Proportionally Fair Online Allocation of Public Goods

SIDDHARTHA BANERJEE, Cornell University
VASILIS GKATZELIS, Drexel University
SAFWAN HOSSAIN, University of Toronto
BILLY JIN, Cornell University
EVI MICHA, University of Toronto
NISARG SHAH, University of Toronto

We study the problem of designing online algorithms for the fair allocation of public goods to \( N \) agents, over a sequence of \( T \) rounds. In each round \( t \), a new public good becomes available and each agent \( i \)'s value \( v_{i,t} \) for this good is revealed to the algorithm, which needs to make an irrevocable decision regarding how much to invest in this good, without knowing the agents’ values for the public goods to arrive in future rounds. Each agent \( i \) receives a fraction of \( v_{i,t} \) that is proportional to that investment. The problem becomes interesting when the algorithm is bound to invest no more than a total budget of \( B \) across all rounds. We study proportional fairness as our notion of fairness, which informally demands that each group of agents receive treatment proportionate to its size and the cohesiveness of its preferences.

First, we prove that an \( O(\log N) \) approximation of proportional fairness can be achieved in the special case with unit budget and binary agent values \( (v_{i,t} \in \{0, 1\}) \), and that this is tight. However, in the general model, no online algorithm can achieve approximation ratio better than \( O(T/B) \). Motivated by the recent surge of work on online algorithms with predictions, we study the extent to which online algorithms can approximate proportional fairness when enhanced with predictions of the total values that agents will have across all rounds. Our main result is an online algorithm that achieves an \( O(\log(T/B)) \) approximation of proportional fairness when the predictions are accurate, which we prove to be the best possible. We further parameterize the performance of the algorithm as a function of the errors in the predictions, and show that the performance degrades smoothly as the errors increase.
1 INTRODUCTION

The classic fair division problem wherein a set of goods must be distributed amongst a group of agents (e.g., for inheritance, divorce settlement, etc.) has long garnered interest from various research communities, dating back to Steinhaus [1948]. Each of the agents may perceive the value of a good differently, and the main goal in this problem is to reach an allocation of goods that is considered fair. Most settings assume the goods to be private, implying only their owner can benefit from them, and this is often the main source of tension, as one agent’s happiness is at the expense of another. This tension exists irrespective of whether the goods are divisible and can be divided across multiple agents, or indivisible and each good must be given entirely to a single agent. In either case, an agent can enjoy only the (fractions of) the goods that they own.

However, in many cases the goods being allocated are better modelled as public goods, whose benefit can be shared across multiple agents (e.g., a highway or a park). While agents with similar interests can now mutually benefit, a principal, tasked with choosing which goods to invest in, needs to strike a balance between agents with different interests. For example, a municipality deciding between a new infrastructure project or green space must account for the preferences of the local residents; some may benefit directly from the new infrastructure, while others may advocate for a greener city. Similarly, when a company is deliberating regarding which candidates to hire, each team within the company may have its own preferences based on the skill sets it values. In contrast to the private goods setting, where the biggest source of tension is between agents who have similar preferences and are thus competing for the same goods, in the public goods setting, it is agents with different preferences that are in direct opposition.

Motivated by applications such as participatory budgeting, committee selection, and shared memory allocation, the literature has moved beyond the private goods model to study fair allocation for public goods (e.g., [Conitzer et al., 2017, Fain et al., 2016, 2018, Friedman et al., 2019, Kunjir et al., 2017, Munagala et al., 2021, Peters et al., 2020]). These works have focused on notions of fairness such as (approximations of) the core, or the (approximate) Nash bargaining solution, both of which are implied by (approximate) proportional fairness, which is the focus of this paper. Unlike other fairness notions, like envy-freeness or proportionality, which essentially disregard efficiency considerations, proportional fairness strikes a balance between fairness and efficiency. For example, a proportionally fair outcome must be Pareto efficient, meaning that no other outcome can benefit an agent without being worse for another agent.

One limitation of this literature on public goods is that it consider “one-shot” settings, where one assumes that the algorithm has full access to the agents’ preferences for all goods before making allocation decisions. In reality, goods often become available over time, and agents may be unaware of their future options, or their value for these future goods, until they become available. For example, a city planner’s decisions regarding how to invest each year’s budget, or a company’s decisions regarding whom to hire, take place over a sequence of rounds, with limited information regarding the options that will arise in the future. To capture the dynamic nature of fair division problems in practice, numerous recent works have focused on the design of online algorithms for fair resource allocation (e.g., [Banerjee et al., 2022, Barman et al., 2022, Benade et al., 2018, Freeman et al., 2017, Gkatzelis et al., 2021, Zeng and Psomas, 2020]. However, with very few exceptions [Freeman et al., 2017], this literature has focused on the private goods setting.

In this paper we address this gap in the literature by studying a natural online fair division model involving public goods that become available over a sequence of rounds. Our goal is to design an algorithm that makes irrevocable decisions regarding how much to invest in each good, aiming to reach an outcome that optimizes for proportional fairness as best as possible in the worst case. We first exhibit the obstacles that arise if the algorithm has no information regarding the future values
(except in a special case), and then we propose online algorithms, enhanced with predictions of the future, that overcome these obstacles.

1.1 Our Results

We study the design of online algorithms for the fair allocation of public goods to \( N \) agents, over a sequence of \( T \) rounds. In each round \( t \), a new public good (which we refer to as good \( t \)) arrives and the algorithm learns the value \( v_{i,t} \) of each agent \( i \) for this good. Using this information, the algorithm needs to make an irrevocable decision regarding an amount \( x_t \in [0, 1] \) to invest in this good, before we reach round \( t + 1 \); this value \( x_t \) can also be interpreted as the probability of implementing the public good in round \( t \). Each agent \( i \) then receives value \( v_{i,t} \cdot x_t \) from this investment, and the total value derived by an agent is the sum of the values gained in each round. Clearly, the algorithm would like to increase the \( x_t \)'s as much as possible, but the algorithm is limited by a total budget constraint: \( \sum_t x_t \leq B \).

Informally, the budget constraint forces the algorithm to invest more in goods that are highly valued by many agents, which is challenging in an online setting when future goods and agent values for them are unknown.

We seek algorithms that balance the preferences of different agents fairly. Specifically, we focus on the quantitative fairness objective of proportional fairness (see Definition 2.1), which can be computed once all agent values are known (but the algorithm must optimize it, approximately, in an online fashion). Our results are divided into two technical sections.

Public Goods with Binary Valuations (Section 3). As a warm-up, we first focus on the interesting special case where agent values are binary (i.e., \( v_{i,t} \in \{0, 1\} \) for all agents \( i \) and goods \( t \)) and the budget is \( B = 1 \). Note that binary values correspond to agents either liking a good or not, while unit budget forces \( \sum_t x_t \leq 1 \), meaning that \( x_t \) can also be viewed as the fraction of an available resource invested in good \( t \).

Despite its apparent simplicity, this class of instances already exhibits some of the obstacles that arise in the online fair division of public goods. Specifically, our first result shows that no algorithm can achieve better than an \( O(\log N) \) approximation of proportional fairness. In fact, this result holds even if the algorithm knows in advance how many goods each agent is going to like (i.e., the sum of each agent’s values) and what the horizon is going to be (i.e., the total number of rounds, \( T \)). The main obstacle is that the algorithm cannot foresee whether some future good will be, say, liked by all the agents, in which case the algorithm should reserve its budget for that good.

On the positive side, we complement this impossibility result by presenting an algorithm that matches this bound by achieving an \( O(\log N) \) approximation of proportional fairness. Interestingly, the algorithm does not require any information regarding the horizon \( T \) or the number of goods that each agent is going to like. Note that, no matter how large \( T \) is, this bound depends only on the number of agents \( N \).

General Public Goods and the Role of Predictions (Section 4). Then, we move on to the main result of our paper for the general case with arbitrary \( v_{i,t} \geq 0 \) and \( B \geq 0 \). In contrast to the aforementioned special case, for this more general setting it is impossible to achieve any approximation of proportional fairness better than the trivial \( T/B \) (achieved by just uniformly distributing the budget across the rounds), even for a single agent, unless the algorithm has some additional information regarding the agents’ future values.

Following the literature focusing on “algorithms with predictions” [Lykouris and Vassilvitskii, 2021, Mitzenmacher and Vassilvitskii, 2021] and its recent application to the online fair division of private goods [Banerjee et al., 2022], we assume that the algorithm is provided with a prediction regarding each agent \( i \)'s total value over the \( T \) rounds, i.e., a prediction \( \tilde{V}_i \) of the quantity \( V_i = \sum_t v_{i,t} \). Using this information, we provide an algorithm that improves the proportional fairness
approximation from $T/B$ to $O(\log(T/B))$ if the predictions are reasonably accurate (in particular, as long as the multiplicative distortion $\tilde{V}_i/V_i$ on the prediction for each agent $i$ is independent of the problem parameters $T, N, B$). On the other hand, we show this proportional fairness approximation to be the best possible, by exhibiting a hardness proof which shows that no online algorithm can achieve an approximation of $o(\log(T/B))$ even if provided with perfect predictions ($\tilde{V}_i = V_i$ for each agent $i$). In fact, we provide general bounds, parameterized by the prediction error, showing that the algorithm’s performance degrades gracefully when provided with inaccurate predictions.

