Fair and Efficient Social Choice in Dynamic Settings

Rupert Freeman and Seyed Majid Zahedi and Vincent Conitzer
Department of Computer Science
Duke University
Durham, NC 27708, USA
{rupert,zahedi,conitzer}@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 [Procaccia, 2013] as well as other models [Ghodsi et al., 2011; Parkes et al., 2012]. 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 [Chamberlin and Courant, 1983; Monroe, 1995; Brams et al., 2007; Aziz et al., 2017].

A classical approach to obtaining a fair outcome in a context where agents have utility functions is to maximize the Nash welfare [Nash, 1950], 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 [Parkes and Procaccia, 2013; Kash et al., 2014; Aleksandrov et al., 2015].

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 [Cole and Gkatzelis, 2015; Darmann and Schauer, 2015; Ramezani and Endriss, 2010]. 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 \( \frac{1}{m} \) 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 [Nash, 1950] 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 \( v^t_i(a_j) \in \mathbb{N} \) for every alternative \( a_j \).\(^2\) Thus the input at every round is a matrix \( V^t = (v^t_i(a_j))_{ij} \). Let \( v^t_i(a_j) \) 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 \( t \), \( u_t \), 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 \( t \), \( u_t(i) = \sum_{t'=1}^{t} v^t_i(c_{t'}) \). 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 \( u_{t-1}(i) = 0 \) when \( t \) is round, \( C_t \). The average utility of the agents over the first \( T \) rounds is \( u^{avg}_T = \frac{1}{T} u_T \).

A DSCF is anonymous if applying permutation \( \sigma \) 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 \( \sigma \) 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 \( u \), \( NW(u) \), is defined to be the product of the agents’ utilities, \( NW(u) = \prod_{i=1}^{n} u(i) \). We also define \( NW^+(u) = \prod_{i=0}^{n} u(i) \) to be the product of all positive entries of \( u \). Our aim is to maximize the NW of the average utility across all \( T \) rounds, \( NW\left(u^{avg}_T\right) \). 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 \( \{V^t\}_{t=1}^{T} \), 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 \( \epsilon \). Numbers involving \( \epsilon \) take the form \( \sum_{t=-\infty}^{\infty} a_t \epsilon^t \).\(^3\) For two such numbers \( a = \sum_{t=-\infty}^{\infty} a_t \epsilon^t \) and \( b = \sum_{t=-\infty}^{\infty} b_t \epsilon^t \), 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 \( \epsilon \). Two numbers are equal if all coefficients are equal.

Many proofs, as well as additional simulations, are omitted due to space constraints. The details can be found in the full version of the paper, which appears on the authors’ websites.

### 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

---

\(^1\)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 \( \epsilon \), in this paper we utilize only powers of \( \epsilon \) between \( \epsilon^{-1} \) and \( \epsilon^6 \).
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, [Caragiannis et al., 2012; Elkind et al., 2014]. The first algorithm, GREEDY, simply chooses $c_t$ to maximize $NW(u_t^{\text{avg}})$, 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 [Viswanath et al., 2000], 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 $w$) equal to the inverse of her accrued reward at the start of each round and chooses $C_t = \arg\max_{a_j \in A} w \cdot v^t(a_j)$, 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 $V^t = (\frac{3}{2}, \frac{1}{3})$. 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 $a_2$.

Under PF, each agent is given weight inversely proportional to their own accrued utility. That is, agent 1 has weight $1/3$ and agent 2 has weight $1/2$. 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 $NW(u_t^{\text{avg}}) = 0$ 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)**

1. Input $u_{t-1}$
2. for $i = 1, \ldots, n$ do
3. Randomly choose $0 < x_i \leq 1$
4. end for
5. Return $c_t \in \arg\max_{a_j \in A} \prod_{i=1}^n \max\{u_{t-1}(i) + v^t_i(a_j), x_i\}$

