Fair and Efficient Social Choice in Dynamic Settings

Rupert Freeman
Duke University
rupert@cs.duke.edu

Vincent Conitzer
Duke University
conitzer@cs.duke.edu

Seyed Majid Zahedi
Duke University
zahedi@cs.duke.edu

Abstract

We study a dynamic social choice problem in which an alternative is chosen at each round according to the reported valuations of a set of agents. In the interests of obtaining a solution that is both efficient and fair, we aim to maximize the long-term *Nash welfare*, which is the product of all agents' utilities. We present and analyze two greedy algorithms for this problem, including the classic Proportional Fair (PF) algorithm. We analyze several versions of the algorithms and how they relate, and provide an axiomatization of PF. Finally, we evaluate the algorithms on data gathered from a computer systems application.

1 Introduction

Fairness is a topic of rapidly increasing interest in social choice. On the one hand, there has been much recent interest in the fair allocation of resources—cake cutting [24] as well as other models [14, 23]. On the other hand, in voting, fairness considerations have received attention in selecting a committee, in the form of a focus on the voters being *represented* in the committee [9, 18, 5, 3].

A classical approach to obtaining a fair outcome in a context where agents have utility functions is to maximize the *Nash welfare* [19], which is the product of the agents' utilities. One attractive feature of using the maximum Nash welfare (MNW) solution is scale invariance: if an agent doubles all her utilities (or, equivalently, changes the units in which she expresses her utilities), this does not change which outcomes maximize the objective (the solution is not however stable under additive transformations, where an agent simply adds some constant value to all her reported utilities).

In life, it is often difficult to make a completely fair decision in a single-shot context; often, every option will leave some agents unhappy. Fortunately, we can often address this over time—we will go to my most preferred restaurant today, and to yours next week. Achieving fairness over time is the topic of our paper. Ours is certainly not the first work to consider fairness or social choice in dynamic settings [21, 17, 2].

When we make multiple decisions over time, we could simply maximize the Nash welfare in each round separately. But it is easy to see that this can lead to dominated outcomes. For example, suppose there are two agents, and we can choose an alternative that gives one a reward of 3, and the other a reward of 0; or vice versa; or an alternative that gives each of them 1. Within a round, the last alternative maximizes Nash welfare; but if this scenario is repeated every round, then it would be better to alternate between the first two alternatives, so that each agent obtains 1.5 per round on average. Of course, *initially*, say in the first round, we may not realize we will have these options every round, and so we may choose the last alternative; but if we do have these options every round, we should *eventually* catch on to this pattern and start alternating. Ideally, we would maximize the long-term Nash welfare, that is, the product of the long-run utilities (which are the sums of each agent's rewards), rather than, for example, the sum of the products within the rounds.

In this work, we do not focus primarily on strategic concerns. Of course it is fairly common to ignore strategic concerns in the social choice literature, but we do think this is an important topic for future work. On the other hand, there are also important contexts where strategic concerns do not come into play. For example, instead of considering a setting where there are multiple *agents* that have different utility functions, we can consider a setting where there are multiple *objectives* that each alternative contributes towards. For example, consider faculty hiring. Suppose the three objectives that we want our faculty hires to contribute to are research, teaching, and service; moreover, suppose that at the time of hiring we can predict well how much each candidate would contribute to each of these objectives, if hired. Then, it stands to reason that, one year, we may hire a top researcher that we do not expect to contribute much to the other objectives. But we would be loath to make such a decision *every* year; having hired a few top researchers who are not good at teaching or service, pressure will mount to address these needs. This fits well into our framework, if we simply treat each of the three objectives as an agent that is "happy" with an alternative to the extent to which it addresses the corresponding objective.

The rest of the paper is organized as follows. In Section 2 we introduce notation and preliminaries. In Section 3 we present two simple greedy algorithms for choosing alternatives, and provide intuitive interpretations of them, including an axiomatic justification for one of them. After presenting the algorithms, we evaluate them on data from a computer systems application in Section 4.

Justification for Nash welfare: The Nash welfare is frequently used as an objective in the fair division literature as it strikes a balance between maximizing efficiency and fairness [10, 12, 25]. Caragiannis *et al.* [2016] have recently shown the MNW solution to satisfy envy freeness up to one good, as well as approximating the maximin share guarantee. However, work in fair division focuses primarily on the allocation of *private* goods, where each alternative gives utility to exactly one agent. This is not the case in our setting, where each alternative can be valued positively by many agents. Conitzer *et al.* [2017] explicitly consider fairness axioms in the public good setting, including *proportionality*, which states that each agent should derive at least a $\frac{1}{n}$ fraction of the utility she could obtain by choosing the outcome at each round. It turns out that a proportional solution may not exist in our setting, but the MNW solution always satisfies a weaker criterion: For each agent i, there exists a round such that if i is given control of that round, then i achieves their utility guaranteed by proportionality.

We can also appeal to Nash's original axiomatization of the MNW solution [19] as the only solution that satisfies scale-freeness, Pareto optimality, independence of irrelevant alternatives, and symmetry, which are all natural in our setting (although without an explicit focus on fairness).

Related work: Parkes and Procaccia [2013] examine a similar problem by modeling agents' evolving preferences with Markov Decision Processes, with a reward function defined over states and actions (alternatives). However, their goal is to maximize the sum of (discounted) rewards and they do not explicitly consider fairness as an objective. Kash *et al.* [2014] examine a model of dynamic fair division where agents arrive at different times and must be allocated resources; however, they do not allow for the preferences of agents to change over time as we do. Aleksandrov *et al.* [2015] consider an online fair division problem in a setting where items appear one at a time, and agents declare yes/no preferences over that item. In our setting, each round has many alternatives and agents express more general utilities. Our work is related to the literature on dynamic mechanism design (Parkes *et al.* [2010] provide an overview), except that we do not consider monetary transfers. Guo *et al.* [2009] consider a setting similar to ours, also without money, except that they are not explicitly interested in fairness, only welfare, and focus on incentive compatibility.

2 Preliminaries

Consider a set of n agents and let $A = \{a_1, \ldots, a_m\}$ be a set of m possible alternatives.[1] At every round $t = 1, \ldots, T$, every agent i reports her valuation $v_i^t(a_j) \in \mathbb{N}$ for every alternative a_j.[2] Thus the input at every round is a matrix $V^t = (v_i^t(a_j))_{ij}$. Let $\boldsymbol{v^t(a_j)}$ denote the j-th column of matrix V^t, the vector of valuations for alternative a_j. For every round t, a *Dynamic Social Choice Function (DSCF)* chooses a set of alternatives C_t, from which a single alternative c_t is chosen arbitrarily. Importantly, the problem is *online*, so we may only use information up to time t in order to choose C_t.

We define a vector of *accrued rewards at round* t, $\boldsymbol{u_t}$, where the accrued reward of agent i at round t is the sum of i's valuations for the chosen alternatives up to and including round t, $u_t(i) = \sum_{t'=1}^{t} v_i^{t'}(c_{t'})$. We will often be interested in an agent's accrued reward *before* the start of round t, $u_{t-1}(i)$. For convenience, we will refer to the set of agents with $u_{t-1}(i) = 0$ by I_0 when the round, t, is clear. The average utility of the agents over the first t rounds is $\boldsymbol{u_t^{\textbf{avg}}} = \frac{1}{t}\boldsymbol{u_t}$.