1.2 Related Work

**Online allocation of private goods.** The majority of the prior work on online fair division has focused on allocating private goods. In the absence of any information regarding the agents’ future values, this mostly leads to pessimistic results suggesting that achieving even basic notions of fairness needs to come at the cost of extreme inefficiency (e.g., [Benade et al., 2018, Zeng and Psomas, 2020]). However, the assumption that the algorithm has absolutely no information regarding future values is rather unrealistic in most real world applications, since there is an abundance of historical data available that can be used to at least formulate reasonable predictions regarding what these values may be.

Even beyond the context of fair division, this shortcoming in the worst-case analysis of online algorithms has motivated a surge of work on “algorithms with predictions” [Mitzenmacher and Vassilvitskii, 2021]. This line of work assumes that the online algorithm is augmented with predictions regarding the future input, and the goal is to use these predictions as a guide. An ideal solution would be an algorithm that performs well when the predictions are accurate, but still maintains some non-trivial worst-case guarantees even when the predictions are inaccurate.

Recently, Banerjee et al. [2022] leverage this approach to design online algorithms for fair division of private goods, which achieve improved fairness guarantees with respect to the Nash social welfare objective. Their algorithm uses a prediction of the total value of each agent over all goods. This is related to work which assumes that the values of each agent are normalized to add up to 1 [Barman et al., 2022, Gkatzelis et al., 2021] or that they are drawn randomly from a normalized distribution [Bogomolnaia et al., 2019].

Our work builds on the work of Banerjee et al. [2022], and extends their techniques (specifically, their set-aside greedy algorithm) to find fair online allocations of public goods. Reinforcing their findings with private goods, we show that predictions also help achieve improved fairness in online settings with public goods.

**Allocation to public goods.** Much of the literature on public goods focuses on the offline setting, where all the goods and agent preferences over these goods are known in advance. In the most canonical setting, a fixed budget is to be portioned between the goods, requiring us to decide on the allocation $x_t \geq 0$ to each good $t$ such that $\sum_t x_t \leq 1$, and agents have approval preferences for the goods which scale linearly in the $x_t$’s, making the setting precisely the offline version of the special case we study in Section 3. This setting has been studied under various names, such as probabilistic voting [Bogomolnaia et al., 2005], fair mixing [Aziz et al., 2019a], fair sharing [Duddy, 2015], and portioning [Brandl et al., 2021]. With $x_t$ interpreted as the probability of selecting good $t$, this can be viewed as randomized single-winner voting, where the goods are the candidates and the agents are the voters.

An extension to this is multi-winner voting, where the goal is to select a subset of candidates of a given size $k$ (this corresponds to $B = k$ in our model). Here, various fairness notions have been studied for offline solutions which require every group of agents to have representation in the selected committee, with larger and more cohesive groups having better representation. Example
notions includes justified representation (JR), extended justified representation (EJR) [Aziz et al., 2017], proportional justified representation (PJR) [Sánchez-Fernández et al., 2017], proportionality degree [Skowron, 2021], and stability [Cheng et al., 2020, Jiang et al., 2020].

Multi-winner voting is in turn a special case of fair public decision-making [Conitzer et al., 2017] and participatory budgeting; in the latter, each candidate has a cost and the goal is to select a subset of candidates with total cost at most a given budget. Fain et al. [2016] proposed a polynomial-time algorithm for finding an outcome in the core for participatory budgeting via the Lindahl equilibrium under a suitable class of agent preferences; in their model, the core is no longer implied by proportional fairness, but they show that it is nonetheless closely related. Fain et al. [2018] studied a model of public goods generalizing both fair public decision-making and participatory budgeting, and achieved different approximations to the core under various constraints on feasible outcomes by maximizing objective functions closely related to the Nash welfare and proportional fairness. To the best of our knowledge, the only work to consider online allocation to public goods is that of Freeman et al. [2017], who consider optimizing the Nash welfare in an online setting similar to ours. However, they do not provide any approximation guarantees; instead, they study natural online rules from an axiomatic viewpoint. Such an axiomatic approach is also taken by the literature on voting over rounds [Bredereck et al., 2020a,b, Lackner, 2020]; results from this literature do not have any overlap with our work.

**Primal-dual analysis.** Finally, we remark that our main positive result is derived using a primal-dual-style analysis, which is common in the online resource allocation literature (see the survey of Buchbinder and Naor [2009] for an excellent overview). Almost all of this work deals with additive objective functions. Two notable exceptions to this are the work of [Devanur and Jain, 2012], who show how to extend these approaches to non-linear functions of additive rewards, as well as Azar et al. [2016], who consider a variant of the proportional allocation objective, but require additional boundedness conditions on the valuations. Directly applying their guarantees in our setting gives poor bounds (although this is unsurprising since these approaches do not use predictions, which, as Proposition 4.1 shows, implies strong hardness results).

## 2 MODEL

When working with fractions $x/y$ with $x, y \geq 0$, we adopt the convention that $x/y = 1$ when both $x = y = 0$, while $x/y = +\infty$ if $y = 0$ and $x > 0$. For $k \in \mathbb{N}$, define $[k] = \{1, \ldots, k\}$ and $H_k = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{k}$ to be the $k$-th harmonic number.

We study an online allocation problem in which there are $N$ agents and $T$ rounds. Our algorithms always know $N$, but we study both horizon-aware algorithms which know $T$ in advance and horizon-independent algorithms which do not.

**Online arrivals.** In each round $t \in [T]$, a public good arrives, which we refer to as good $t$ for simplicity. Upon its arrival, we learn the value $v_{i,t} \geq 0$ of every agent $i \in [N]$ for it.

**Online allocations.** When good $t$ arrives, the online algorithm must irrevocably decide the allocation $x_t \in [0, 1]$ to good $t$, before the next round. We use $x = (x_t)_{t \in [T]}$ to denote the final allocation computed by the online algorithm. In the absence of any further constraints, the decision would be simple: allocate as much as possible to every good by setting $x_t = 1$ for each $t \in [T]$. We consider an overall budget constraint: $\sum_{t=1}^T x_t \leq B$, where $B \geq 0$ is a fixed budget known to the online algorithm in advance.

---

1In this work, we focus algorithms which make these decisions deterministically.
Linear agent utilities. Finally, we assume that an allocation $x_t$ to good $t$ simultaneously yields utility $u_{i,t} \cdot x_t$ to every agent $i$. Moreover, we assume that agent utilities are additive across goods, i.e., the final utility of agent $i$ is given by $u_i(x) = \sum_{t=1}^{T} u_{i,t} \cdot x_t$.

While linear utilities is a somewhat restrictive assumption, we note that it admits several natural interpretations depending on the application of interest. In applications like budget division, each public good $t$ is a project (e.g., an infrastructure project), and $x_t$ is the amount of a resource (e.g., time or money) invested in the project. In applications such as participatory budgeting or voting, each public good $t$ is an alternative or a candidate, and $x_t$ is the (marginal) probability of it being selected (indeed, one can compute a lottery over subsets of goods of size at most $B$ under which the marginal probability of selecting each good $t$ is precisely $x_t$).

### 2.1 Proportional Fairness

We want the allocation $x$ computed by our online algorithm to be fair. In this work, we use the notion of proportional fairness, which is a quantitative fairness notion that was first proposed in the context of rate control in communication networks [Kelly et al., 1998].

**Definition 2.1 (Proportional Fairness).** For $\alpha \geq 1$, allocation $x$ is called $\alpha$-proportionally fair if, for every other allocation $w$, we have

$$\frac{1}{N} \sum_{i=1}^{N} \frac{u_i(w)}{u_i(x)} \leq \alpha.$$ 

If $x$ is 1-proportionally fair, we simply refer to it as proportionally fair\(^2\). We say that an online algorithm is $\alpha$-proportionally fair if it always produces an $\alpha$-proportionally fair allocation.

It is known that in the offline setting, where all agent values are known up front, a 1-proportionally fair allocation $x$ always exists, and this is the lowest possible value of proportional fairness [Fain et al., 2018].

It is also known that proportional fairness is a strong guarantee that implies several other guarantees sought in the literature. We point out two such examples.

**Proportional fairness implies the core.** For $\alpha \geq 1$, allocation $x$ is said to be in the $\alpha$-core if there is no subset of agents $S$ and allocation $w$ such that $\frac{|S|}{N} \cdot u_i(w) \geq \alpha \cdot u_i(x)$ for all $i \in S$ and at least one of these inequalities is strict. We say that an online algorithm is $\alpha$-core if it always produces an allocation in the $\alpha$-core. The following is a well-known relation between proportional fairness and the core.

