optimal at $t = 3$, i.e., $b^3(S) = \text{OPT}(3) \geq \gamma$. As such, agent $i \in \{4d+1, \dots, 5d\}$ must be matched to g_i for all three rounds. Furthermore, agent $i \in \{d+1, \dots, 4d\}$ has to be matched to g_{i-d} at least twice; otherwise, $v_i^3(S)$ would be smaller than 6ϵ or $4\epsilon + \gamma/2$, which are strictly less than γ by our choice of ϵ . As such, each good $g_1, \dots, g_{3d}$ can only be matched to agent

$1, \dots, d$ at most once over all rounds. We claim that these goods must be matched to these agents exactly once over all rounds. Suppose, for sake of contradiction, there is some good g that is not matched to any agents in $\{1, \dots, d\}$. Then, we have

$$\sum_{i=1}^d v_i^3(S) \leq \left(\sum_{j=1}^{3d} u_1(g_j) \right) - u_1(g) + 2\epsilon < \gamma d - 1 + \frac{1}{3} < \gamma d,$$

since $u_1(g)$ is a positive integer and $\epsilon < 1/6$. As such, at least one agent $i \in \{1, \dots, d\}$ has $v_i^3(S) < \gamma$, which contradicts to the optimality of S . Since $g_1, \dots, g_{3d}$ and $g_{4d+1}, \dots, g_{5d}$ must be allocated three times to agents $1, \dots, 5d$, these cannot be matched to any agent $5d+1, 5d+2, 5d+3$. Thus, at least one agent $i \in \{5d+1, 5d+2, 5d+3\}$ has $v_i^1(S) \leq \epsilon < \text{OPT}(1)$, which contradicts our assumption that S is an anytime optimal sequence. Thus, $\mathcal{I}$ is a NO instance.

Suppose R is a NO instance. We claim that $\text{OPT}(3) \leq \gamma - 1$. Suppose, for sake of contradiction, that $\text{OPT}(3) > \gamma - 1$. Then, agent $i \in \{4d+1, \dots, 5d\}$ must be matched to g_i for all three rounds. Furthermore, agent $i \in \{d+1, \dots, 4d\}$ has to be matched to g_{i-d} at least twice; otherwise, $v_i^3(S)$ would be smaller than 6ϵ or $4\epsilon + \gamma/2$, which are strictly less than $\gamma - 1$ by our choice of ϵ . Observe that if there is some $g \in \{g_1, \dots, g_{3d}\}$ that is not matched to any agent in $\{1, \dots, d\}$ for all rounds, then there must exist an agent $i \in \{1, \dots, d\}$ that is matched to $g' \notin \{g_1, \dots, g_{3d}\}$, and we can strictly improve its valuation by swapping out g' with g . Thus, we only need to consider when every goods in $\{1, \dots, 3d\}$ are matched to exactly one agent once. However, since R is a NO instance, we know that for all partition of $\{g_1, \dots, g_{3d}\}$ into d triplets $G_1, \dots, G_d$, there must exist some subset G_i in which the sum of its valuation is at most $\gamma - 1$. Thus, there exist an agent with valuation at most $\gamma - 1$.

Now, we construct an anytime optimal sequence S with $\text{OPT}(1) = 2\epsilon$, $\text{OPT}(2) = 4\epsilon$, and $\text{OPT}(3) = \gamma - 1$. Consider the allocation A in which

$$A_{ij} = \begin{cases} 3, & \text{if } i \in \{1, \dots, d\} \text{ and } j = 4d + i, \\ 2, & \text{if } i \in \{d+1, \dots, d+6\} \text{ and } j = i - d, \\ 1, & \text{if } i \in \{d+1, \dots, d+6\} \text{ and } j = 5d + \lceil (i-d)/3 \rceil, \\ 3, & \text{if } i \in \{d+7, \dots, 4d\} \text{ and } j = i - d, \\ 3, & \text{if } i \in \{4d+1, \dots, 5d\} \text{ and } j = i - d, \\ 1, & \text{if } i = 5d+1 \text{ and } j \in \{1, 2, 5d+3\}, \\ 1, & \text{if } i = 5d+2 \text{ and } j \in \{3, 4, 5d+3\}, \\ 1, & \text{if } i = 5d+3 \text{ and } j \in \{5, 6, 5d+3\}, \\ 0, & \text{otherwise.} \end{cases}$$