A DSCF is *anonymous* if applying permutation σ to the agents, for all t, does not change the set of chosen alternatives C_t, for any t. A DSCF is *neutral* if applying permutation σ to the alternatives, for all t, results in choosing alternatives $\sigma(C_t)$ for all t. For the rest of this paper we only consider anonymous, neutral DSCFs.

The *Nash welfare (NW)* of valuation vector $\boldsymbol{u}$, $NW(\boldsymbol{u})$, is defined to be the product of the agents' utilities, $NW(\boldsymbol{u}) = \prod_{i=1}^{n} u(i)$. We also define $NW^+(\boldsymbol{u}) = \prod_{i:u(i)\neq 0} u(i)$ to be the product of all positive entries of $\boldsymbol{u}$. Our aim is to maximize the NW of the average utility across all T rounds, $NW(\boldsymbol{u_T^{\textbf{avg}}})$. Note that while our setting allows for discounting, we do not need to explicitly address it since the input matrices can be pre-multiplied by the necessary factor before being passed as input to the DSCF.

The benchmark algorithm is the optimal algorithm for the offline problem, where an instance is given by the set $\{V^t\}_{t\in\{1,\ldots,T\}}$, and can be solved by a mixed integer convex program. We denote the optimal Nash welfare by OPT.

Our algorithms and analysis use a formal infinitesimal quantity ϵ. Numbers involving ϵ take the form $\sum_{i=-\infty}^{i=\infty} a_i\epsilon^i$.[3] For two such numbers $a = \sum_{i=-\infty}^{i=\infty} a_i\epsilon^i$ and $b = \sum_{i=-\infty}^{i=\infty} b_i\epsilon^i$, let i' be the smallest index for which $a_i \neq b_i$, if it exists. Then $a > b$ if and only if $a_{i'} > b_{i'}$. That is, we compare numbers lexicographically by the lowest powers of ϵ. Two numbers are equal if all coefficients are equal.

[1] For simplicity of presentation, we define the set of alternatives to be static. However, all of our algorithms and results hold if the set, and even the number, of alternatives changes from round to round.

[2] Restricting valuations to be non-negative integers is necessary for some of our results in Section 3. This is still sufficient for agents to express their preferences to arbitrary levels of precision.

[3] While our framework allows for unbounded powers of ϵ, in this paper we utilize only powers of ϵ between ϵ^{-1} and ϵ^n.

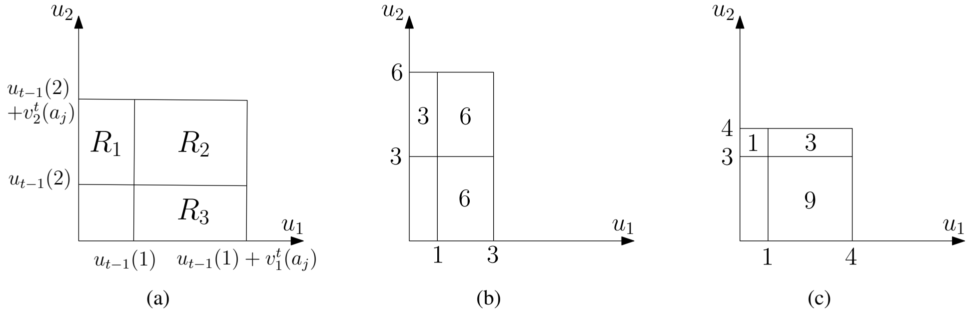

Figure 1: Illustration of the difference between GREEDY and PF for an instance with two agents. The horizontal axis represents agent 1's reward, and the vertical axis represents agent 2's reward. Figure 1(a) shows a general instance. GREEDY chooses the alternative that maximizes area $R_1 + R_2 + R_3$, while PF chooses the alternative that maximizes $R_1 + R_3 = v_2^t(a_j)u_{t-1}(1) + v_1^t(a_j)u_{t-1}(2) = u_{t-1}(1)u_{t-1}(2)\left[\frac{v_1^t(a_j)}{u_{t-1}(1)} + \frac{v_2^t(a_j)}{u_{t-1}(2)}\right]$. Figures 1(b) and 1(c) illustrate the choice of alternative a_1 and a_2 in Example 1, respectively.

3 Greedy Algorithms

3.1 Algorithm Definitions

In this section we present two greedy algorithms. We note that, although these algorithms are designed to give an approximate solution to that which maximizes Nash welfare, much of this section is devoted to showing that they satisfy desirable properties as algorithms in their own right. Such an approach is not new in computational social choice – for other papers that treat approximation algorithms as distinct voting rules see, for example, [6, 7, 13]. The first algorithm, GREEDY, simply chooses c_t to maximize $NW(\boldsymbol{u}_t^{\mathbf{avg}})$, the Nash welfare at the end of the round. The second algorithm is a linearized version of greedy known as PROPORTIONALFAIR (PF) in the networking community [26, 16], which maximizes the sum of percentage increases in accrued reward at each round. Equivalently, it works by assigning each agent a weight w_i (denote the vector of weights by $\boldsymbol{w}$) equal to the inverse of her accrued reward at the start of each round and chooses $C_t = \mathrm{argmax}_{a_j \in A}\, \boldsymbol{w} \cdot \boldsymbol{v^t(a_j)}$, the alternatives that maximize the weighted sum of valuations. Note that w_i is proportional to the product of the other agents' accrued rewards.

Example 1. *Let $n = m = 2$ and suppose that $u_{t-1}(1) = 1$, $u_{t-1}(2) = 3$, and $V^t = \left(\begin{smallmatrix} 2 & 3 \\ 3 & 1 \end{smallmatrix}\right)$. That is, agent 1 has valuation 2 for alternative a_1 and valuation 3 for alternative a_2. Agent 2 has valuation 3 for alternative a_1 and valuation 1 for alternative a_2. Choosing a_1 results in Nash welfare of $(1 + 2) \cdot (3 + 3) = 18$, while choosing a_2 results in Nash welfare of $(1 + 3) \cdot (3 + 1) = 16$. Thus* GREEDY *chooses a_1.*

Under PF*, each agent is given weight inversely proportional to their own accrued utility. That is, agent 1 has weight 1 and agent 2 has weight $\frac{1}{3}$. Now, taking the weighted sum of valuations yields $(1 \cdot 2) + (\frac{1}{3} \cdot 3) = 3$ for alternative a_1, and $(1 \cdot 3) + (\frac{1}{3} \cdot 1) = \frac{10}{3}$ for alternative a_2. Thus* PF *chooses a_2.*

A graphical illustration of the difference between the two algorithms is given in Figure 1.

Unfortunately, both algorithms encounter problems while there exist agents with zero accrued reward. For GREEDY, it can (and, unless some alternative is valued positively by all agents, will) be the case that $NW(\boldsymbol{u}_t^{\mathbf{avg}}) = 0$ for all choices of c_t, even when one alternative is weakly preferred to all other alternatives by all agents. For PF, it is impossible to set a weight $w_i = \frac{1}{u_{t-1}(i)}$ for an agent with $u_{t-1}(i) = 0$.

As a general framework for addressing this issue, we endow each agent $i \in I_0$ with some arbitrary, infinitesimal reward at the start of each round. This is a natural way to allow the algorithms to give high priority to agents with zero accrued reward while avoiding mathematical inconsistencies, and it allows us to efficiently choose an alternative c_t if we are happy with selecting any member of the choice set C_t.