**Proposition 2.2.** For $\alpha \geq 1$, every $\alpha$-proportionally fair allocation is in the $\alpha$-core.

**Proof.** If an $\alpha$-proportionally fair allocation $x$ is not in the $\alpha$-core, then by definition, there exists a subset of agents $S$ and an allocation $w$ such that $\frac{|S|}{N} u_i(w) \geq \alpha \cdot u_i(x)$ for all $i \in S$ and at least one of these inequalities is strict. Then, $\frac{u_i(w)}{u_i(x)} \geq \alpha \cdot \frac{|S|}{N}$ for all $i \in S$ and at least one of these inequalities is strict, which implies

$$\sum_{i \in S} \frac{u_i(w)}{u_i(x)} \geq \alpha \cdot N \Rightarrow \frac{1}{N} \sum_{i=1}^{N} \frac{u_i(w)}{u_i(x)} \geq \frac{1}{N} \sum_{i \in S} u_i(w) > \alpha,$$

contradicting the fact that $x$ is $\alpha$-proportionally fair. \(\square\)

\(^2\)Proportional fairness is typically defined as $\frac{1}{N} \sum_{i=1}^{N} \frac{u_i(w)-u_i(x)}{u_i(x)} \leq 0$, which is 1-proportional fairness in our definition.
Proportional fairness implies optimal Nash welfare. A common objective function studied in multi-agent systems is the Nash social welfare, which aggregates individual agent utilities into a collective measure by taking the geometric mean. That is, the Nash social welfare of allocation \( x \) is given by \( \text{NSW}(x) = \left( \prod_{i=1}^{N} u_i(x) \right)^{1/N} \). For \( \alpha > 1 \), we say that allocation \( x \) achieves an \( \alpha \)-approximation of the Nash welfare if \( \frac{\text{NSW}(w)}{\text{NSW}(x)} \leq \alpha \) for all allocations \( w \). We say that an online algorithm achieves an \( \alpha \)-approximation of the Nash welfare if it always produces an allocation that achieves an \( \alpha \)-approximation of the Nash welfare. It is also well-known that \( \alpha \)-proportional fairness implies an \( \alpha \)-approximation of the Nash welfare (in particular, a proportionally fair allocation has the maximum possible Nash welfare).

**Proposition 2.3.** For \( \alpha > 1 \), if allocation \( x \) is \( \alpha \)-proportionally fair, then \( x \) achieves an \( \alpha \)-approximation of the Nash welfare.

**Proof.** Take any allocation \( w \). The result follows by observing that

\[
\frac{\text{NSW}(w)}{\text{NSW}(x)} = \left( \prod_{i=1}^{N} \frac{u_i(w)}{u_i(x)} \right)^{1/N} \leq \frac{1}{N} \sum_{i=1}^{N} \frac{u_i(w)}{u_i(x)} \leq \alpha,
\]

where the second transition is the AM-GM inequality. \( \Box \)

We remark that the upper bounds derived in this work hold for the stronger notion of proportional fairness, while the lower bounds hold even for the weaker notion of Nash welfare approximation.

### 2.2 Set-Aside Greedy Algorithms

In order to compute (approximately) proportionally fair allocations, we consider a family of online algorithms, called Set-Aside Greedy Algorithms. Recent work [Banerjee et al., 2022, Barman et al., 2022] has demonstrated how such algorithms can be used to get strong performance guarantees for online allocation of private goods; we show that with non-trivial modifications, they also achieve compelling fairness guarantees for allocating public goods.

At a high level, an algorithm in this family works as follows. It divides the overall budget \( B \) into two equal portions.

1. The first half, called the set-aside budget, is used to allocate \( y_t \in [0, 1] \) to each good \( t \) in such a manner that \( \sum_{t=1}^{T} y_t \leq B/2 \) and this portion of the allocation guarantees each agent \( i \) a certain minimum utility of \( \Delta_i \) (i.e., \( \sum_{t=1}^{T} v_{i,t} \cdot y_t \geq \Delta_i \)). For example, if \( y_t = B/(2T) \) for each \( t \in [T] \), then we can use \( \Delta_i = \frac{B}{2T} \cdot \sum_{t=1}^{T} v_{i,t} \). This ensures that in the proportional fairness definition (Definition 2.1), the ratio \( \frac{u_i(w)}{u_i(x)} \) does not become excessively large for any agent \( i \).

2. The second half, called the greedy budget, is used to allocate \( z_t \in [0, 1 - y_t] \) to each good \( t \) in such a manner that \( \sum_{t=1}^{T} z_t \leq B/2 \). This portion of the budget is used for a truly online optimization in a greedy-like fashion.

The final allocation to good \( t \) is set to \( x_t = y_t + z_t \). We refer to \( y_t \) and \( z_t \) as semi-allocations to good \( t \). An important quantity in our analysis is the promised utility to an agent.

**Definition 2.4 (Promised Utilities).** Consider the arrival of good \( t \). From the set-aside budget, semi-allocations \( y_1, \ldots, y_T \) (already realized or not) guarantee that by the end, they will provide each agent \( i \) a utility of at least \( \Delta_i \leq \sum_{t=1}^{T} v_{i,t} \cdot y_t \). From the greedy budget, the algorithm has already set semi-allocations \( z_1, \ldots, z_{t-1} \) in the previous steps, and needs to now decide \( z_t \). At this stage, as a function of \( z_t \), the algorithm is assured that each agent \( i \) will eventually receive at least a promised utility of

\[
\tilde{u}_{i,t}(z_t) = \Delta_i + \sum_{\tau=1}^{t} v_{i,\tau} \cdot z_{\tau}.
\]
3 WARM UP: ONLINE ALLOCATION OF PUBLIC GOODS WITH BINARY UTILITIES

Before presenting our main results, let us first build some intuition about our online setting and the proportional fairness objective by considering a very restricted scenario wherein agents have binary utilities for goods (i.e., \(v_{i,t} \in \{0, 1\}\) for each \(i, t\)) and the total budget is \(B = 1\). We say that agent \(i\) “likes” good \(t\) if \(v_{i,t} = 1\), and does not like good \(t\) otherwise. Note that with \(B = 1\), the budget constraint is \(\sum_{t=1}^{T} x_t \leq 1\), which means \(x_t\) can be interpreted as the fraction of an available resource (e.g., time or money) that is dedicated to good \(t\).

First, in the trivial case with a single agent \((N = 1)\), we can simply set \(x_t = 1\) when the first good \(t\) liked by the agent arrives,\(^3\) which easily yields (exact) proportional fairness.

It is tempting to extend this idea to the case of \(N > 1\) agents. However, we find that even in this restricted scenario with binary utilities and unit budget, no online algorithm achieves \(o(\log N)\)-proportional fairness, or even the weaker guarantee of \(o(\log N)\)-approximation of the Nash welfare. In fact, this remains true even if the algorithm is horizon-aware (i.e., knows \(T\) in advance) and knows precisely how many goods each agent will like in total. Intuitively, this is because we show that no online algorithm can sufficiently distinguish between instances where many agents like the same goods and those where agents like mostly disjoint goods.

**Theorem 3.1.** With binary agent utilities \((v_{i,t} \in \{0, 1\}, \forall i, t)\) and unit budget \((B = 1)\), every online algorithm is \(\Omega(\log N)\)-proportionally fair (in fact, achieves \(\Omega(\log N)\)-approximation of the Nash welfare), even if the algorithm is horizon-aware and knows in advance the total number of goods each agent will like.

Before we prove the theorem, we need the following technical lemma. Recall that for \(k \in \mathbb{N}\), \(H_k\) is the \(k\)-th harmonic number.

**Lemma 3.2.** For every \(S \in \mathbb{N}\), \(W \geq 1\), and \(y = (y_t)_{t \in [S]} \in [0, 1]^{[S]}\) with \(\sum_{t=1}^{S} y_t \leq W\), we have

\[
\max_{t \in [S]} \frac{\ell + 1}{W + \sum_{j=1}^{t} j \cdot y_j} \leq \frac{H_{S}}{2W}.
\]

**Proof.** Suppose for contradiction that there exist \(S \in \mathbb{N}\), \(W \geq 1\), and \(y = (y_t)_{t \in [S]} \in [0, 1]^{[S]}\) such that \(\sum_{t=1}^{S} y_t \leq W\) and, for every \(t \in [S]\),

\[
\frac{\ell + 1}{W + \sum_{j=1}^{t} j \cdot y_j} < \frac{H_{S}}{2W} \Rightarrow \sum_{j=1}^{t} j \cdot y_j > \frac{2W \cdot (\ell + 1)}{H_{S}} - W.
\]

Dividing the above equation by \(\ell \cdot (\ell + 1)\) and summing over \(t \in [S]\), we have that

\[
\sum_{t=1}^{S} \sum_{j=1}^{t} \frac{j \cdot y_j}{\ell \cdot (\ell + 1)} > \sum_{t=1}^{S} \left( \frac{2W}{H_{S} \cdot \ell} - \frac{W}{\ell \cdot (\ell + 1)} \right).
\]

Let us analyze the LHS in Equation (1) by exchanging the order of summations. We have

\[
\text{LHS} = \sum_{j=1}^{S} j \cdot y_j \cdot \left( \sum_{t=1}^{S} \frac{1}{\ell \cdot (\ell + 1)} \right) = \sum_{j=1}^{S} j \cdot y_j \cdot \left( \sum_{t=1}^{S} \frac{1}{t} - \frac{1}{\ell + 1} \right)
\]

\[
= \sum_{j=1}^{S} j \cdot y_j \cdot \left( \frac{1}{j} - \frac{1}{S + 1} \right) \leq \sum_{j=1}^{S} y_j \leq W,
\]

where the third transition holds due to the telescopic sum.