It is straightforward to verify that $v_i(A) \geq \gamma - 1$. By Lemma 2.1, we can convert A into a sequence S of three matchings where each $i \in N$ satisfy $v_i^3(S) \geq \gamma - 1$. Furthermore, since all the goods that are matched to each agent has value at least 2ϵ , we have $v_i^1(S) \geq 2\epsilon$ and $v_i^2(S) \geq 4\epsilon$. This implies that S is an anytime optimal sequence. Thus, $\mathcal{I}$ is a YES instance. $\square$

Theorem 4.4. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$, there always exist a sequence of matchings that is approximate anytime optimal. Furthermore, Algorithm 2 outputs a sequence of matchings S , in polynomial time, that satisfy*

$$b^t(S) \geq \text{OPT}(t) - 5m \cdot \max_{i \in N} \max_{g \in G} u(g), \quad \forall t \in [T].$$

Proof. Observe that since (P1) has $m^2 + 5m$ inequality constraints and $m^2 + 1$ variables, $m^2 + 1$ constraints will be tight at a vertex solution, meaning there are at most $5m$ non-zero entries in B , which implies that $d \leq 5m$.

Let n_{kt} be the value of n_k after round t . After each round $t \in [T]$, we claim that our choice of matching M^t maintains the invariant $\alpha_k t - n_{kt} \leq 1$ for all $k \in [d]$. If the invariant is kept, we have

$$\begin{aligned}
v_i^t(S) &= \sum_{k=1}^d u_i(M_k(i)) n_{kt} \\
&\geq \sum_{k=1}^d u_i(M_k(i)) (\alpha_k t - 1) \\
&\geq \sum_{k=1}^d u_i(M_k(i)) \alpha_k t - d \cdot \max_{g \in G} u_i(g) \\
&= \sum_{g_j \in G} u_i(g_j) t B_{ij} - d \cdot \max_{g \in G} u_i(g) \\
&\geq tb - d \cdot \max_{g \in G} u_i(g) \\
&\geq \text{OPT}(t) - 5m \cdot \max_{g \in G} u_i(g),
\end{aligned}$$

for all agents $i \in N$, where the fourth line is true because

$$\sum_{k=1}^d u_i(M_k(i)) \alpha_k = \sum_{k=1}^d \sum_{g_j \in G} u_i(g_j) \alpha_k (M_k)_{ij} = \sum_{g_j \in G} u_i(g_j) \left[\sum_{k=1}^d \alpha_k M_k \right]_{ij} = \sum_{g_j \in G} u_i(g_j) B_{ij}.$$

We are left to show that the invariant is kept after each round. Let $g_{kt} = (n_{kt} + 1)/\alpha_k$ for each $k \in [d]$. Suppose, for sake of contradiction, that $\alpha_k t - n_{kt} > 1$ for some $t \in [T]$ and some $k \in [d]$. By rearranging the terms, we have $t > (n_{kt} + 1)/\alpha_k = g_{kt}$.

For all other matching $M_l \neq M_k$, if M_l is not chosen for any round $s \leq t$, then we have $n_{lt} = 0 < \alpha_l t$. Otherwise, suppose that M_l is chosen for the n_{lt} time on round $s \leq t$, that is, $n_{lt} = n_{ls} = n_{l,s-1} + 1$. Since M_l is chosen over M_k , we must have $g_{l,s-1} \leq g_{k,s-1}$. Then, we have

$$\frac{n_{lt}}{\alpha_l} = \frac{n_{l,s-1} + 1}{\alpha_l} = g_{l,s-1} \leq g_{k,s-1} = \frac{n_{k,s-1} + 1}{\alpha_k} \leq \frac{n_{kt} + 1}{\alpha_k} < t$$

Thus, we have that $n_{lt} < \alpha_l t$ for all $l \in [d]$. Summing across all n_{lt} , we have

$$\sum_{l \in [d]} n_{lt} < \sum_{l \in [d]} \alpha_l t = t \sum_{l \in [d]} \alpha_l = t$$

where the last equality is due to the fact that the sum of the weights is 1. However, this is a contradiction because we select a matrix at every timestep, and thus $\sum_{l \in [d]} n_{lt}$ has to be t . $\square$

F Omitted Proofs in Section 5

We first define circulation with demand.