However, once we endow rewards (even infinitesimal ones), we immediately lose scale-freeness, one of the appealing properties of using Nash welfare. Further, if we want to choose a member of the choice set C_t uniformly at random, there is no obvious distribution over endowed rewards that allows us to do this – choosing endowed rewards uniformly at random from some interval will not, in general, result in drawing uniformly from C_t. So, while the technique of randomly endowing infinitesimal reward is a general and intuitive way for the algorithms to handle all situations, we also want an algorithm to compute the entire choice set C_t.

In the following, for both GREEDY and PF, we first present the algorithm to select a single alternative via nondeterministically endowing infinitesimal reward, followed by an algorithm to compute the entire choice set C_t.

Algorithm 1 GREEDY (select one alternative)

Input $\boldsymbol{u_{t-1}}$
for $i = 1, \ldots, n$ **do**
    Randomly choose $0 < x_i \leq 1$
end for
Return $c_t \in \text{argmax}_{a_j \in A} \prod_{i=1}^{n} \max\{u_{t-1}(i) + v_i^t(a_j), x_i \epsilon\}$

The alternatives chosen by Algorithm 1 are exactly the alternatives that result in a maximal number of agents with positive accrued reward and, subject to holding fixed the set of agents with positive accrued reward, maximizes the product of these agents' rewards.

Algorithm 2 GREEDY (select all alternatives)

Input $\boldsymbol{u_{t-1}}$
$C_t \leftarrow \text{argmax}_{a_j \in A} |\{i : u_{t-1}(i) + v_i^t(a_j) > 0\}|$
for $j \in C_t$ **do**
    if $\exists j'$ such that
$\{i : u_{t-1}(i) + v_i^t(a_j) > 0\} = \{i : u_{t-1}(i) + v_i^t(a_{j'}) > 0\}$ and $NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(a_j)}) < NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(a_{j'})})$ **then**
        $C_t \leftarrow C_t \backslash \{a_j\}$
    end if
end for
Return C_t

The version of PF for selecting a single alternative is presented as Algorithm 3.

To determine the complete choice set C_t, we solve a linear program for each alternative that explicitly determines whether there is some infinitesimal endowment that results in the alternative being chosen by PF.

A notable difference in the algorithms is that unlike GREEDY, PF may leave some agents with zero accrued utility even when it was possible to give positive utility to all agents.

Algorithm 3 PROPORTIONALFAIR (select one alternative)

Input $\boldsymbol{u_{t-1}}$
for $i \in I_0$ **do**
  Randomly choose $0 < x_i \leq 1$
  Randomly choose $y_i \in \mathbb{R}$
end for
$w_i \leftarrow \begin{cases} x_i \frac{1}{\epsilon} + y_i, & \text{if } u_{t-1}(i) = 0 \\ \frac{1}{u_{t-1}(i)}, & \text{if } u_{t-1}(i) > 0 \end{cases}$
Return $c_t \in \operatorname{argmax}_{a_j \in A} \boldsymbol{w} \cdot \boldsymbol{v^t(a_j)}$

Algorithm 4 PROPORTIONALFAIR (select all alternatives)

Input u_{t-1}
$C_t \leftarrow \emptyset$
for $j = 1, \ldots, m$ **do**
  if the following linear program is unbounded

$$\begin{aligned} &\text{Maximize } L \\ &\text{subject to } \boldsymbol{w'} \cdot \boldsymbol{v^t(a_j)} \geq \boldsymbol{w'} \cdot \boldsymbol{v^t(a_{j'})} && \forall j' \\ &\quad w'_i = \frac{1}{u_{t-1}(i)} && \forall i \text{ such that } u_{t-1}(i) > 0 \\ &\quad w'_i \geq L && \forall i \text{ such that } u_{t-1}(i) = 0 \end{aligned}$$

  then
    $C_t \leftarrow C_t \cup \{a_j\}$
  end if
end for
Return C_t

Example 2. *Let $n = 2$, $m = 3$, and $t = 1$. Suppose that $V_1 = \left(\begin{smallmatrix} 3 & 0 & 1 \\ 0 & 3 & 1 \end{smallmatrix}\right)$. Because $t = 1$, $u_{t-1}(1) = u_{t-1}(2) = 0$.*

GREEDY *chooses a_3 since it is the only alternative that provides non-zero reward to both agents. However,* PF *assigns the agents weights w_1, w_2 and chooses* $\operatorname{argmax}_{j \in \{1,2,3\}} \boldsymbol{w} \cdot \boldsymbol{v^t(a_j)}$. *Since it must be the case that either $3w_1 > w_1 + w_2$ or that $3w_2 > w_1 + w_2$, it is not possible for a_3 to be chosen by* PF.

For each algorithm, we prove equivalence of the two versions in the sense that the set generated by the 'select all' version consists exactly of the alternatives that the 'select one' version generates for some nondetermistic choices.

Theorem 1. *The set of alternatives C_t chosen by Algorithm 2 at round t is exactly the set of alternatives that can be chosen at round t by Algorithm 1.*

The proof uses the fact that the product on Line 5 of Algorithm 1 is maximized when the number of ϵ terms appearing in the product is minimized.

Proof. We begin by showing that every alternative that can be selected by Algorithm 1 is also selected by Algorithm 2. Let c_t be an alternative chosen by Algorithm 1 for some choices of $\{x_i\}$ and let $p = |\{i : u_{t-1}(i) + v_i^t(c_t) > 0\}|$. Therefore, the lowest power of

ϵ with non-zero coefficient in the product on Line 5 of Algorithm 1 is ϵ^{n-p}. If some other alternative a_j has $|\{i : u_{t-1}(i) + v_i^t(a_j) > 0\}| > p$ then the corresponding product has non-zero coefficient on a lower power of ϵ, contradicting optimality of c_t. That is, $c_t \in \text{argmax}_{a_j \in A} |\{i : u_{t-1}(i) + v_i^t(a_j) > 0\}|$.

Next, let $a_{j'}$ be an alternative with $\{i : u_{t-1}(i) + v_i^t(c_t) > 0\} = \{i : u_{t-1}(i) + v_i^t(a_{j'}) > 0\}$. The product on Line 5 of Algorithm 1 is

$$NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(c_t)})\epsilon^{n-p} \prod_{i:u_{t-1}(i)+v_i^t(c_t)=0} x_i$$

for c_t and

$$\begin{aligned} &NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(a_{j'})})\epsilon^{n-p} \prod_{i:u_{t-1}(i)+v_i^t(a_{j'})=0} x_i \\ =&NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(a_{j'})})\epsilon^{n-p} \prod_{i:u_{t-1}(i)+v_i^t(c_t)=0} x_i \end{aligned}$$

for alternative $a_{j'}$. Since c_t is chosen by Algorithm 1, it must be the case that $NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(c_t)}) \geq NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(a_{j'})})$. Therefore, c_t is chosen by Algorithm 2.