\(^3\) If the agent does not like any good, (exact) proportional fairness is trivially achieved.
Now, let us analyze the RHS in Equation (1) using the same telescopic sum.

\[
\text{RHS} = \frac{2W}{HS} \sum_{t=1}^{S} \frac{1}{t} - W \cdot \sum_{t=1}^{S} \left( \frac{1}{t} - \frac{1}{t+1} \right) = \frac{2W}{HS} \cdot HS - W \cdot \left( 1 - \frac{1}{S+1} \right) \geq W.
\]

Hence, we proved that in Equation (1), the LHS is at most \( W \) and the RHS is at least \( W \). But the equation proves the LHS to be strictly greater than the RHS, which is the desired contradiction. □

Now, we are ready to prove Theorem 3.1.

**Proof of Theorem 3.1.** We prove that the Nash social welfare approximation of every online algorithm is at least \( H_N / 2 = \Omega(\log N) \). Suppose for contradiction that there is an online algorithm with a smaller approximation ratio.

Set \( T = N \cdot (N^2 + N - 2) / 2 \). We use \( \mathbf{v}_t = (v_{i,t})_{i \in [N]} \) to denote the vector of values of all the agents for good \( t \in [T] \). For \( r \in [N] \cup \{0\} \), we denote with \( S_r \) a sequence of \( N \) goods that arrive consecutively for which the agents have the following valuations. For the first good in the sequence, \( v_{i,1} = 1 \) for \( i \leq r \) and \( v_{i,1} = 0 \) for \( i > r + 1 \). For \( t \in \{2, \ldots, N\} \), \( \mathbf{v}_t \) is obtained by cyclically permuting \( \mathbf{v}_{t-1} \) to the right. For example, if \( N = 3 \), then \( S_2 \) consists of 3 goods that arrive consecutively with \( v_1 = (1, 1, 0), v_2 = (0, 1, 1), \) and \( v_3 = (1, 0, 1) \). Notice that \( S_r \) consists of \( N \) goods such that each good is liked by \( r \) agents and each agent likes \( r \) goods. Also, notice that \( S_0 \) consists of \( N \) goods for which all agents have zero value.

Next, we use \( S_r \) to construct two building blocks of our adversarial instances.
• For $r \in [N - 1]$, let $L_r$ be a sequence of consecutive goods constructed by concatenating $S_{r+1}$ followed by $r$ copies of $S_0$.

• For $r \in [N - 1]$, let $L'_r$ be a sequence of consecutive goods constructed by concatenating $r + 1$ copies of $S_1$.

Notice that for each $r \in [N - 1]$, $L_r$ and $L'_r$ both consist of $(r + 1) \cdot N$ goods of which each agent likes exactly $r + 1$ goods.

Finally, for each $k \in [N - 1]$, define instance $I_k = (L_1, L_2, \ldots, L_k, L'_{k+1}, \ldots, L'_{N-1})$, i.e., instance $I_k$ is constructed by concatenating $L_1, L_2, \ldots, L_k, L'_{k+1}, \ldots, L'_{N-1}$ in that order. Figure 1 shows instances $I_1$ and $I_2$ with $N = 4$. Notice that for each $k \in [N - 1]$, instance $I_k$ consists of a total of $T$ goods of which each agent likes $(N^2 + N - 2)/2$ goods. We let the algorithm know in advance that it will see instance $I_k$ for some $k \in [N - 1]$, and prove that it still cannot achieve proportional fairness better than $H_N/2$.

With slight abuse of notation, for $r \in [N - 1]$, let $L_r$ also denote the set of goods that appear in the sequence $L_r$. Let $x$ denote the allocation produced by the online algorithm on instance $I_{N-1}$, i.e., when the algorithm observes $L_1, L_2, \ldots, L_{N-1}$. For $k \in [N - 1]$, let $x_{L_k} = \sum_{t \in L_k} x_t$ denote the total allocation to goods in $L_k$.

Now, fix $k \in [N - 1]$ and let $x^k$ denote the allocation produced by the algorithm on instance $I_k$ with values $u_{i,t}$. Because the algorithm cannot distinguish between $I_k$ and $I_{N-1}$ for the first $k$ blocks (namely, $L_1, \ldots, L_k$), even under instance $I_k$, the algorithm must assign a total allocation of $x_{L_\ell}$ to block $L_\ell$ for each $\ell \leq k$. Hence, we have

$$
NSW(x^k) = \left( \prod_{i=1}^{N} \left( \sum_{t=1}^{k} \sum_{i \in L_\ell} u_{i,t} \cdot x^k_t + \sum_{t=k+1}^{N-1} \sum_{i \in L'_t} u_{i,t} \cdot x^k_t \right) \right)^{1/N}
$$

$$
\leq \frac{1}{N} \cdot \left( \sum_{i=1}^{N} \left( \sum_{t=1}^{k} \sum_{i \in L_\ell} u_{i,t} \cdot x^k_t + \sum_{t=k+1}^{N-1} \sum_{i \in L'_t} u_{i,t} \cdot x^k_t \right) \right)
$$

$$
= \frac{1}{N} \cdot \left( \sum_{t=1}^{k} \sum_{i \in L_\ell} N \cdot u_{i,t} \cdot x^k_t + \sum_{t=k+1}^{N-1} \sum_{i \in L'_t} N \cdot u_{i,t} \cdot x^k_t \right)
$$

$$
\leq \frac{1}{N} \cdot \left( \sum_{t=1}^{k} (\ell + 1) \cdot x^k_t + \sum_{t=k+1}^{N-1} x^k_t \right)
$$

$$
\leq \frac{1}{N} \cdot \left( \sum_{t=1}^{k} (\ell + 1) \cdot x_{L_\ell} + 1 - \sum_{t=1}^{k} x_{L_\ell} \right)
$$

where the second transition follows from the AM-GM inequality and the fourth transition follows because, for each $\ell \in [N - 1]$, each good $t \in L_\ell$ is liked by at least $\ell + 1$ agents and each good $t \in L'_\ell$ is liked by a single agent.

Consider an alternative allocation $y^k$ which allocates $i/N$ to each of the first $N$ goods of $L_k$ (i.e., goods of its $S_{k+1}$ portion). This provides each agent utility equal to $(k + 1)/N$, thus achieving Nash social welfare equal to $(k + 1)/N$.

Hence, the algorithm’s approximation ratio $\alpha_k$ on instance $I_k$ satisfies

$$
\alpha_k \geq \frac{k + 1}{\sum_{t=1}^{k} (\ell + 1) \cdot x_{L_\ell} + 1 - \sum_{t=1}^{k} x_{L_\ell}} \geq \frac{k + 1}{1 + \sum_{t=1}^{k} (\ell + 1) \cdot x_{L_\ell}}.
$$
ALGORITHM 1: Set-Aside Greedy Algorithm for Binary Values and Unit Budget

Input: Target proportional-fairness level $\alpha$

1: for all $t = 1$ to $T$ do
2: (Set-aside semi-allocation) If there exists $i \in [N]$ with $v_{i,t} = 1$ and $v_{i,r} = 0$ for each $r \in [t-1]$, then set $y_t = 1/(2N)$, else set $y_t = 0$.
3: (Greedy semi-allocation) Compute $z_t = \min \left\{ z_t : \frac{1}{N} \sum_{t=1}^{N} \frac{v_{i,t}}{u_{i,t}(z_t)} \leq \alpha, z_t \geq 0 \right\}$.
4: Allocate $x_t = y_t + z_t$ to good $t$.
5: end for

Hence, the worst-case approximation ratio is at least $\max_{k \in [N-1]} \alpha_k$. Applying Lemma 3.2 with $S = N$ and $W = 1$, we get that this is at least $H_N/2$, as desired. $\square$

Next, we provide a set-aside greedy algorithm that achieves $O(\log N)$ proportional fairness (and therefore, $O(\log N)$-NSW optimality), thus establishing $\Theta(\log N)$ as the best possible approximation in this restricted scenario. We remark that this restricted case of binary utilities and unit budget already poses an interesting challenge by preventing constant approximation, but $O(\log N)$ approximation is still quite reasonable as it does not depend on the horizon $T$ (which can typically be very large) and in practice the number of agents $N$ is reasonably small. We also remark that we achieve the $O(\log N)$ upper bound using a horizon-independent algorithm, while the lower bound of Theorem 3.1 holds even when the algorithm is horizon-aware.