Definition F.1 (Circulation with demand). Let $G = (V, E)$ be a directed graph where each vertex $v \in V$ has a demand $d(v)$. A *circulation with demand* is a function $f : E \rightarrow \mathbb{R}$ that assigns non-negative value to each edge $(u, v) \in E$ such that

$$\sum_{(u,v) \in E} f(u, v) - \sum_{(v,w) \in E} f(v, w) = d(v), \quad \forall v \in V.$$

Lemma 5.1. *Let G' be the goods in a maximum matching. Then, for any matching M , there is a matching M_* that weakly Pareto dominates M and that the goods matched by M_* is a subset of G' .*

Proof. We prove by construction. Take any matching M and let M' be any maximum matching with goods in G' . If all agents in M is matched to some good in G' , then we are done. Let N_0 be the set of agents in M that are not matched to goods in M' and let G_0 be the set of goods in M' that are not matched to agents in M . Note that each agent $i \in N_0$ must be in M' ; otherwise, we can add $(i, M(i))$ to M' to increase its cardinality, contradicting to the maximality of M' .

We define an *augmenting graph* as follows. The goods G are the vertices of the augmenting graph and $(g, g') \in E$ if there exists an agent that receives g in M and g' in M' . Note that each vertex have at most one incoming edge and at most one outgoing edge, and that for each $i \in N_0$, we have $(M(i), M'(i)) \in E$. Furthermore, this edge cannot be part of a cycle in the augmenting graph because $M(i)$ is not in G' , thus it follows a non-cyclic path $P = (M(i) = g^1, \dots, g^k)$. We denote i^q to be the agent such that $M(i^q) = g^q$ and $M'(i^q) = g^{q+1}$.

We first process the agents in N_0 where no agent in M is matched to g^k on its path P . We modify M by removing (i^q, g^q) and adding (i^q, g^{q+1}) for each $q \in [k-1]$. This ensures that each agent in M who is on P is now matched to a good in G' , and that i^1 and g^k can be removed from N_0 and G_0 respectively.

We claim that after the processing step, we are done. Specifically, there is no agent in N_0 where an agent in M is matched to g^k on its path P . Suppose such an agent $i \in N_0$ exist. Since $|N_0| = |G_0|$, there must be a good $g' \in G_0$ and an agent i' in which $M'(i') = g'$. Since no agent is matched to g' in M (by definition of G_0) and no good is matched to i' in M (because this reduces to the previously processed case, which we assume are all processed), we can add (i', g') to M to increase its cardinality, which contradicts to its maximality. $\square$

Theorem 5.2. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$ with binary valuations, we can find an optimal sequence of matchings in polynomial time.*

Proof. We solve this with binary search on the optimal bottleneck value b . Let b be the guess of the optimal bottleneck value.

Let M be a maximum matching and let G' be the goods allocated in M . By Lemma 5.1, it is sufficient to consider only matchings formed with these goods at every timestep. Then, if $nb > |G'|T$, we automatically reject as there can be no outcome where all agents are satisfied at b timesteps.

Otherwise, we will consider the following circulation with demand problem.

For each agent $i \in N$, we create a vertex u_i with demand b and for each $g_j \in G'$, we create a vertex v_j with demand $-T$. Then, We add an edge (u_i, v_j) if $u_i(g_j) = 1$. Finally, we create a vertex i_0 with demand $|G'|T - nb$ and add an edge between i_0 and all nodes v_j for $g_j \in G'$.

We claim that there is a feasible circulation if and only if there is a sequence of matching with bottleneck value at least b .

($\Rightarrow$) Suppose there is a feasible circulation f . Note that since all the demands are integer-valued, the resulting circulation is also integer-valued. As such, we can consider the allocation A where $A_{ij} = f(u_i, v_j)$. We note that by our demand constraint, the sum of all rows are b and the sum of all columns is at most T . Then, we can add $n - |G'|$ empty columns (that represents ‘fake’ zero-valued goods) and by Lemma 2.1, there exist a sequence of matching S such that $v_i^T(S) \geq b$, which implies that $b^T(S) \geq b$.