To complete the proof, we show that every alternative selected by Algorithm 2 can also be selected by Algorithm 1. To that end, let $c_t \in C_t$. We exhibit a specific choice of $\{x_i\}$ which results in c_t being selected by Algorithm 1. Let K be some integer greater than the largest entry in V^t and let

$$x_i = \begin{cases} \frac{1}{2(K+1)^n}, & \text{if } u_{t-1}(i) + v_i^t(c_t) > 0 \\ 1 & \text{if } u_{t-1}(i) + v_i^t(c_t) = 0. \end{cases}$$

Then the product on Line 5 of Algorithm 1 that results from c_t being selected is

$$NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(c_t)})\epsilon^{n-p},$$

where $p = |\{i : u_{t-1}(i) + v_i^t(c_t) > 0\}|$. Now consider some alternative $a_j \neq c_t$. If $\{i : u_{t-1}(i) + v_i^t(c_t)) > 0\} = \{i : u_{t-1}(i) + v_i^t(a_j) > 0\}$ and $NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(c_t)}) \geq NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(a_j)})$ then the leading term in the product on Line 5 of Algorithm 1 that results from a_j being selected is

$$NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(a_j)})\epsilon^{n-p} \leq NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(c_t)})\epsilon^{n-p}.$$

Similarly, an alternative a_j with $|\{i : u_{t-1}(i) + v_i^t(a_j) > 0\}| < |\{i : u_{t-1}(i) + v_i^t(c_t) > 0\}|$ has coefficient 0 for the ϵ^{n-p} term (and larger terms) in the corresponding product on Line 5. In both cases, this product is greater for c_t than for a_j.

The final case is when $|\{i : u_{t-1}(i) + v_i^t(c_t) > 0\}| = |\{i : u_{t-1}(i) + v_i^t(a_j) > 0\}|$ but the two sets are not equal. In this case, the dominant term in the product on Line 5 of Algorithm 1 that results from a_j being selected is at most

$$NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(a_j)})\frac{1}{2(K+1)^n}\epsilon^{n-p}$$

by the choice of $\{x_i\}$ and noting that at least one agent with $u_{t-1}(i) + v_i^t(c_t) > 0$ has $u_{t-1}(i) + v_i^t(a_j) = 0$. But, since the maximum reward any agent derives from any alternative is K,

$$\begin{aligned} NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(a_j)}) &\leq (K+1)^p NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(c_t)}) \\ &\leq (K+1)^n NW^+(\boldsymbol{u_{t-1}} + \boldsymbol{v^t(c_t)}). \end{aligned}$$

Therefore,

$$NW^{+}(\boldsymbol{u_{t-1}} + \boldsymbol{v^{t}(a_j)})\frac{1}{2(K+1)^n} \le (K+1)^n NW^{+}(\boldsymbol{u_{t-1}} + \boldsymbol{v^{t}(c_t)})\frac{1}{2(K+1)^n}$$
$$< NW^{+}(\boldsymbol{u_{t-1}} + \boldsymbol{v^{t}(c_t)}),$$

so the product from Line 5 of Algorithm 1 is larger for c_t than for a_j. Hence the particular choice of $\{x_i\}$ results in c_t being chosen by Algorithm 1. □

Theorem 2. *The set of alternatives C_t chosen by Algorithm 4 at round t is exactly the set of alternatives that can be chosen at round t by Algorithm 3.*

Proof. We begin by showing that every alternative that can be selected by Algorithm 3 is also selected by Algorithm 4. Let c_t be an alternative chosen by Algorithm 3 for some choices of $\{x_i\}_{i\in I_0}$ and $\{y_i\}_{i\in I_0}$. For all $i \notin I_0$, set $w'_i = \frac{1}{u_{t-1}(i)}$, and for all $i \in I_0$, set $w'_i = \frac{x_i}{\delta} + y_i$ for any $\delta > 0$. As $\delta \to 0$, the variables w'_i grow arbitrarily large. Therefore, to show feasibility of the variables $\{w'_i\}$ we need to show that the first set of constraints in the LP in Algorithm 4 hold for sufficiently small δ.

Fix an alternative a_j. From Line 7 of Algorithm 3, we know that $\boldsymbol{w}\cdot\boldsymbol{v^{t}(c_t)} \ge \boldsymbol{w}\cdot\boldsymbol{v^{t}(a_j)}$. The dominant coefficient in this expression is that of ϵ^{-1}. Comparing these coefficients gives us

$$\sum_{i\in I_0} x_i v_i^t(c_t) \ge \sum_{i\in I_0} x_i v_i^t(a_j). \tag{1}$$

If Inequality 1 is strict, then we know that $\sum_{i\in I_0} \frac{x_i}{\delta} v_i^t(c_t) > \sum_{i\in I_0} \frac{x_i}{\delta} v_i^t(a_j)$ for any $\delta > 0$, and we can make the gap arbitrarily large by setting δ sufficiently small. In particular, we can force the gap to be large enough that the following inequality holds for any fixed values of $\{y_i\}_{i\in I_0}$ and $\{u_{t-1}(i)\}_{i\notin I_0}$:

$$\sum_{i\in I_0}\left(\frac{x_i}{\delta} + y_i\right) v_i^t(c_t) + \sum_{i\notin I_0}\frac{1}{u_{t-1}(i)} v_i^t(c_t) > \sum_{i\in I_0}\left(\frac{x_i}{\delta} + y_i\right) v_i^t(a_j) + \sum_{i\notin I_0}\frac{1}{u_{t-1}(i)} v_i^t(a_j),$$

which is precisely the first constraint in the linear program from Algorithm 4.

If Inequality 1 holds with equality, then we turn attention to the coefficient of ϵ^0 in the dot product from Line 7 of Algorithm 3. This tells us that

$$\sum_{i\in I_0} y_i v_i^t(c_t) + \sum_{i\notin I_0}\frac{1}{u_{t-1}(i)} v_i^t(c_t) \ge \sum_{i\in I_0} y_i v_i^t(a_j) + \sum_{i\notin I_0}\frac{1}{u_{t-1}(i)} v_i^t(a_j). \tag{2}$$

Dividing Inequality 1 by δ and adding Inequality 2 gives $\sum_{i=1}^n w'_i v_i^t(c_t) \ge \sum_{i=1}^n w'_i v_i^t(a_j)$, satisfying the first constraint of the LP, so the weights $\{w'_i\}$ are feasible. These weights allow us to set L to arbitrarily large values as $\delta \to 0$, so the LP is unbounded and Algorithm 4 selects c_t.

We now show the other direction, that every alternative selected by Algorithm 4 can also be selected by Algorithm 3. Let $c_t \in C_t$. That is, the optimal value for the LP in Algorithm 4 is unbounded. Then it is the case that there exist vectors $\mathbf{p}$ and $\mathbf{q} \neq \mathbf{0}$ for the values of the variables in the LP such that $\mathbf{p} + k\mathbf{q}$ is feasible for all $k > 0$ and $\mathbf{q}$ has positive objective value (this is a known fact about linear programs with unbounded value; see, e.g., [20], Theorem 4.7). We use these to exhibit values of $\{x_i\}_{i\in I_0}$ and $\{y_i\}_{i\in I_0}$ so that c_t is chosen by Algorithm 3.