The algorithm, provided as Algorithm 1, works as follows. The algorithm uses the set-aside portion of the budget to set $y_t = 1/(2N)$ whenever good $t$ is the first liked good of at least one agent (note that $\sum_t y_t \leq 1/2$). This ensures that each agent $i$ gets utility at least $\Delta_i = 1/(2N)$.\footnote{We can set $y_t = 0$ and $y_t = x_{L,t-1}$ for all $t \in \{2, \ldots, N\}$ when applying the lemma.} Based on this, the algorithm uses the following expression of promised utility to agent $i$ in round $t$:

$$
\bar{u}_{i,t}(z_t) = \frac{1}{2N} + \sum_{r=1}^{t} v_{i,r} \cdot z_r. \tag{2}
$$

The algorithm chooses $z_t$ in a greedy manner (i.e., smallest possible) such that, for each agent $i$, the ratio of her value $v_{i,t}$ for good $t$ to her promised utility $\bar{u}_{i,t}(z_t)$ is at most a target quantity. We defer the proof of this result to the appendix because we will present this technique in much more generality in Section 4.2 (the only adjustment required in the proof of the next result is the slightly different expression of $\Delta_i = 1/(2N)$ specific to this case of binary utilities and unit budget).

THEOREM 3.3. Algorithm 1 with $\alpha \geq 2 \ln(2N)$ realizes an $\alpha$-proportional fair allocation.

4 THE GENERAL SETTING

Having built some intuition about the online setting and the proportional fairness objective by considering the restricted special case of the problem wherein all values $v_{i,t}$ are in $\{0, 1\}$ and the budget is $B = 1$, we now turn to the more general model described in Section 2. Recall that this model generalizes the setting in Section 3 in two ways:

(1) Agent values $v_{i,t}$ can now be any (non-negative) real number.
(2) The budget constraint is $\sum_{t=1}^{T} x_t \leq B$, for an arbitrary $B \geq 0$, so the per-round constraint of $x_t \leq 1$, for each $t \in [T]$, is no longer redundant.
4.1 The Case for Predictions

In this general case, we first prove that the problem becomes significantly more difficult: every online algorithm is \( \Omega(T/B) \)-proportionally fair (in fact, achieves \( \Omega(T/B) \)-approximation of the Nash welfare). This is in stark contrast to the \( O(\log N) \)-proportional fairness we were able to achieve in the previous section.

**Proposition 4.1.** Under general agent values and budget \( B \), every online algorithm is \( \Omega(T/B) \)-proportionally fair (in fact, achieves \( \Omega(T/B) \)-approximation of the Nash welfare).

**Proof.** The hardness instances we consider will have \( N = 1 \) agent. In this case, the proportional fairness of an online algorithm on a given instance is equal to the maximum possible utility the agent could have obtained in hindsight divided by the utility she obtained under the algorithm; this is also equal to the approximation ratio for the Nash welfare. For clarity, we will omit the subscript \( i \) in the notation in this proof, so that \( v_t \) is the value that the agent has for good \( t \), and \( u(x) \) is the utility of the agent under allocation \( x \). With \( N = 1 \), a problem instance is defined by a sequence of \( T \) values \( (v_1, v_2, \ldots, v_T) \).

Consider the following family of \( T \) instances: for \( t \in [T] \), instance \( I_t := (1, M, \ldots, M^{t-1}, 0, \ldots, 0) \), where \( M \) is a sufficiently large number. The algorithm knows in advance that it will see instance \( I_t \) for some \( t \in [T] \), and prove that it still cannot achieve proportional fairness better than \( T/2B \).

Let \( x = (x_1, \ldots, x_T) \) be the allocation produced by the algorithm on instance \( I_T = (1, M, \ldots, M^{T-1}) \). For any \( t \in [T] \), since the algorithm cannot distinguish between \( I_t \) and \( I_T \) up to round \( t \), the allocation up to round \( t \) under instance \( I_t \) must also be \( (x_1, \ldots, x_t) \). Further, without loss of generality, we can assume that the algorithm allocates 0 in any round where the value is 0. Therefore, for each \( t \in [T] \), the allocation produced by the algorithm on instance \( I_t \) is \( (x_1, \ldots, x_t, 0, \ldots, 0) \).

We claim that in order to be \( \frac{T}{2B} \)-proportionally fair, the algorithm needs to set \( x_t \geq \frac{B}{T} \) for all \( t \). To see this, suppose to the contrary that \( x_t < \frac{B}{T} \) for some \( t \). Then on instance \( I_t \), the agent’s utility under the algorithm is

\[
    u(x) = \sum_{r=1}^{t} x_r M^r < \sum_{r=1}^{t-1} M^r + \frac{B}{T} \cdot M^t \leq \frac{2B}{T} M^t,
\]

where the last inequality holds when \( M \) is large enough. On the other hand, the hindsight-optimal allocation on \( I_t \) is to simply set \( x^*_t = 1 \), which gets utility \( M^t \). Therefore, if \( x_t < \frac{B}{T} \) then the algorithm cannot be \( \frac{T}{2B} \)-proportionally fair on \( I_t \).

We have shown that \( x_t \geq \frac{B}{T} \) for all \( t \) is necessary for the algorithm to be \( \frac{T}{2B} \)-proportionally fair. However, since the overall budget is \( B \), the only way this can happen is if \( x_t = \frac{B}{T} \) for all \( t \). The same calculation as above shows that under this allocation, the algorithm is at most \( (T/2B) \)-proportionally fair, establishing the desired \( \Omega(T/B) \) lower bound.

The hardness instance above is specifically engineered to exploit the fact that the algorithm has no information about the future. In most practical settings, it is reasonable to assume that the algorithm has access to some information about the input. This could come from historical data, stochastic assumptions, or simply from properties of the problem at hand (e.g. if \( v_{i,t} \) represents the monetary value that agent \( i \) has for good \( t \), then we may have bounds on how large \( v_{i,t} \) can be.)

Motivated by this, we now turn to the growing literature on prediction-augmented algorithms and allow the algorithm access to additional side-information about agents’ valuations. Clearly, if the entire valuation matrix \( (v_{i,t})_{i \in [N], t \in [T]} \) is available beforehand, then the problem is trivial; the challenge lies in understanding what minimal additional information (or ‘prediction’) can lead to sharp improvements in performance, and how robust these improvements are to errors in these...
predictions. To this end, we now adapt an idea introduced by Banerjee et al. [2022] for online allocation with private goods, and assume that the algorithm has side information about each agent’s total value for all the goods.

**Definition 4.2 (Total Value Predictions).** For any agent $i$, we define her total value to be $V_i = \sum_{t=1}^{T} u_{i,t}$. Moreover, for $c_i, d_i \geq 1$, $\bar{V}_i$ is said to be a $[c_i, d_i]$-prediction of $V_i$ if $\bar{V}_i \in \left[ \frac{1}{d_i} V_i, c_i V_i \right]$.

In other words, $c_i$ and $d_i$ denote the multiplicative factors by which the prediction $\bar{V}_i$ may overestimate and underestimate, respectively, the value of $V_i$. When $c_i = d_i = 1$, we say we have perfect predictions.

In the next section, we assume that we have access to $\bar{V}_i$ for each agent $i$. The purpose of the $c_i$ and $d_i$ is to parameterize the robustness of our algorithm, i.e., the degradation in its performance as the predictions get worse. We note that our algorithm below does need to know (an upper bound on) the $d_i$’s for tuning one of its parameters; it does not however need to know the $c_i$’s (these are only used in the analysis).

### 4.2 Achieving Proportional Fairness with Predictions

Using the above notion of predictions, we now present, as Algorithm 2, a variant of our earlier Set-Aside Greedy algorithm that has a dramatically better proportional fairness guarantee compared to the hardness result of $\Omega(T/B)$ in Proposition 4.1. Given perfect predictions, our algorithm achieves a proportional fairness of $O(\log(T/B))$. Moreover, Algorithm 2 turns out to be remarkably robust to prediction errors; in particular, all our asymptotic guarantees remain unchanged as long as all the $c_i = O(1)$ and $d_i = O(\log(T/B))$.

As before, the algorithm splits the budget into two parts, and the total allocation to the good in round $t$ is obtained by adding the contributions (semi-allocations) from each part, i.e., $x_t = y_t + z_t$. The semi-allocation from the first (set-aside) part is $y_t = B/(2T)$ for each $t \in [T]$. This portion of the allocation guarantees each agent $i$ utility at least $\Delta_i = \frac{B}{2T} \cdot V_i$. Now in each round $t \in [T]$, the algorithm uses the second the part of the budget to compute a greedy semi-allocation $z_t$. This is done by choosing $z_t$ to optimize a function of the agents’ predicted promised utilities.

**Definition 4.3 (Predicted Promised Utility).** Given a prediction $\bar{V}_i$ of the total value of agent $i$, the predicted promised utility of agent $i$ in round $t \in [T]$ is defined as

$$\bar{u}_{i,t}(z_t) = \frac{B}{2T} \cdot \bar{V}_i + \sum_{r=1}^{t} u_{i,r} \cdot z_r.$$  \quad (3)

We omit $z_1, \ldots, z_{t-1}$ from the argument of $\bar{u}_{i,t}$ since they are fixed prior to round $t$.