($\Leftarrow$) Suppose there is a sequence of matching S with $b^T(S) \geq b$. By Lemma 5.1, we can assume that S is chosen such that for every timestep, $u_i(M_t(i)) = 1$ if and only if $M_t(i) \in G'$. Then, let A be the allocation that correspond to that matching. We note that every row sums up to at most T . Then the circulation $f(u_i, v_j) = A_{ij}$ for $i \in [n], j \in [|G'|]$ and $f(u_0, v_j) = T - \sum_{i \in [n]} A_{ij}$ for $j \in [|G'|]$ is feasible. $\square$

Theorem 5.3. *Given an instance $(N, G_1 \cup G_2, T, \{u_i\}_{i \in N})$ with two types of goods, we can find an optimal sequence of matchings in polynomial time.*

Proof. We solve this with binary search on the optimal bottleneck value b . Let b be the guess of the optimal bottleneck value. If $T \cdot \max_{g \in G} u_i(g) \leq b$ for some agent $i \in N$, we can immediately reject b . Otherwise, we will consider the following circulation with demand problem:

For each good $g \in G$, create a vertex g with demand T . For each agent $i \in N$, if agent i is indifferent between G_0 and G_1 , we create a vertex i_0 with demand $-T$ and add an edge (i_0, g) to all $g \in G$. Otherwise, agent i strictly prefers G_r for some $r \in \{0, 1\}$. Let $g_r \in G_r$, $g_{1-r} \in G_{1-r}$, and

$$k_i = \left\lceil \frac{b - T \cdot u_i(g_{1-r})}{u_i(g_r) - u_i(g_{1-r})} \right\rceil$$

be the minimum number of rounds that agent i needs to receive goods from G_r to achieve at least b in valuation. We create two vertices i_r and i_{1-r} with demand $-k_i$ and $-(T - k_i)$ respectively. We then add an edge (i_r, g) if $g \in G_r$ and another edge (i_{1-r}, g) for all $g \in G$. Finally, we create a source s with demand $-T(m - n)$ and add an edge (s, g) for all $g \in G$. We claim that there is a feasible circulation if and only if there is a sequence of matching with bottleneck value at least b .

($\Rightarrow$) Suppose there is a feasible circulation f . Note that since all the demands are integer-valued, the resulting circulation is also integer-valued. As such, we can consider the allocation A where

$$A_{ij} = \begin{cases} f(i_0, g_j), & \text{if agent } i \text{ is indifferent between } G_0 \text{ and } G_1, \\ f(i_r, g_j), & \text{if agent } i \text{ strictly prefers } G_r \text{ and } g_j \in G_r, \\ f(i_0, g_j) + f(i_{1-r}, g_j), & \text{if agent } i \text{ strictly prefers } G_r \text{ and } g_j \in G_{1-r}, \\ 0, & \text{otherwise.} \end{cases}$$

Let $i \in N$. If agent i is indifferent between G_0 and G_1 , then it trivially satisfy $v_i(A) \geq b$. Suppose agent i strictly prefers G_r . Then, we have

$$\begin{aligned} v_i(A) &= \sum_{g \in G_r} f(i_r, g) u_i(g) + \sum_{g \in G} f(i_{1-r}, g) u_i(g) \\ &\geq u_i(g_r) \sum_{g \in G_r} f(i_r, g) + u_i(g_{1-r}) \sum_{g \in G} f(i_{1-r}, g) \\ &= k_i \cdot u_i(g_r) + (T - k_i) \cdot u_i(g_{1-r}) \\ &\geq b, \end{aligned}$$

where the last inequality holds by our choice of k_i . Then, by Lemma 2.1, there exist a sequence of matching S such that $v_i^T(S) \geq b$, which implies that $b^T(S) \geq b$.

($\Leftarrow$) Suppose there is a sequence of matching S with $b^T(S) \geq b$. For each $r \in \{0, 1\}$, let L_r be a dynamic set that is initialized to contain T of each good $g \in G_r$, and define the operation $\text{remove} : \{0, 1\} \times \mathbb{N} \rightarrow \mathcal{P}(G \times \mathbb{N})$ such that $\text{remove}(r, c)$ removes c elements from L_r and returns the number of times each good from G_r is removed.