Set $y_i = p_i$ and $x_i = q_i$ for all $i \in I_0$. Let $a_j \in A$. By the first set of constraints from the LP,

$$\sum_{i \in I_0}(p_i + kq_i)v_i^t(c_t) + \sum_{i \notin I_0} \frac{1}{u_{t-1}(i)} v_i^t(c_t) \geq \sum_{i \in I_0}(p_i + kq_i)v_i^t(a_j) + \sum_{i \notin I_0} \frac{1}{u_{t-1}(i)} v_i^t(a_j) \quad (3)$$

for all $k > 0$. In particular, this implies that it can not be the case that $\sum_{i \in I_0} q_i v_i^t(c_t) < \sum_{i \in I_0} q_i v_i^t(a_j)$, or else Inequality 3 would be violated for large enough values of k. There are two possiblities.

First, suppose that $\sum_{i \in I_0} q_i v_i^t(c_t) > \sum_{i \in I_0} q_i v_i^t(a_j)$. Then, by our choice of $x_i = q_i$ for all $i \in I_0$, we have that $\sum_{i \in I_0} x_i v_i^t(c_t) > \sum_{i \in I_0} x_i v_i^t(a_j)$. But, as discussed earlier, $\sum_{i \in I_0} x_i v_i^t(a_j)$ is exactly the dominant term in Line 7 of Algorithm 3. Therefore, this dot product is maximized by c_t, so c_t is chosen by Algorithm 3.

Finally, suppose that $\sum_{i \in I_0} q_i v_i^t(c_t) = \sum_{i \in I_0} q_i v_i^t(a_j)$. So the dominant term in Line 7 of Algorithm 3 is equal for c_t and a_j. By Inequality 3, it must be the case that

$$\sum_{i \in I_0} p_i v_i^t(c_t) + \sum_{i \notin I_0} \frac{v_i^t(c_t)}{u_{t-1}(i)} \geq \sum_{i \in I_0} p_i v_i^t(a_j) + \sum_{i \notin I_0} \frac{v_i^t(a_j)}{u_{t-1}(i)}.$$

By the choice of $y_i = p_i$ for all $i \in I_0$, the above inequality holds when we substitute y_i for every instance of p_i. After making that substitution, we are left with exactly the expression for the coefficient of ϵ^0 in Line 7 of Algorithm 3. Since the coefficient is at least as large for c_t as for a_j, and the ϵ^{-1} coefficients are equal (and there are no further non-zero terms), c_t may be chosen by Algorithm 3. □

3.2 Axiomatization of PROPORTIONALFAIR

Now that we have given a precise definition of the PF mechanism and justified it, in this section we provide an axiomatization of the PF mechanism.

A DSCF is scale-free if it is not affected by a uniform (multiplicative) scaling of some agent's valuations. This property is desirable because it means we do not require any sort of agreement or synchronization as to the units of measurement used by the agents in their reporting.

Definition 1. *Let $k > 0$. Say that a DSCF satisfies* scale-free-ness (SF) *if C_t is unchanged (for the same choice of tiebreaking in earlier rounds) if we replace $v_i^t(a_j)$ by $k \cdot v_i^t(a_j)$ for all $a_j \in A$ for every $t = 1, \ldots, T$.*

Lemma 1. PF *satisfies SF.*

Proof. Let $c \in C_t$ and suppose that agent i scales all her valuations by $k > 0$. We show by induction that PF still chooses c at round t. Consider a round t such that the chosen alternative is unchanged in all previous rounds.

Suppose that $u_{t-1}(i) = 0$. So for any L there exists vector of weights $\mathbf{w}'$ such that $w'_i \geq L$ and alternative c maximizes the weighted sum of valuations. After i scales her valuations by a factor of k, we can simply scale w'_i by a factor of $\frac{1}{k}$ (this will still allow unbounded values of w'_i). Therefore, the value $\mathbf{w}' \cdot \mathbf{v^t}(\mathbf{a_j})$ is unchanged for every alternative a_j. Thus, alternative c still maximizes this expression.

Now suppose that $u_{t-1}(i) > 0$. Then i's weight w'_i in the scaled instance is a factor of k smaller than in the un-scaled instance, but $v_i^t(a_j)$ is a factor of k larger than in the un-scaled instance for all alternatives a_j. Thus, for any setting of weights $\{w'_{i'}\}_{i' \neq i}$ in the un-scaled instance, the value $\mathbf{w}' \cdot \mathbf{v^t(a_j)}$ is unchanged in the scaled instance. Thus, the existence of a feasible set of weights such that c is chosen in the unscaled instance implies that c is chosen in the scaled instance also, for the same choice of weights.

Finally we need to rule out the possibility that some new alternative, $a_j \notin C_t$, is chosen at round t in the scaled instance. But if this were the case, then we can just scale the scaled instance by $\frac{1}{k}$ and return to the original instance where, by the above proof, $a_j \in C_t$. □

A DSCF is separable into single-minded agents if the chosen alternative at a round is unchanged by replacing an agent by several new agents with the same accrued reward, each of which has unit positive valuation for only one alternative. The axiom reflects that we can interpret utilities cardinally rather than just ordinally.

Definition 2. *Say that a DSCF is* separable into single-minded agents (SSMA) *if, when all agents have the same accrued utility $u_{t-1}(i) = u > 0$, C_t is unchanged if we replace each agent with several new agents (denoted generically by x) according to the following scheme: For every $v_i^t(a_j) \in V^t$, create $v_i^t(a_j)$ agents each with $u_{t-1}(x) = u$, $v_x^t(a_j) = 1$, and $v_x^t(a_{j'}) = 0$ for all $j' \neq j$.*

Lemma 2. PF *satisfies SSMA.*

Proof. Consider round t with valuation matrix V^t. PF chooses all alternatives that maximize the expression

$$\sum_{i=1}^{n} \frac{1}{u} v_i^t(a_j). \tag{4}$$

Now consider the instance expanded as defined by Definition 2. For every alternative a_j, there are exactly $\sum_{i=1}^{n} v_i^t(a_j)$ agents that have valuation 1 for a_j being chosen, while all other agents have valuation 0. Since each new agent has accrued utility u, PF chooses all alternatives which maximize Equation 4. □

The plurality axiom says that if all agent valuation vectors are unit vectors, and we have no reason to distinguish between agents, then the alternatives favored by the most agents should be chosen.

Definition 3. *Say that a DSCF satisfies* plurality (P) *if, when all agents have unit valuation for only a single alternative, and all agents have the same (non-zero) accrued utility, then C_t consists of the alternatives with non-zero valuation from the most agents.*

Plurality says nothing about the case when some agent has $u_{t-1}(i) = 0$. The idea of the axiom (in combination with SF) is that we should choose the alternative which provides the greatest utility, relative to what agents already have. However, if agents have zero accrued reward then it is not possible to make accurate comparisons as to the relative benefit each agent receives.

Observation 1. PF *satisfies plurality.*

The final axiom says that, if we restrict attention to only agents with zero accrued reward, alternatives which are dominated by a mixture of other alternatives should not be played. In the case that two alternatives are equivalent with respect to agents with $u_{t-1}(i) = 0$, we should only choose an alternative if it would still be chosen in the absence of the agents with $u_{t-1}(i) = 0$. The definition is inspired by mixed strategy dominance in game theory and, intuitively, formalizes that we should prioritize agents with zero utility above all others.

We first define the notion of 0-dominance.