Note that this quantity can be computed by the algorithm, as a function of $z_t$ it wants to choose, since it has knowledge of $\bar{V}_i$ (prediction) and semi-allocations $\{z_r\}_{r \leq t}$ from the previous rounds. We use these predicted promised utilities in Algorithm 2 to achieve the following guarantee.

**Theorem 4.4.** For any $\alpha \geq 4 \ln \left( \frac{2T}{B} \right) + 4 \frac{N}{N} \sum_{i} \ln(d_i)$, Algorithm 2 produces a feasible allocation $x$, which satisfies

$$\max_{w} \frac{1}{N} \sum_{i=1}^{N} \frac{u_i(w)}{c_i \cdot u_i(x)} \leq \alpha,$$

where the maximum is taken over all feasible allocations $w$.

The proof of this result is somewhat long, and so we defer it to its own section (Section 4.4). First, however, we consider some implications of this result. In particular, note that the expression in the
Algorithm 2: Set-Aside Greedy algorithm for the general setting

Input: Target threshold \( \alpha \); total value predictions \((\hat{V}_i)_{i \in [N]}\)

1: for all \( t = 1 \) to \( T \) do
2: Set-aside semi-allocation: set \( y_t = \frac{B}{2T} \).
3: Greedy semi-allocation: \( z_t = \min\{z_t^*, 1 - y_t\} \), where \( z_t^* = \min \left\{ z \geq 0 : \frac{1}{N} \sum_{i=1}^{N} \frac{u_{i,t}(z)}{c_i(z)} \leq \frac{\alpha}{2B} \right\} \).
4: Allocate \( x_t = y_t + z_t \).
5: end for

The statement of Theorem 4.4 is not exactly the proportional fairness objective, since the term for each agent \( i \) is scaled with a (potentially different) factor \( c_i \). Applying the arguments in Section 2.1, we can turn this into an approximation guarantee on proportional fairness, the core, and the Nash social welfare.

**Corollary 4.5.** For \( \alpha \geq 4 \ln \left( \frac{2T}{B} \right) + \frac{4}{N} \sum_i \ln(d_i) \), Algorithm 2 is

1. \((\alpha \cdot \max_{i \in [N]} c_i)\)-proportionally fair, and hence in the \((\alpha \cdot \max_{i \in [N]} c_i)\)-core, and
2. achieves \((\alpha \cdot (\prod_{i \in [N]} c_i)^{\frac{1}{N}})\)-approximation of the Nash welfare.

**Proof.** (1) Theorem 4.4 implies that

\[
\frac{1}{\max_i c_i} \cdot \max_w \frac{1}{N} \sum_{i=1}^{N} \frac{u_i(w)}{u_i(x)} \leq \max_w \frac{1}{N} \sum_{i=1}^{N} \frac{u_i(w)}{c_i u_i(x)} \leq \alpha,
\]

which directly implies the desired proportional fairness guarantee and, by Proposition 2.2, the desired core guarantee. While the same guarantee carries over to Nash welfare approximation, repeating the proof of Proposition 2.3 actually provides a better approximation.

(2) Let \( x^* \) be the allocation maximizing the Nash social welfare in hindsight. We have

\[
\max_w \frac{1}{N} \sum_{i=1}^{N} \frac{u_i(w)}{c_i u_i(x)} \geq \frac{1}{N} \sum_{i=1}^{N} u_i(x^*) \overset{(a)}{=} \left( \prod_{i=1}^{N} \frac{u_i(x^*)}{c_i u_i(x)} \right)^{\frac{1}{N}} \overset{(b)}{=} \left( \prod_{i=1}^{N} \frac{1}{c_i} \right)^{\frac{1}{N}} \frac{\text{NSW}(x^*)}{\text{NSW}(x)}
\]

where (a) is by the AM-GM inequality, and (b) is by the definition of the Nash welfare. This, together with Theorem 4.4, yields the second part of the corollary.

### 4.3 Hardness with Predictions

Theorem 4.4 shows that in online allocation of public goods with general values and budget, having access to reasonable predictions of each agent’s total value can lead to a dramatic improvement in the proportional fairness guarantee from \( \Omega(T/B) \) to \( O(\log(T/B)) \). Given the size of the side information (which lies in \( \mathbb{R}^N \), since we need one prediction per agent) relative to the ambient size of the input (which lies in \( \mathbb{R}^{NT} \), with one valuation per agent per round), this is a surprising improvement in performance.

One may wonder whether these predictions are so strong that one can do even better. The following result shows, however, that even with a single agent, and perfect knowledge of her total value \( V_1 \), any online algorithm is \( \Omega(\log(T/B))\)-proportionally fair. Thus Algorithm 2 is essentially optimal for our setting.

**Theorem 4.6.** For \( N = 1 \) agent, every online algorithm is \( \Omega(\log(T/B))\)-proportionally fair (in fact, achieves an \( \Omega(\log(T/B))\)-approximation for the Nash welfare), even with perfect knowledge of horizon \( T \) and the total value of the agent \( \hat{V}_1 = V_1 = \sum_{t=1}^{T} v_{1,t} \).
PROOF. As in the proof of Proposition 4.1, approximations to proportional fairness and Nash welfare are equivalent with \( N = 1 \) agent, both coinciding with the ratio of the agent’s maximum utility in hindsight to the agent’s utility under the algorithm.

For the sake of contradiction, suppose that that there exists an online algorithm whose approximation ratio is \( o(\log(T/B)) \). Let \( T = B \cdot (T'(T' + 1) - 2)/2 \) for some \( T' \). For \( r \in [T' - 1] \), we denote with \( S_r \) the sequence of \( B \cdot (r + 1) \) goods that arrive consecutively for which the agent has utility equal to \((r + 1)/T'\) for the first \( B \) goods and utility equal to 0 for the remaining goods and with \( S'_r \) the sequence of \( B \cdot (r + 1) \) goods such that the agent has utility equal to \( 1/T' \) for each of the goods. Now, for each \( k \in [T' - 1] \), let \( I_k = (S_1, S_2, \ldots, S_k, S'_k, \ldots, S'_{T-1}) \), i.e. \( I_k \) be the instance that is constructed by concatenating \( S_1, S_2, \ldots, S_k, S'_k, \ldots, S'_{T-1} \) in that order. Notice that for \( r \in [T' - 1] \), \( S_r \) and \( S'_r \) have the same number of goods, equal to \( B(r + 1) \), and for each of them the agent has the same accumulated utility, equal to \( B(r + 1)/T' \). Hence, each instance \( I_k \) consists of \( T \) goods and the agent has the same accumulated utility for all these instances. We assume that the algorithms is aware that it will see instance \( I_k \) for some \( k \in [T' - 1] \).

With slight abuse of notation, for \( r \in [T' - 1] \), let \( S_r \) to also denote the set of goods that appear in the sequence \( S_r \). We denote with \( x \) the allocation produced by the online algorithm on instance \( I_{T-1} \). For \( r \in [T' - 1] \), let \( x_{S_r} = \sum_{t \in S_r} x_t \) denote the total allocation to goods in \( S_r \). Now, for \( k \in [T' - 1] \), let \( x^k \) denote the allocation produced by the algorithm on instance \( I_k \). As the algorithm cannot distinguish between \( I_k \) and \( I_{T-1} \) for the first \( k \) blocks (i.e. \( S_1, \ldots, S_k \)), under instance \( I_k \), the algorithm must assign a total allocation of \( x_{S_k} \) to block \( S_k \) for each \( \ell \leq k \).

Moreover, under instance \( I_k \), the optimal algorithm allocates 1 to each of the first \( B \) goods of \( S_k \). Hence, we have that under instance \( I_k \), if \( \alpha_k \) is the approximation ratio of the algorithm, then
\[
\alpha_k \geq \frac{B \cdot (k + 1)/N}{(2 \cdot x_{S_1} + 3 \cdot x_{S_2} + \ldots + (k + 1)x_{S_k} + (1 - x_{S_1} - x_{S_2} - \ldots - x_{S_k})) / N}.
\]
\[
= \frac{B \cdot x_{S_1} + 2 \cdot x_{S_2} + \ldots + kx_{S_k} + 1}{(k + 1)}.
\]

Hence, if \( \alpha \) is the worst-case approximation ratio over all the instances \( I_k \), we have
\[
\alpha \geq B \cdot \max_{k \in [T' - 1]} \frac{(k + 1)}{x_{S_1} + 2 \cdot x_{S_2} + \ldots + kx_{S_k} + 1}
\]
and by applying Lemma 3.2, with \( S = T' \) and \( W = B \), we get that \( \alpha \geq B \cdot H_{T'}/(2B) \). Thus, we have that \( \alpha = \Omega(\log(T')) = \Omega(\log(T/B)) \).

\[\square\]

4.4 Proof of Theorem 4.4

Finally, we return to the proof of our main performance guarantee in Theorem 4.4. Henceforth, in this section, we let \( x \) denote the final allocation produced by Algorithm 2.