Let A be an allocation associated with the sequence S . We now construct the flow f . For $i \in N$, if agent i is indifferent between G_0 and G_1 , then we set

$$f(i_0, g_j) = \begin{cases} k, & \text{if } g_j \in G_0 \text{ and } (g_j, k) \in \text{remove}(0, A_{ij}), \\ k, & \text{if } g_j \in G_1 \text{ and } (g_j, k) \in \text{remove}(1, A_{ij}), \\ 0, & \text{otherwise.} \end{cases}$$

Otherwise, suppose that agent i strictly prefer G_r over G_{1-r} . Since $v_i(A) \geq b$, we know that the number of rounds agent i is allocated goods in G_r is at least k_i . Let

$$A_{ir} = \sum_{g_j \in G_r} A_{ij} \quad \text{and} \quad A_{i(1-r)} = \sum_{g_j \in G_{1-r}} A_{ij}.$$

We can set

$$f(i_{r'}, g_j) = \begin{cases} k, & \text{if } r' = r \text{ and } g_j \in G_r \text{ and } (g_j, k) \in \text{remove}(r, k_i), \\ k, & \text{if } r' = 1 - r \text{ and } g_j \in G_r \text{ and } (g_j, k) \in \text{remove}(r, A_{ir} - k_i), \\ k, & \text{if } r' = 1 - r \text{ and } g_j \in G_{1-r} \text{ and } (g_j, k) \in \text{remove}(1 - r, A_{i(1-r)} - k_i), \\ 0, & \text{otherwise.} \end{cases}$$

We note that thus far for all vertices corresponding to the agents, the sum of flow outgoing from the node is equal to its demand and for all nodes corresponding to the goods, the flow incoming is at most T . Thus, we can set the flow outgoing from node s appropriately. $\square$

Theorem 5.4. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$ with identical valuations, finding an optimal sequence of matchings is NP-complete.*

Proof. We will first define the Promise Balanced Partition problem and show it is NP-hard before proving Theorem 5.4 is NP-complete.

PROMISE BALANCED PARTITION (PBP)

Input: A list of distinct non-negative integer $E = \{e_1, \dots, e_k\}$. Let τ be the sum of the elements in E . It is guaranteed that for all multisets of size $k/2$, the sum of its elements does not equal $\tau/2$ if it contains duplicates.

Question: Is there a partition of E into two equal-sized subsets E_1, E_2 such that the sum of the elements in E_1 equals to the sum of the elements in E_2 ?

Let Φ be an instance of 1-IN-3-SAT. For each $i \in [n]$, we define two integers t_i and f_i that encode the assignments of the variable x_i to TRUE and FALSE, respectively. In particular, we define

$$t_i = (2n + 1)^{m+n-i} + \sum_{j: x_i \in c_j} (2n + 1)^{m-j} \quad \text{and} \quad f_i = (2n + 1)^{m+n-i} + \sum_{j: \bar{x}_i \in c_j} (2n + 1)^{m-j}.$$

We additionally define $t_{n+1} = \sum_{j=1}^m (2n + 1)^{m-j}$ and $f_{n+1} = 0$. Let

$$E = \{t_1, f_1, \dots, t_{n+1}, f_{n+1}\}.$$

Summing over all elements in E , we have

$$\tau = 2 \sum_{i=1}^n (2n + 1)^{m+n-i} + 4 \sum_{j=1}^m (2n + 1)^{m-j}$$

Note that it is helpful to view the construction in base $2n + 1$, as this makes several of the claims in the proof easier to verify. For example, in base $2n + 1$, it becomes immediately apparent that all elements in E are distinct.

Consider any multiset E' of size $n + 1$. We now show that if there exists an index $i \in [n + 1]$ such that E' contains zero or multiple occurrences of the elements in $\{t_i, f_i\}$, then the sum of E' is not equal to $\tau/2$. Since any multiset with a duplicated element necessarily violates this condition, it follows that no multiset of size $|E|/2$ with duplicates can sum to $\tau/2$, thereby satisfying the promise condition.