Definition 4. *Let $z_1, \ldots, z_m$ be nonnegative coefficients with $\sum_{j'} z_{j'} = 1$. We say that an alternative a_j is* strictly 0-dominated *by the mixture of alternatives $\sum_{j'} z_{j'} a_{j'}$ at round t if $\sum_{j'} z_{j'} v_i^t(a_{j'}) \geq v_i^t(a_j)$ for all agents i with $u_{t-1}(i) = 0$, with at least one of these inequalities being strict. If all inequalities hold with equality, then we say that a_j is* weakly 0-dominated *by the mixture $\sum_{j'} z_{j'} a_{j'}$.*

We say that a_j is (strictly, weakly) 0-dominated if there exists some mixture of alternatives that (strictly, weakly) 0-dominates it.

Definition 5. *A DSCF f satisfies* No 0-Dominated Alternatives (NZDA) *if it never chooses a strictly 0-dominated alternative, and chooses a weakly 0-dominated alternative a_j only if a_j would be chosen by f under a scenario where V^t was modified to include (1) only the agents with $u_{t-1}(i) > 0$, and (2) only the (mixtures of) alternatives that weakly 0-dominate a_j (including a_j itself).*

Lemma 3. PF *satisfies NZDA.*

Proof. Let a_j be a strictly 0-dominated alternative. Note that the dominant coefficient in Line 7 of Algorithm 3 is that of ϵ^{-1}, which is determined by the values of $\{x_i\}_{i \in I_0}$. Therefore, an alternative is chosen by PF only if it maximizes $\sum_{i \in I_0} x_i v_i^t(a_j)$. So, to show that a_j is not selected by PF, it suffices to show that there does not exist any allowed choice of $\{x_i\}$ for which

$$\sum_{i \in I_0} x_i v_i^t(a_j) \geq \sum_{i \in I_0} x_i v_i^t(a_{j'})$$

for all other alternatives $a_{j'}$.

Fix $\{x_i\}_{i \in I_0}$, and consider drawing an alternative $a_{j'}$ from the distribution defined by the weights $z_1, \ldots, z_m$. By the dominance condition and the fact that all x_i are positive,

$$\sum_{i \in I_0} x_i v_i^t(a_j) < \sum_{i \in I_0} x_i v_i^t(a_{j'})$$

in expectation. Thus there must exist a particular j' for which the above inequality holds, so a_j is not chosen by PF.

Fix a choice of $\{x_i, y_i\}_{i \in I_0}$ and let a_j be a weakly 0-dominated alternative – suppose that it is weakly 0-dominated by alternative $a_{j'}$ (which may be a mixture of several alternatives). Since $v_i^t(a_{j'}) = v_i^t(a_j)$ for all agents $i \in I_0$,

$$\sum_{i \in I_0} (\frac{x_i}{\epsilon} + y_i) v_i^t(a_j) = \sum_{i \in I_0} (\frac{x_i}{\epsilon} + y_i) v_i^t(a_{j'}).$$

Suppose that a_j is chosen by PF. Then, by the definition of PF,

$$\sum_{i \in I_0} (\frac{x_i}{\epsilon} + y_i) v_i^t(a_j) + \sum_{i \notin I_0} \frac{1}{u_{t-1}(i)} v_i^t(a_j) \geq \sum_{i \in I_0} (\frac{x_i}{\epsilon} + y_i) v_i^t(a_{j'}) + \sum_{i \notin I_0} \frac{1}{u_{t-1}(i)} v_i^t(a_{j'}),$$

which requires that

$$\sum_{i \notin I_0} \frac{1}{u_{t-1}(i)} v_i^t(a_j) \geq \sum_{i \notin I_0} \frac{1}{u_{t-1}(i)} v_i^t(a_{j'}). \tag{5}$$

Equation 5 exactly says that PF would still choose a_j if only alternatives that weakly 0-dominated a_j were included in V^t, and in the absence of all agents with $u_{t-1}(i) = 0$, which completes the proof. □

We now show that any mechanism that achieves SF, SSMA, P, and NZDA simultaneously must agree with PF. We note that of the four axioms, GREEDY satisfies only SF and P. Despite GREEDY being (arguably) simpler than PF, we do not know a good axiomatization for it.

Theorem 3. *Let f be a DSCF that satisfies SF, SSMA, P, and NZDA. Suppose that f chooses alternative c_t at round t. Then* PF *must also choose c_t at round t (for the same history up to that point).*

Proof. We have already shown that PF satisfies SF, SSMA, P, and NZDA.

It remains to show that f's choice of alternative can also be chosen by PF.

First suppose that all agents have $u_{t-1}(i) > 0$. Without loss of generality, let $u_{t-1}(i) = u$ for all agents i. We may assume this because, by SF, f and PF would choose the same alternatives at round t even if the valuation vectors of some agent(s) were multiplied by a constant across all rounds. Multiplying each agent i's valuations by $\prod_{i' \neq i} u_{t-1}(i')$, we obtain an instance in which all agents have the same accrued utility, $\prod_i u_{t-1}(i)$.

By SSMA, we can replace the agent i with $\sum_{j=1}^{m} v_i^t(a_j)$ agents, such that $v_i^t(a_j)$ of them have unit valuation for alternative a_j (and 0 valuation for all other alternatives), all with accrued reward u. Then, by plurality, f chooses $c_t \in \text{argmax}_{a_j \in V^t} \sum_{i=1}^{n} v_i^t(a_j)$. Note that PF assigns equal weight w_i to each agent since $u_{t-1}(i) = u_{t-1}(i')$ for all i, i'. Thus PF chooses precisely the alternatives which maximize $\sum_{i=1}^{n} v_i^t(a_j)$, which includes any alternative chosen by f.

The more intricate case is when there exists at least one agent with $u_{t-1}(i) = 0$. Since f satisfies NZDA, we know that f never chooses a strictly 0-dominated alternative and only chooses a weakly 0-dominated alternative if f would still choose that alternative when V^t is modified according to Definition 5. To complete the proof, we show that PF selects all alternatives that can possibly be chosen by f. Specifically, we show that PF can choose all alternatives that are not (strictly or weakly) 0-dominated, as well as any weakly 0-dominated alternative a_{j^*} that is chosen by PF for the modified V^t. That is, when all alternatives are removed other than a_{j^*} and (mixtures of) alternatives that weakly 0-dominate it, and all agents with $u_{t-1}(i) = 0$ are removed. This is sufficient since we have shown that PF chooses every alternative chosen by f when all agents have $u_{t-1}(i) > 0$ (which is the case when all agents with $u_{t-1}(i) = 0$ are removed).