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)**

1. Input $u_{t-1}$
2. $C_t \leftarrow \arg\max_{a_j \in A} |\{i : u_{t-1}(i) + v^t_i(a_j) > 0\}|$
3. for $j \in C_t$ do
4. if $\exists j'$ such that
5. $\{i : u_{t-1}(i) + v^t_i(a_j) > 0\} = \{i : u_{t-1}(i) + v^t_i(a_{j'}) > 0\}$ and $NW^+(u_{t-1} + v^t(a_j)) < NW^+(u_{t-1} + v^t(a_{j'}))$
6. then
7. $C_t \leftarrow C_t \setminus \{a_j\}$
8. end if
9. end for
10. 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
Algorithm 3 PROPORTIONAL FAIR (select one alternative)
1: Input $u_{t-1}$
2: for $i \in I_0$ do
3: Randomly choose $0 < x_i \leq 1$
4: Randomly choose $y_i \in \mathbb{R}$
5: end for
6: $w_i = \begin{cases} \frac{1}{u_{t-1}(i)} + k_i, & \text{if } u_{t-1}(i) = 0 \\ x_i^{1/2} + y_i, & \text{if } u_{t-1}(i) > 0 \end{cases}$
7: Return $c_t \in \arg\max_{a_j \in A} w \cdot v^t(a_j)$

Algorithm 4 PROPORTIONAL FAIR (select all alternatives)
1: Input $u_{t-1}$
2: $C_t \leftarrow \emptyset$
3: for $j = 1, \ldots, m$ do
4: if the following linear program is unbounded then
5: $C_t \leftarrow C_t \cup \{a_j\}$
6: end if
7: end for
8: Return $C_t$

even when it was possible to give positive utility to all agents.

Example 2. Let $n = 2$, $m = 3$, and $t = 1$. Suppose that $V_1 = (\frac{3}{2}, \frac{3}{2})$. 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 $\arg\max_{j \in \{1,2,3\}} w \cdot v^t(a_j)$. 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 nondeterministic 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 $\epsilon$ terms appearing in the product is minimized.

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 $\{x_i\} \in I_0$ and $\{y_i\} \in I_0$. For all $i \notin I_0$, set $w_i = \frac{1}{u_{t-1}(i)}$, and for all $i \in I_0$, set $w_i = \frac{1}{u_{t-1}(i)} + k_i$ 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 $\delta$.

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

$$\sum_{i \in I_0} x_i v^t(c_t) \geq \sum_{i \in I_0} x_i v^t(a_j). \quad (1)$$

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

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

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 $\epsilon^0$ in the dot product from Line 7 of Algorithm 3. This tells us that

$$\sum_{i \in I_0} y_i v^t(c_t) + \sum_{i \in I_0} \frac{1}{u_{t-1}(i)} v^t(c_t) \geq \sum_{i \in I_0} y_i v^t(a_j) + \sum_{i \in I_0} \frac{1}{u_{t-1}(i)} v^t(a_j). \quad (2)$$

Dividing Inequality 1 by $\delta$ and adding Inequality 2 gives

$$\sum_{i = 1}^n w_i v^t(c_t) \geq \sum_{i = 1}^n w_i v^t(a_j),$$

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 $p$ and $q \neq 0$ for the values of the variables in the LP such that $p + kq$ is feasible for all $k > 0$ and $q$ has positive objective value (this is a known fact about linear programs with unbounded value; see, e.g., [Nemhauser and Wolsey, 1988], Theorem 4.7). We use these to exhibit values of $\{x_i\} \in I_0$ and $\{y_i\} \in I_0$ 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,

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

4583
for all $k > 0$. In particular, this implies that it cannot be the case that $\sum_{i \in I_0} q_i v_i^k(c_i) < \sum_{i \in I_0} q_i v_i^k(a_j)$, or else Inequality 3 would be violated for large enough values of $k$. There are two possibilities.