Let i^* be the smallest such index, and for all $i < i^*$, let $\ell_i \in \{t_i, f_i\}$ denote the unique element from the pair that appears in E' . If E' contains multiple occurrences of elements from $\{t_{i^*}, f_{i^*}\}$, let $\ell_{i^*}^1$ and $\ell_{i^*}^2$ denote two such occurrences. Then,

$$\sum_{\ell \in E'} \ell \geq \ell_1 + \dots + \ell_{i^*-1} + \ell_{i^*}^1 + \ell_{i^*}^2 > \frac{\tau}{2},$$

where the last inequality holds because

$$\begin{aligned}
\ell_1 + \dots + \ell_{i^*-1} + \ell_{i^*}^1 + \ell_{i^*}^2 &= (11 \dots 11 \underline{2} 00 \dots 00 \mid \tilde{3} \tilde{3} \dots \tilde{3} \tilde{3})_{2n+1} \\
&> (11 \dots 11 \underline{1} 33 \dots 33 \mid 33 \dots 33)_{2n+1} \\
&> (11 \dots 11 \underline{1} 11 \dots 11 \mid 22 \dots 22)_{2n+1} \\
&= \tau/2.
\end{aligned}$$

In the expressions above, numbers are written in base $2n + 1$. The underlined digit corresponds to the position indexed by $m + n - i^*$. The bar separates the representation into two parts: the left portion has n digits, and the right portion has m digits. A tilde over a digit indicates that the digit is at most that value.

If E' contains no occurrence of elements from $\{t_{i^*}, f_{i^*}\}$, let $\bar{E}' = E' \setminus \{\ell_1, \dots, \ell_{i^*-1}\}$. Then for each $\ell \in \bar{E}'$, we have

$$\begin{aligned}
\ell &= (00 \dots 00 \underline{0} \tilde{1} \tilde{1} \tilde{1} \dots \tilde{1} \tilde{1} \mid \tilde{3} \tilde{3} \dots \tilde{3} \tilde{3})_{2n+1} \\
&\leq (00 \dots 00 \underline{0} 11 \tilde{1} \tilde{1} \dots \tilde{1} \tilde{1} \mid \tilde{3} \tilde{3} \dots \tilde{3} \tilde{3})_{2n+1} \\
&< (00 \dots 00 \underline{0} 1200 \dots 00 \mid 00 \dots 00)_{2n+1} \\
&= (2n + 1)^{m+n-i^*-1} + 2(2n + 1)^{m+n-i^*-2}.
\end{aligned}$$

Furthermore, observe that

$$\begin{aligned}
\sum_{i=1}^{i^*-1} \ell_i + (2n + 1)^{m+n-i^*} &= (11 \dots 11 \underline{1} 00 \dots 00 \mid \tilde{3} \tilde{3} \dots \tilde{3} \tilde{3})_{2n+1} \\
&< (11 \dots 11 \underline{1} 11 \dots 11 \mid 55 \dots 55)_{2n+1} \\
&< \frac{\tau}{2} + 3 \sum_{j=1}^m (2n + 1)^{m-j}.
\end{aligned}$$

Combining these results, we obtain

$$\sum_{\ell \in S'} \ell < \sum_{i=1}^{i^*-1} \ell_i + n(2n + 1)^{m+n-i^*-1} + 2n(2n + 1)^{m+n-i^*-2} < \frac{\tau}{2},$$

where the final inequality follows immediately by substituting the earlier bound and observing that

$$n(2n + 1)^{m+n-i^*-1} + 2n(2n + 1)^{m+n-i^*-2} + 3 \sum_{j=1}^m (2n + 1)^{m-j} < (2n + 1)^{n+m-i^*}.$$

Thus, our promise condition is satisfied.

We now show that the constructed PBP instance E is a YES instance if and only if the 1-IN-3-SAT instance Φ is a YES instance.