An alternative a_{j^*} is either (a) strictly 0-dominated, or (b) weakly 0-dominated and not chosen by PF when V^t is modified according to Definition 5, if and only if the optimal value of the following LP is negative for arbitrarily large values of H. We omit the index of the round t from the agents' valuations for clarity.

$$\text{Minimize } H \sum_{i \in I_0} \sum_{a_j \in A} (v_i(a_{j^*}) - v_i(a_j)) z_j + \sum_{i \notin I_0} \sum_{a_j \in A} \frac{1}{u_{t-1}(i)} (v_i(a_{j^*}) - v_i(a_j)) z_j \tag{6}$$

$$\text{subject to } \sum_{a_j \in A} v_i(a_j) z_j \geq v_i(a_{j^*}) \qquad \forall i \in I_0$$

$$\sum_{a_j \in A} z_j = 1$$

$$z_j \geq 0 \qquad \forall j$$

If a_{j^*} is strictly dominated then the first term of the objective can be made negative (and therefore the whole objective can be made negative when H is large enough). If a_{j^*} is only weakly dominated, then the first term can be set to 0, and the second term to be negative when there exists a mixture of alternatives that is chosen by PF (ahead of a_{j^*}) according to the modified V^t. Conversely, if the optimal value of the objective is negative then either

Table 1: Spark Applications

Category	Applications
Statistics	`Correlation`
Classification	`DecisionTree, GradientBoostedTrees, SVM, LinearRegression, NaiveBayesian`
Pattern Mining	`FP Growth`
Clustering	`KMeans`
Collaborative Filtering	`ALS`
Graph Processing	`Pagerank, ConnectedComponents, TriangleCounting`

there exist values for $\{z_j\}$ such that the first term is negative (which, combined with the first set of constraints, says that a_{j^*} is strictly 0-dominated), or there exist values for $\{z_j\}$ such that the first term is zero and the second term is negative. If the second term is negative then the weighted sum of valuations for the mixed alternative defined by $\{z_j\}$ is higher than the weighted sum of valuations for a_{j^*}, for the weights defined by PF when restricted to agents $i \notin I_0$. This proves correctness of the LP.

We want to show that PF can choose any alternative for which the the optimal value of LP (6) is nonnegative. Let a_{j^*} be such an alternative. We show that a_{j^*} can be chosen by PF by considering the dual, which has variables w_i for all $i \in I_0$ (one for each constraint) and r (corresponding to the constraint on the sum of the z_j):

$$\begin{aligned}
\text{Maximize } & \sum_{i \in I_0} v_i(a_{j^*}) w_i - r \\
\text{subject to } & \sum_{i \in I_0} v_i(a_j) w_i - r \leq H \sum_{i \in I_0} (v_i(a_{j^*}) - v_i(a_j)) \\
& \quad + \sum_{i \notin I_0} \frac{v_i(a_{j^*}) - v_i(a_j)}{u_{t-1}(i)} \quad \forall j \in \{1, \ldots, m\} \\
& w_i \geq 0 \quad \forall i \in I_0
\end{aligned}$$

Let $\overline{r} = \sum_{i \in I_0} v_i(a_{j^*}) w_i - r$ denote the objective. The first set of constraints can now be rewritten as

$$\overline{r} + \sum_{i \in I_0} (w_i + H) v_i(a_j) + \sum_{i \notin I_0} \frac{1}{u_{t-1}(i)} v_i(a_j) \leq \sum_{i \in I_0} (w_i + H) v_i(a_{j^*}) + \sum_{i \notin I_0} \frac{1}{u_{t-1}(i)} v_i(a_{j^*}).$$

Since a_{j^*} is not 0-dominated, the optimal value of LP (6) is at least zero for any arbitrarily large value of H. By strong duality, the optimal value of the dual is therefore also at least zero for arbitrarily large values of H. Thus, if we set $w'_i = w_i + H$ for all $i \in I_0$ and $w'_i = \frac{1}{u_{t-1}(i)}$ for all $i \in I_0$, we have an unbounded and feasible set of weights for the linear program to choose a_{j^*} in the definition of Algorithm 4. Therefore, a_{j^*} can be chosen by PF. □

4 Simulations

We ran the algorithms on data gathered from a power boost allocation problem. In this problem, n computer applications are each allocated a base level of power, and compete for $m < n$ additional (indivisible) units of extra power (*power boosts*) at each of T rounds (each application gets at most one boost per round). We obtain our instance from Apache Spark [27] benchmarks.

Table 1 lists the twelve Spark applications in our instance, each of which is defined by a fixed number of tasks. We profile tasks' completion time. We take the number of tasks

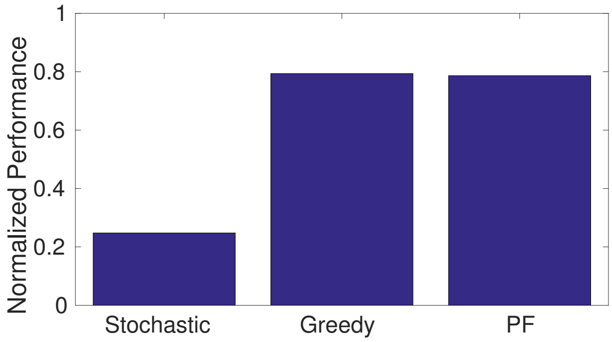

Figure 2: Nash Social Welfare achieved by the algorithms, normalized against OPT (which has performance 1).

completed in a round by an application as that application's utility. Thus, for each application a, we estimate the base and boosted power utility ($u_{a,t}^{\text{base}}$ and $u_{a,t}^{\text{boost}}$) in each round.

In our instance, there are two power boosts to be allocated. So at each round there are $\binom{12}{2}$ alternatives, one for each pair of applications. For an alternative j corresponding to power boosts for applications a and b, we have that $v_a^t(j) = u_{a,t}^{\text{boost}}$, $v_b^t(j) = u_{b,t}^{\text{boost}}$, and $v_c^t(j) = u_{c,t}^{\text{base}}$ for all other applications $c \neq a, b$. We have 497 rounds in the instance we tested.

We evaluate the performance of GREEDY and PF against the optimal offline solution, and also against an algorithm designed for online stochastic convex programming[4] [1] - a class of problems that includes the one we study. To our knowledge this algorithm is the state of the art for such problems in terms of theoretical guarantees. We refer to this algorithm as STOCHASTIC. The algorithm works by maximizing a weighted sum of valuations at each round, where the weights are updated at each round using online convex optimization techniques. The theoretical guarantees for STOCHASTIC are in expectation over instances where the order of the input matrices is *randomly permuted*. In the instance we test, however, the utilities are highly correlated over time. Applications that would benefit from a power boost in some round t are more likely to also benefit from a power boost in round $t+1$, because application phases may span multiple rounds. Due to this and other technical reasons, the theoretical guarantees do not apply here. The performance of the three algorithms is shown in Figure 2.

We see that STOCHASTIC performs relatively poorly, while GREEDY and PF each achieve about 80% of the performance of OPT. This motivates us to examine the difference in performance between GREEDY and PF for smaller values of T, as the difference between these two algorithms is most pronounced while a single decision has a relatively large effect.

To generate smaller instances, we sample starting rounds from the full set of 497 rounds. For each value of T in Figure 3, we randomly generate a starting round $t \in [1, 497 - T]$ and consider the T rounds starting at t, for 100 random choices of t. Our measure of performance is $NW(\boldsymbol{u}_{\boldsymbol{T}}^{\textbf{avg}})$, allowing for fair comparisons between different values of T.

We note that PF consistently performs slightly worse than GREEDY, which is consistent with the performance on the full instance. The difference is most pronounced on small values of T, since this is where the two algorithms differ the most. Performance increases with T, as we would expect, since more rounds allow the algorithms to choose more flexibly once all applications have positive accrued reward. However, the increase is not monotonic. One

[4]Of course, there are other online scheduling algorithms but they do not pursue Nash welfare as an objective.