Central to our analysis is the following linear program (and its dual), which essentially tries to maximize (i.e., find the worst case) the proportional-fairness level achieved by Algorithm 2:

\[
(P) \quad \max_{w \in \mathbb{R}^T} \frac{1}{N} \sum_{i=1}^{N} \frac{u_i(w)}{c_i(x)} \quad \text{subject to} \quad \sum_{t=1}^{T} w_t \leq B
\]

\[
(D) \quad \min_{p \in \mathbb{R}^T, q \in \mathbb{R}} \sum_{t=1}^{T} p_t + Bq \quad \text{subject to} \quad p_t + q \geq \frac{1}{N} \sum_{i=1}^{N} \frac{u_i(w)}{c_i(x)} \quad \forall t \in [T]
\]

\[0 \leq w_t \leq 1 \quad \forall t \in [T] \quad p_t, q \geq 0
\]

Algorithm 2 can now be viewed as choosing \( x \) in a manner such that it forces the LP \((P)\) to have as small a value as possible. One way to achieve this is to consider a ‘target’ dual solution \( q = \frac{1}{N} \alpha \)
and \( p_t = \max \{ 0, \frac{1}{N} \sum_{i=1}^{N} \frac{u_{i,t}}{u_{i,t}(z)} - q \} \) (for some appropriate choice of \( \alpha \)). To realize this, Algorithm 2 wants to set the vector of allocations \( x_t \) for each round \( t \), as well as the corresponding dual variable \( p_t \), in a manner such that the primal and dual solutions are consistent (i.e., all constraints are feasible, and they obey complementary slackness).

The challenge in generating allocation \( x \) and dual certificate \( (p_t, q) \) in an online fashion is that at time \( t \), the dual constraint depends on \( u_i(x) \), the final utilities under the entire allocation made by the algorithm. However, these quantities are not known to the algorithm at time \( t \), because it cannot look into the future. To work around this, we use the predicted utilities (Definition 4.3) as proxy in place of \( u_i(x) \) when making the decision in round \( t \). The next lemma explains the sense in which the predicted utilities serve as a proxy; in particular, we show that under Algorithm 2, the predicted utility \( \tilde{u}_{i,t}(z_t) \) for each agent \( i \) and in each round \( t \), is a lower bound on the true final utility of agent \( i \) under \( x \) up to a multiplicative factor of \( c_i \).

**Lemma 4.7.** For every agent \( i \in [N] \) and time \( t \in [T] \), we have \( \tilde{u}_{i,t}(z_t) \leq c_i u_i(x) \).

**Proof.** We have
\[
u_i(x) = \sum_{r=1}^{T} u_{i,r}(y_r + z_r) = \sum_{r=1}^{T} u_{i,r}y_r + \sum_{r=1}^{T} u_{i,r}z_r,
\]
We next bound the contribution due to the set-aside allocations \( y \). At any time \( t \in [T] \), agent \( i \) receives at least \( v_{i,t} y_t = v_{i,t} \frac{B}{2T} \) utility from the set-aside semi-allocation \( y_t \). Hence,
\[
\sum_{r=1}^{T} u_{i,r}y_r \geq \sum_{r=1}^{T} v_{i,r}y_r \geq \frac{B}{2T} \sum_{r=1}^{T} v_{i,r}y_r \geq \frac{1}{c_i} \left( \frac{B}{2T} \tilde{V}_t + \sum_{r=1}^{T} v_{i,r}z_r \right) = \frac{1}{c_i} \tilde{u}_{i,t}(z_t).
\]
Using this, we now formally define our new dual certificates as follows:

**Dual Certificates.** Given allocation \( x = y + z \) from Algorithm 2, we define
\[
q = \frac{\alpha}{2B} \geq \frac{2}{B} \left[ \ln \left( \frac{2T}{B} \right) + \frac{1}{N} \sum_{i} \ln(d_i) \right],
\]
\[
p_t = \max \left\{ 0, \frac{1}{N} \sum_{i=1}^{N} \frac{u_{i,t}}{u_{i,t}(z_t)} - q \right\}
\]
(4)

Before proving Theorem 4.4, we list some key properties of the allocation and certificate which we need for our performance guarantee.

**Proposition 4.8 (Some Properties of the Algorithm).** The allocation \( x = y + z \) returned by Algorithm 2, and corresponding dual certificates \( p, q \) in Eq. (4) together satisfy the following:

1. \( z_t < 1 - \frac{B}{2T} \implies p_t = 0 \).
2. \( z_t > 0 \implies p_t + q = \sum_{i=1}^{N} \frac{u_{i,t}}{u_{i,t}(z_t)} \).

**Proof.** (1) follows because if \( z_t < 1 - \frac{B}{2T} \), then that must mean \( z_t = z^*_t \). Hence, \( \frac{1}{N} \sum_{i=1}^{N} \frac{u_{i,t}}{u_{i,t}(z_t)} \leq \frac{\alpha}{2B} = q \), which implies \( p_t = 0 \).
(2) follows because if \( z_t > 0 \), then that must mean \( z_t^* > 0 \). Let \( \Phi(z) = \frac{1}{N} \sum_{i=1}^{N} \frac{u_i(z)}{u_i(z_t)} \). Since \( \Phi \) is a continuous function in \( z \), this implies \( \Phi(z_t^*) = \frac{\alpha}{2B} = q \). Because \( z_t = \min\{z_t^*, 1 - y_t\} \leq z_t^* \) and \( \Phi \) is decreasing, we have \( \Phi(z_t) \geq \Phi(z_t^*) = q \). Thus \( p_t = \max\{0, \Phi(z_t) - q\} = \Phi(z_t) - q \). \( \square \)

We can now prove our main performance guarantee for Algorithm 2, which we restate below.

**Theorem 4.4.** For any \( \alpha \geq 4 \ln \left( \frac{2T}{B} \right) + \frac{4}{N} \sum_i \ln(d_i) \), Algorithm 2 produces a feasible allocation \( x \), which satisfies

\[
\max_w \frac{1}{N} \sum_{i=1}^{N} \frac{u_i(w)}{c_i \cdot u_i(x)} \leq \alpha,
\]

where the maximum is taken over all feasible allocations \( w \).

**Proof.** To prove the theorem, it suffices to check the following 3 statements:

1. Algorithm 2 returns a feasible allocation \( x \).
2. The dual certificate \((p_t, q)\) in Proposition 4.8 is feasible to the dual LP \((D)\).
3. \( \sum_t p_t + Bq \leq \alpha \).

Given these properties, the guarantee follows via weak LP duality for programs \((P)\) and \((D)\).

For Claim (2), note that by Lemma 4.7, predicted utilities are a lower bound on \( c_i u_i(x) \) for all \( i \in [N], t \in [T] \). Now by definition of our dual certificate, we have for all \( i \in [N], t \in [T] \)

\[
 p_t + q \geq \sum_{i=1}^{N} \frac{u_i}{u_i(z_t)} \geq \sum_{i=1}^{N} \frac{u_i}{c_i u_i(x)}
\]

Claims (1) and (3) are implied by the following key invariant:

**Key Invariant.** For any \( \alpha \geq 4 \ln \left( \frac{2T}{B} \right) + \frac{4}{N} \sum_i \ln(d_i) \), the corresponding \( x, p, q \)

\[
\sum_t (p_t + q)z_t \leq \frac{\alpha}{4}
\]  

(5)

We first show how the invariant in Eq. (5) implies Claims (2) and (3).

1. Clearly \( x \) satisfies the per-round constraints by definition of the algorithm, so we just need to show that it satisfies the overall budget constraint. Since the set-aside semi-allocation is chosen such that \( \sum_t y_t = \frac{B}{2} \), we only need to check that \( \sum_t z_t \leq \frac{B}{2} \). Now, since \( p_t \geq 0 \ \forall \ t \), Eq. (5) implies

\[
\sum_t z_t \leq \frac{1}{q} \cdot \frac{\alpha}{4} = \frac{B}{2},
\]

by the definition of \( q = \frac{\alpha}{2B} \).