($\Rightarrow$) Suppose E is a YES instance, and let E_1, E_2 be a valid partition. Without loss of generality, assume that $t_{n+1} \in E_1$. Furthermore, by construction, for each $i \in [n + 1]$, the set E_1 must contain exactly one element from the pair $\{t_i, f_i\}$. We define a truth assignment as follows: for each $i \in [n]$, set $x_i = \text{TRUE}$ if $t_i \in E_1$ and $x_i = \text{FALSE}$ if $f_i \in E_1$. Then,

$$\begin{aligned}
\sum_{\ell \in E_1} \ell - t_{n+1} &= (11 \dots 11 \mid 22 \dots 22)_{2n+1} - (00 \dots 00 \mid 11 \dots 11)_{2n+1} \\
&= (11 \dots 11 \mid 11 \dots 11)_{2n+1}.
\end{aligned}$$

Here, the least significant m digits (after the vertical bar) count how many literals are satisfied in each clause. Since each digit is exactly 1, this implies that, under the assignment $x_1, \dots, x_n$, each clause of Φ is satisfied by exactly one literal. Hence, Φ is a YES instance.

($\Leftarrow$) Suppose Φ is a YES instance, and let $x_1, \dots, x_n$ be a satisfying assignment such that exactly one literal is satisfied in each clause. For each $i \in [n]$, let $\ell_i = t_i$ if $x_i = \text{TRUE}$ and $\ell_i = f_i$ if $x_i = \text{FALSE}$. Additionally, let $\ell_{n+1} = t_{n+1}$, and define the set $E = \{\ell_1, \dots, \ell_{n+1}\}$. Since exactly one of $\{t_i, f_i\}$ is selected for each $i \in [n]$, each of the n most significant digit receives exactly one contribution. Moreover, because the assignment satisfies each clause exactly once, each of the m least significant digit also receives exactly one contribution. Therefore, the sum of the elements $\ell_1 + \dots + \ell_n = (11 \dots 11 \mid 11 \dots 11)_{2n+1}$. Adding ℓ_{n+1} to it, we get

$$\begin{aligned} \sum_{\ell \in E_1} \ell &= (11 \dots 11 \mid 11 \dots 11)_{2n+1} + (00 \dots 00 \mid 11 \dots 11)_{2n+1} \\ &= (11 \dots 11 \mid 22 \dots 22)_{2n+1} \\ &= \tau/2. \end{aligned}$$

We note that $|E_1| = |E_2| = |E|/2$ by construction. Hence, E is a YES instance.

We now complete the proof by reducing from the PBP problem. Given a set $E = \{e_1, \dots, e_k\}$ from an instance of PBP, we construct an instance $\mathcal{I} = (N, G, T, \{u_i\}_{i \in N})$ of ERM with $n = m = k$, $T = n/2$, $\kappa = \tau/2$, and identical valuations defined by $u_i(g_j) = e_j$ for all $i \in [n]$ and $j \in [m]$. We claim that there exists a balanced partition E_1, E_2 of E if and only if there exists a sequence of matchings in $\mathcal{I}$ that achieves a bottleneck value of κ .

($\Rightarrow$) Suppose E is a YES instance, and let E_1, E_2 be a balanced partition. Since there are $T = n/2$ rounds, construct an allocation in which the first $n/2$ agents receive, over the course of the T rounds, all the goods in E_1 , one per round, and the remaining $n/2$ agents receive all the goods in E_2 , again one per round. Under this allocation, each agent receives exactly T goods and accumulates a total value of $\tau/2 = \kappa$, and each good is matched once per round and appears in exactly T rounds. By Lemma 2.1, there exists a sequence $S \in \mathbb{S}^T$ such that $v_i^T(S) \geq \kappa$ for all $i \in N$. Hence, $(\mathcal{I}, \kappa)$ is a YES instance.

($\Leftarrow$) Suppose $(\mathcal{I}, \kappa)$ is a YES instance, and let $S \in \mathbb{S}^T$ be a sequence of matchings such that each agent $i \in N$ receives total value $v_i^T(S) \geq \kappa$. Since each good appears in exactly $T = n/2$ rounds and all valuations are identical, the total value across all agents is exactly $n\kappa$. But since each agent receives at least κ , and there are n agents, it follows that each agent must receive exactly κ . Now consider the multiset of goods that agent 1 receives under S . By the promise condition, any multiset of size $n/2 = k/2$ whose sum is $\kappa = \tau/2$ must consist of distinct elements. Therefore, the goods assigned to agent 1 are all distinct. Let $E_1 \subset E$ be the set of integers corresponding to the goods received by agent 1, and let $E_2 = E \setminus E_1$. Then, $|E_1| = |E_2| = k/2$ and the sum of each set is $\tau/2$. Thus, E_1 and E_2 form a balanced partition. Hence, E is a YES instance. $\square$