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

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

$$\sum_{i \in I_0} p_i v_i^k(c_i) + \sum_{i \in I_0} \frac{v_i^k(c_i)}{u_i(i)} \geq \sum_{i \in I_0} p_i v_i^k(a_j) + \sum_{i \in I_0} \frac{v_i^k(a_j)}{u_i(i)}.$$ 

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 $c_i$ in Line 7 of Algorithm 3. Since the coefficient is at least as large for $c_i$ as for $a_j$, and the $c_i$ coefficients are equal (and there are no further non-zero terms), $c_i$ 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-freeness (SF) if $C_t$ is unchanged (for the same choice of tiebreaking in earlier rounds) if we replace $v_i^k(a_j)$ by $k \cdot v_i^k(a_j)$ for all $a_j \in A$ for every $t = 1, \ldots, T$.

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 $v_i^k(a_j) \in V^t$, create $v_i^k(a_j)$ agents each with $u_{t-1}(x) = u$, $v_i^k(a_j) = 1$, and $v_i^k(a_j') = 0$ for all $j' \neq j$.

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.

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 $z_1, \ldots, z_n$ 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 time $t$ if $\sum_j z_j v_i^k(a_j) \geq v_i^k(a_j)$ 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 $a_j$.

**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-dominates $a_j$ (including $a_j$ itself).

**Lemma 1.** PF satisfies SF, SSMA, P, and NZDA.

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. In the case that all agents have $u_{t-1}(i) > 0$, we use the fact that $f$ satisfies SF, SSMA, and P to show that $f$’s choice of alternative is the same as one chosen by PF. First we use SF to scale the agents’ accrued utilities to be the same (since they are all non-zero), then use SSMA to replace each agent with several new agents which have unit vector valuations, and then use plurality to choose...
there exist values for \( f \) that 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 \( V^f \) 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 \( V^f \). 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 \( V^f \) 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.

Minimize

\[
H \sum_{i \in I_0} \sum_{a_j \in A} (v_i(a_j) - v_i(a_j))z_j + \sum_{i \notin I_0} \sum_{a_j \in A} \frac{1}{u_{t-1}(i)}(v_i(a_j) - v_i(a_j))z_j \quad (4)
\]

subject to

\[
\sum_{a_j \in A} v_i(a_j)z_j \geq v_i(a_j) \quad \forall i \in I_0
\]

\[
\sum_{a_j \in A} z_j = 1, \quad z_j \geq 0 \quad \forall j
\]

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 \( V^f \). Conversely, if the optimal value of the objective is negative then either 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 optimal value of LP (4) 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 \( z_j \):

Maximize

\[
\sum_{i \in I_0} v_i(a_j)w_i - r
\]

subject to

\[
\sum_{i \in I_0} v_i(a_j)w_i - r \leq H \sum_{i \in I_0} (v_i(a_j) - v_i(a_j)) + \sum_{i \notin I_0} \frac{v_i(a_j) - v_i(a_j)}{u_{t-1}(i)} \quad \forall j \in \{1, \ldots, m\}
\]

\[
w_i \geq 0 \quad \forall i \in I_0
\]

Let \( \tau = \sum_{i \in I_0} v_i(a_j)w_i - r \) denote the objective. The first set of constraints can now be rewritten as

\[
\tau + \sum_{i \in I_0} (w_i + H)v_i(a_j) + \sum_{i \notin I_0} \frac{1}{u_{t-1}(i)}v_i(a_j)
\]

\[
\leq \sum_{i \in I_0} (w_i + H)v_i(a_j) + \sum_{i \notin I_0} \frac{1}{u_{t-1}(i)}v_i(a_j).
\]

Since \( a_j \) is not 0-dominated, the optimal value of LP (4) 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 \( i \notin I_0 \), 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 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 [Zaharia et al., 2010] 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 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 (\( u_{a, base} \) and \( u_{a, boost} \)) 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 \( v_i(j) = v_{a, boost}^i, v_i(j) = u_{b, boost}^i, \) and \( v_i(j) = u_{c, base}^i \) for all other applications \( c \neq a, b \). We have 497 rounds in the instance we tested.
Our measure of performance is $NW(u_{T}^{\text{avg}})$, 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 explanation for this is because we throw away any choice of starting round $t$ for which it is impossible to achieve $NW(u_{t}^{\text{avg}}) > 0$ (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 [Boutilier and Procaccia, 2012; Parkes and Procaccia, 2013]. 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.

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