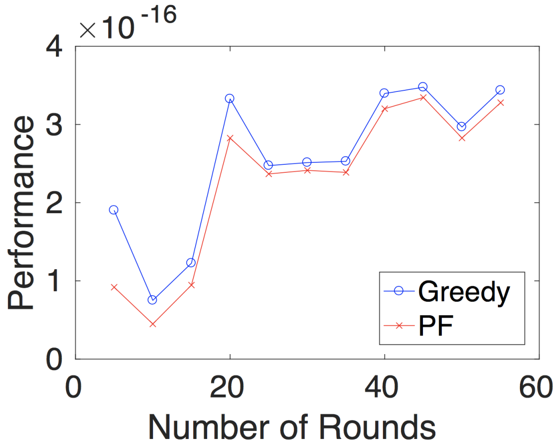

Figure 3: Nash Social Welfare achieved by GREEDY and PF as a function of the number of rounds.

explanation for this is because we throw away any choice of starting round t for which it is impossible to achieve $NW(\boldsymbol{u}_{\boldsymbol{T}}^{\mathbf{avg}}) > 0$ (it might be the case that for all T rounds, some application cannot receive positive utility). Since smaller values of T result in more choices of t being disqualified, there is a sense in which we are selecting for 'easier' instances for smaller values of T.

5 Conclusion

Election designers and social choice researchers often do not consider the fact that many elections do not occur in isolation, but rather are repeated over time. In this work, we have provided a framework to allow for the design and analysis of dynamic election protocols, and repeated decision making rules generally. We have presented two candidate online algorithms for solving these dynamic problems. Our simulations show that both algorithms perform well, but do not determine that either is clearly a better choice than the other. While GREEDY achieves slightly higher performance, PROPORTIONALFAIR has the advantage of being justified by the axiomatization given in this paper.

Our work leaves a lot of scope for future research. One direction would be to design a more precise model of voter preferences, possibly modeling changing preferences by an MDP [4, 21]. We have also not considered modeling discounting of the agents' utilities. Finally, there are many interesting questions regarding strategic behavior by the agents. In the most general setting, there appears to be no hope for a fair, strategy-proof rule due to the free-rider problem: agents are incentivized to under-report their utility for an alternative that gets chosen, and are thus indistinguishable from an agent that is genuinely unhappy with the chosen alternative. However, it may be possible to regain some (limited) strategy-proofness in a more restricted setting. For instance, what if we place restrictions on the utilities that can be reported, or restrict our attention to private goods?

Acknowledgments

We sincerely thank Songchun Fan for sharing the data used in Section 4, and Andrew Kephart for helpful discussions. We are thankful for support from NSF under awards IIS-1527434, IIS-0953756, CCF-1101659, CCF-1149252 (CA- REER), CCF-1337215 (XPS-CLCCA), SHF-1527610, and AF-1408784, ARO under grants W911NF-12-1-0550 and W911NF-11-1-0332, and a Guggenheim Fellowship. This work is also supported by STARnet, a Semiconductor Research Corporation program, sponsored by MARCO and DARPA. This work was done in part while Conitzer was visiting the Simons Institute for the Theory of Computing. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of these sponsors.

References

[1] Shipra Agrawal and Nikhil R Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1405–1424. SIAM, 2015.

[2] 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-15)*, 2015.

[3] Haris Aziz, Markus Brill, Vincent Conitzer, Edith Elkind, Rupert Freeman, and Toby Walsh. Justified representation in approval-based committee voting. *Social Choice and Welfare*, 48(2):461–485, 2017.

[4] Craig Boutilier and Ariel D Procaccia. A dynamic rationalization of distance rationalizability. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1278–1284, 2012.

[5] Steven J Brams, D Marc Kilgour, and M Remzi Sanver. A minimax procedure for electing committees. *Public Choice*, 132(3-4):401–420, 2007.

[6] Ioannis Caragiannis, Jason A. Covey, Michal Feldman, Christopher M. Homan, Christos Kaklamanis, Nikos Karanikolas, Ariel D. Procaccia, and Jeffrey S. Rosenschein. On the approximability of Dodgson and Young elections. *Artificial Intelligence*, 187–188:31–51, 2012.

[7] Ioannis Caragiannis, Christos Kaklamanis, Nikos Karanikolas, and Ariel D Procaccia. Socially desirable approximations for Dodgson's voting rule. *ACM Transactions on Algorithms*, 10(2): article 6, 2014.

[8] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum Nash welfare. In *Proceedings of the 17th ACM conference on Electronic Commerce*, pages 305–322, 2016.

[9] John R Chamberlin and Paul N Courant. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *American Political Science Review*, 77(03):718–733, 1983.

[10] Richard Cole and Vasilis Gkatzelis. Approximating the Nash social welfare with indivisible items. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 371–380. ACM, 2015.

[11] Vincent Conitzer, Rupert Freeman, and Nisarg Shah. Fair public decision making. In *Proceedings of the 18th ACM Conference on Electronic Commerce*, pages 629-646, 2017. Forthcoming.

[12] Andreas Darmann and Joachim Schauer. Maximizing Nash product social welfare in allocating indivisible goods. *European Journal of Operational Research*, 247(2):548–559, 2015.

[13] Edith Elkind, Piotr Faliszewski, Piotr Skowron, and Arkadii Slinko. Properties of multiwinner voting rules. In *Proceedings of the Thirteenth International Conference on Autonomous Agents and Multiagent Systems*, pages 53–60, 2014.

[14] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. Dominant resource fairness: Fair allocation of multiple resource types. *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, pages 24–24, 2011.

[15] Mingyu Guo, Vincent Conitzer, and Daniel M. Reeves. Competitive repeated allocation without payments. In *Proceedings of the Fifth Workshop on Internet and Network Economics (WINE)*, pages 244–255, Rome, Italy, 2009.

[16] A. Jalali, R. Padovani, and R. Pankaj. Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system. In *Proceedings of the IEEE Vehicular Technology Conference*, pages 1854–1858. IEEE, 2000.

[17] Ian Kash, Ariel D Procaccia, and Nisarg Shah. No agent left behind: Dynamic fair division of multiple resources. *Journal of Artificial Intelligence Research*, pages 579–603, 2014.

[18] Burt L Monroe. Fully proportional representation. *American Political Science Review*, 89(04):925–940, 1995.

[19] John Nash. The bargaining problem. *Econometrica*, 18(2):155–162, 1950.

[20] George L Nemhauser and Laurence A Wolsey. Integer and combinatorial optimization. interscience series in discrete mathematics and optimization. *ed: John Wiley & Sons*, 1988.

[21] David C Parkes and Ariel D Procaccia. Dynamic social choice with evolving preferences. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pages 767–773, 2013.

[22] David C Parkes, Ruggiero Cavallo, Florin Constantin, and Satinder Singh. Dynamic incentive mechanisms. *AI Magazine*, 31(4):79–94, 2010.

[23] David C Parkes, Ariel D Procaccia, and Nisarg Shah. Beyond dominant resource fairness: Extensions, limitations, and indivisibilities. In *ACM Conference on Electronic Commerce*, pages 808–825. ACM, 2012.

[24] Ariel D Procaccia. Cake cutting: not just child's play. *Communications of the ACM*, 56(7):78–87, 2013.

[25] Sara Ramezani and Ulle Endriss. Nash social welfare in multiagent resource allocation. In *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*, pages 117–131. Springer, 2010.

[26] Pramod Viswanath, David NC Tse, and Rajiv Laroia. Opportunistic beamforming using dumb antennas. *IEEE Transactions on Information Theory*, 48(6):1277–1294, 2002.

[27] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. In *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing*, volume 10, page 10, 2010.