Theorem 5.5. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$ with identical valuations and $T = kn$ for some $k \in \mathbb{Z}$, we can find an optimal sequence of matchings in polynomial time.*

Proof. Let $G_* \subseteq G$ be the top n most valuable goods and consider the allocation A that gives k copies of each good in G_* to each agent $i \in N$. Then, we have $v_i(A) = v_{i'}(A)$ for all agents $i, i' \in N$. Suppose, for sake of contradiction, that A is not optimal. Then, there exist some other allocation A' such that $\min_i v_i(A') > \min_i v_i(A)$. This implies that

$$nk \sum_{g \in G_*} u_i(g) \geq \sum_{i \in N} v_i(A') \geq n \cdot \min_{i \in N} v_i(A') > n \cdot \min_{i \in N} v_i(A) = \sum_{i \in N} v_i(A) = nk \sum_{g \in G_*} u_i(g),$$

where the first inequality is true because there is no way to achieve strictly greater utilitarian value than by assigning out the top n most valuable goods in every round. Since this leads to a contradiction, we conclude that A is optimal. $\square$

Theorem 5.6. *Given an instance $(N, G, T, \{u_i\}_{i \in N})$ with identical valuations, we can find, in polynomial time, a sequence of matchings S that satisfy*

$$b^t(S) \geq \text{OPT}(t) - \Delta, \quad \forall t \in [T],$$

where Δ is the difference in value between the most valuable good and the n -th most valuable good.

Proof. We first describe the polynomial-time algorithm that will return us a sequence of matchings S . At each round $t \in [T]$, sort the agents in increasing order of cumulative valuation up till round $t - 1$. Then, in this order, let each agent choose their favorite good and allocate it to them. Repeat this process until all rounds are completed. Note that since we are considering the setting with identical valuations, it suffices to only look at the top n -valued goods—no agent will choose any of the other (lower-valued) goods in any round.

Fix any round $t \in [T]$. It is easy to observe that

$$\frac{1}{n} \cdot \sum_{i \in N} v_i^t(S) \geq \text{OPT}(t), \quad (1)$$

since our algorithm allows agents to select their favorite good in increasing order of cumulative value up till round t . Let the bottleneck agent at round t be i , that is, $b^t(S) = v_i^t(S)$. Let N' be the set of agents who picks a good before agent i at some point. Then, for each agent $i' \in N'$, let $s_0 \leq t$ be the last round in which i' picks a good before i . By the algorithm, we have

$$v_{i'}^{s_0-1}(S) \leq v_i^{s_0-1}(S) \quad \text{and} \quad u(M^s(i')) \leq u(M^s(i)) \quad \text{for all rounds } s \in [s_0 + 1, t].$$

By the first inequality, we have

$$\begin{aligned} v_{i'}^{s_0}(S) &= v_{i'}^{s_0-1}(S) + u(M^{s_0}(i')) \\ &\leq v_i^{s_0-1}(S) + u(M^{s_0}(i)) - u(M^{s_0}(i)) + u(M^{s_0}(i')) \\ &= v_i^{s_0}(S) - u(M^{s_0}(i)) + u(M^{s_0}(i')) \\ &\leq v_i^{s_0}(S) + \Delta. \end{aligned}$$

Combining this result with the second inequality, we have

$$v_{i'}^t(S) = v_{i'}^{s_0}(S) + \sum_{s=s_0+1}^t u(M^s(i')) \leq v_i^{s_0}(S) + \Delta + \sum_{s=s_0+1}^t u(M^s(i)) = v_i^t(S) + \Delta.$$

Furthermore, for each $i' \in N \setminus N'$, we have

$$v_{i'}^t(S) = \sum_{s=1}^t u(M^s(i')) \leq \sum_{s=1}^t u(M^s(i)) = v_i^t(S) \leq v_i^t(S) + \Delta.$$

Taking the average over all agents and using (1), we get

$$\text{OPT}(t) \leq \frac{1}{n} \cdot \sum_{i' \in N} v_{i'}^t(S) \leq v_i^t(S) + \Delta = b^t(S) + \Delta$$

as desired. □