3. By (the contrapositive of) item 1 of Proposition 4.8, we have that if \( p_t > 0 \) then \( z_t = 1 - \frac{B}{2T} \). Using this and Eq. (5), we have

\[
\sum_t p_t \left( 1 - \frac{B}{2T} \right) = \sum_t p_t z_t \leq \frac{\alpha}{4}.
\]

Moreover, since \( B \leq T \), this implies \( \sum_t p_t \leq \frac{\alpha}{4} \). Now, by our choice of \( q = \frac{\alpha}{2B} \), we have

\[
\sum_t p_t + Bq \leq \frac{\alpha}{2} + \frac{B\alpha}{2B} = \alpha.
\]
Finally we turn to the proof of the key invariant in Eq. (5). We have

\[ \sum_t (p_t + q)z_t = \sum_t \left( \frac{1}{N} \sum_i \frac{v_{i,t}}{\bar{u}_{i,t}(z_t)} \right) z_t \]  
(by item 2 of Proposition 4.8)

\[ = \frac{1}{N} \sum_t z_t \cdot \sum_i \frac{v_{i,t}}{\bar{u}_{i,t}(z_t)} \]

\[ = \frac{1}{N} \sum_t \sum_i \frac{z_t v_{i,t}}{\bar{u}_{i,t}(z_t)} = \frac{1}{N} \sum_t \sum_i \frac{\bar{u}_{i,t}(z_t) - \bar{u}_{i,t}(0)}{\bar{u}_{i,t}(z_t)} \]

Where the last equality follows from our definition of the promised utility (Definition 4.3). Now, using the fact that \( 1 - x \leq -\ln(x) \forall x \in \mathbb{R} \), we have

\[ \sum_t (p_t + q)z_t \leq \frac{1}{N} \sum_t \sum_i \left[ \ln \left( \frac{B}{2T} \bar{V}_i + \sum_t v_{i,t}z_t \right) \right] \ln \left( \frac{B}{2T} \bar{V}_i \right) \]

\[ = \frac{1}{N} \sum_i \left[ \ln \left( \frac{B}{2T} \bar{V}_i + \sum_t v_{i,t}z_t \right) \right] \]  
(by telescoping)

\[ = \ln \left( \frac{2T}{B} \right) + \frac{1}{N} \sum_i \ln \left( \frac{B}{2T} + \frac{1}{V_i} \sum_t v_{i,t}z_t \right) \]

Now, observe that

\[ \sum_t v_{i,t}z_t \leq \sum_t v_{i,t} \left( 1 - \frac{B}{2T} \right) = V_i \left( 1 - \frac{B}{2T} \right), \]

where the first inequality is because \( z_t \leq 1 - \frac{B}{2T} \), a condition which is enforced by the algorithm. Thus

\[ \frac{B}{2T} + \frac{1}{V_i} \sum_t v_{i,t}z_t \leq \frac{B}{2T} + \left( 1 - \frac{B}{2T} \right) \frac{V_i}{V_i} \leq \frac{B}{2T} + \left( 1 - \frac{B}{2T} \right) d_i \leq d_i. \]

Therefore, we conclude that

\[ \sum_t (p_t + q)z_t \leq \ln \left( \frac{2T}{B} \right) + \frac{1}{N} \sum_i \ln(d_i) \leq \frac{\alpha}{4}, \]

which concludes the proof of the key claim. \( \square \)

5 DISCUSSION
We conclude with a discussion of possible future directions.

The Geometry of Online Fair Allocation. Most existing work on online allocation has focused on utilitarian objectives, in particular, welfare and revenue. As a result of this literature, we have a deep understanding of how different user utility functions (additive, unit-demand, constant elasticity of substitution, etc.) and allocation constraints (knapsack, matroid, downward-closed, etc.) affect the design and guarantees of online algorithms. Critical to much of this work, however, is the assumption that rewards are additive across rounds and agents. In settings where decisions are coupled in non-linear ways, as is often the case for fairness-related objectives such as proportional fairness, our state of knowledge is much more nascent. To this end, our paper contributes to an ongoing stream of work in mapping out the geometry of online fair allocation; in particular, our
results provide sharp insights into the effect of public goods (i.e., positive externalities), as well as per-round and overall budget constraints on online fair allocation.

That being said, our work also raises many follow-up questions. Firstly, while we focus on the proportional fairness objective, a natural question is whether our results can be extended to other non-linear objective functions, such as the class of generalized $p$-mean welfare measures (see, for example, the work of Barman et al. [2022] on allocation of private goods). Similarly, while our results give insight into the interaction between per-round budget constraints and overall budget constraints, and one can ask if these ideas can be extended to more complex classes of constraints (e.g., general packing constraints considered by Fain et al. [2018]). Finally, while there is a deep understanding of the challenges of producing integral allocations in offline settings, much less is known about integral allocations in online settings.

Alternate Information Structures. Our results show that while no side-information is needed to get the tight $O(\log N)$ proportional fairness factor in the case of binary valuations, in the case of general valuations the availability of predictions (of agents’ total values) leads to a significant improvement in performance, from $\Theta(T/B)$ to $\Theta(\log(T/B))$. On the other hand, if the algorithm had access to all the $v_{i,t}$ values in advance, then we could directly compute a 1-proportionally fair solution by maximizing the Nash social welfare, which is a convex program. Similarly, our bounds also show that underestimation (i.e., high $d_i$) has a much more benign effect than overestimation (i.e., high $c_i$). While the proofs provide some intuition behind these results, they are still somewhat surprising, and hard to explain in a unified way.

To this end, an interesting direction for future research is to develop a better understanding regarding whether what types of predictions would be appropriate for online fair division. For example, what if the algorithm is also provided with a prediction regarding the total value of each good across all agents, but not specifying which of the agents will like it, and by how much? Could this additional information allow us to overcome the logarithmic lower bounds and maybe even achieve a constant approximation? Alternately, would it help to know more detailed patterns about agent valuations — for example, if they take discrete values, or are periodic, or have low variance? Developing a unified theory that captures these models is an interesting future direction.

Alternate Arrival Models. A related question to the one above is in regards to the process for generating agent valuations. In our model, the values $V = (v_{i,t})_{i \in [N], t \in [T]}$ are allowed to arrive in an adversarial order. However, one can argue that this is too pessimistic. Two other arrival models that have been studied in online algorithms are (1) the random-order model, in which $V$ is selected by an adversary but the arrival order of the rows of $V$ is assumed to be uniformly at random, and (2) the stochastic model, in which the values in each round are drawn independently from a known distribution. Can one obtain improved guarantees for proportional fairness under these models? It is even possible that predictions are not needed when studying these arrival models, since they inherently already give the algorithm some information about the input.

Incentives. Finally, one of the most important open questions is how to convert online algorithms to online mechanisms, whereby agents report their private $v_{i,t}$ values in each round. In such a setting, can we design mechanisms which are incentive compatible? There has been recent work on similar questions for static fair allocation [Amanatidis et al., 2017, Aziz et al., 2019b, Brânzei et al., 2017, Halpern et al., 2020], as well as online welfare maximization [Balseiro et al., 2019, Gorokh et al., 2017, 2021, Guo et al., 2009]; extending these ideas to online fair allocation is an important open problem.
REFERENCES


A PROOF OF THEOREM 3.3

Proof. Let $x$ denote Algorithm 1’s final allocation, and consider the following LP and its dual:

$\begin{align*}
(P) \quad & \max_{w \in \mathbb{R}^+} \frac{1}{N} \sum_{i=1}^{N} u_i(w) \\
& \text{s.t. } \sum_{i=1}^{T} w_i \leq 1
\end{align*}$

$\begin{align*}
(D) \quad & \min_{p \in \mathbb{R}_+} p \\
& \text{s.t. } p \geq \frac{1}{N} \sum_{i=1}^{N} \frac{v_{it}}{u_i(x)} \quad \forall t \in [T]
\end{align*}$

By construction, $\gamma$ is feasible to the dual LP, because

$\gamma \geq \frac{1}{N} \sum_{i=1}^{N} \frac{v_{it}}{u_i(x)} \geq \max_{i} \frac{1}{N} \sum_{i=1}^{N} \frac{v_{it}}{u_i(x)},$

where $(a)$ is by definition of the algorithm, and $(b)$ is because promised utilities are a lower bound on the true final utility. Thus by weak duality,

$\max_{w \in \mathbb{R}^+_T} \frac{1}{N} \sum_{i=1}^{N} \frac{u_i(w)}{u_i(x)} \leq \gamma = 2 \ln(2N).$

To complete the proof, it remains to show that $x$ is a feasible allocation. Since $\sum_{t=1}^{T} y_t \leq \frac{1}{2}$, it suffices to show that $\sum_{t=1}^{T} z_t \leq \frac{1}{2}$. Now we have

$\gamma \sum_{t=1}^{T} z_t = \sum_{t=1}^{T} z_t \cdot \frac{1}{N} \sum_{i=1}^{N} \frac{v_{it}}{u_i(z_t)}$ (since $z_t > 0$ implies $\gamma = \frac{1}{N} \sum_{i=1}^{N} \frac{v_{it}}{u_i(z_t)}$)

$= \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \frac{z_t v_{it}}{u_i(z_t)} = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \left( 1 - \frac{\bar{u}_{it}(0)}{u_i(z_t)} \right)$

$\leq \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \left[ \ln(\bar{u}_{it}(z_t)) - \ln(\bar{u}_{it}(0)) \right]$ (since $1 - x \leq -\ln(x) \forall x \in \mathbb{R}$)

$= \frac{1}{N} \sum_{i=1}^{N} \left[ \ln(\bar{u}_{iT}(z_T)) - \ln(\bar{u}_{i1}(0)) \right] = \frac{1}{N} \sum_{i=1}^{N} \left[ \ln(\bar{u}_{iT}(z_T)) - \ln\left( \frac{1}{2N} \right) \right]$ (since predicted utility lower bounds final utility)

$= \frac{1}{N} \sum_{i=1}^{N} \ln(u_i(x)) + \ln(2N)$ (since final utilities are at most 1)

Hence, in order for $\sum_{t=1}^{T} z_t \leq \frac{1}{2}$, it suffices to have $\gamma = 2 \ln(2N)$. This is why we defined $\gamma$ to be equal to this quantity in the algorithm. \qed