Greedily Maximizing Ex-Ante Fairness

Ruben Becker[1]*, Bojana Kodric[2]*, Cosimo Vinci[3]*

[1]Department of Environmental Sciences, Informatics and Statistics, Ca' Foscari University of Venice, Italy.
[2]Ericsson Nikola Tesla, Zagreb, Croatia.
[3]Department of Mathematics and Physics "Ennio De Giorgi", University of Salento, Lecce, Italy.
rubensimon.becker@unive.it, bojana.kodric@ericssonnikolatesla.com, cosimo.vinci@unisalento.it.

Abstract

We study a general framework of optimization with the aim to compute fair solutions in settings with a set of agents whose valuations are combined using an aggregation function. The strength of our framework lies (1) in its generality and (2) in the fact that we leverage the power of ex-ante fairness, a concept that has recently gained much attention in the scope of fair allocation and fairness in AI in general. More precisely, in our setting there are n set functions $f_1, \dots, f_n$ (e.g., the valuation functions of n agents) that are combined using an aggregation function g (e.g., the minimum, Nash social welfare, p-norm). The power of ex-ante fairness is obtained by allowing as a feasible solution not simply a finite set S, but instead a distribution Π over feasible sets. The goal in our setting is then to find a probability distribution $p \in \Pi$ that maximizes the value resulting from aggregating (using g) the n expected values of the functions $f_1, \dots, f_n$ obtained when sampling a set S according to the distribution $\mathbf{p}$. We stress that this is different from maximizing the expected value of g (ex-post fairness) and typically allows for much fairer solutions. We give three different greedy algorithms for three different settings of this framework and prove that they achieve constant approximation guarantees under certain realistic assumptions. For some of the settings, we show that these approximation guarantees are tight. Specific scenarios that can be modelled using our framework include fair information diffusion in social networks, fair submodular matching problems and ex-ante versions of item assignment problems.

1 Introduction

In this work, we consider a general multi-agent setting where the satisfaction of each agent is measured by an individual utility. A central authority must make choices such as to improve the overall utility of the various agents involved. This overall utility is measured by a function that aggregates the individual utilities, resulting in an aggregated value that can be used as an indicator of *fairness* and/or *social welfare*.

A well-studied example of such a multi-agent setting are information dissemination dynamics in social networks. Consider a social network where nodes represent individuals and edges represent their social interactions and assume that a government agency wants to disseminate information on certain nodes in the network to raise awareness about a specific issue with social impact (e.g., health (Valente and Pumpuang 2007; Wilder et al. 2018; Yadav et al. 2018; Amoruso et al. 2020), finance (Banerjee et al. 2013), etc.) on which people have limited knowledge or are often misled, e.g., due to fake news. Given the complexity and vastness of social networks, it may be impractical to target every node with a tailored awareness campaign. A more feasible strategy is to carefully select a limited number of nodes to target so that, due to their centrality in the network and the word-of-mouth effect, the message reaches as many nodes as possible. The underlying algorithmic problem has been studied in the artificial intelligence and network analysis communities under the term *influence maximization*. Here the goal is to select a limited number of *seed nodes* in the network such as to maximize the expected number of nodes that are reached by the campaign. Clearly however, when pursuing the goal of maximizing the expected number of influenced nodes, i.e., the sum of the probabilities of influencing each node, there may be nodes not covered by the campaign at all. While this may not only be perceived as unfair, over time uninfluenced nodes may become channels for misinformation and disinformation again and thus have a negative global impact. A more reasonable and equitable choice would be to consider alternative aggregation functions instead of simply summing the node's probabilities. For example, one could use the minimum instead (as proposed by Fish et al. (2019)) or other social metrics with better fairness guarantees.

A General Framework for Ex-Ante Maximization. To model the above scenario, and many others, we introduce and study the following general framework. We start from an optimization problem whose feasible solutions are subsets S of a universe U of m elements. Calling $\mathcal{S}$ the set of all such feasible sets, we aim to maximize an objective function that results by aggregating the values $f_1(S), \dots, f_n(S)$ of n monotone set functions $\mathbf{f} = (f_1, \dots, f_n)$ with $f_i : 2^U \to \mathbb{R}_{\geq 0}$. This aggregation is done using another function $g : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ that we assume to satisfy $g(0) = 0$, non-decreasing in every component and concave. The probably most straightforward examples of the aggregation function g are the sum or 1-norm, i.e., $g(\mathbf{x}) = \|\mathbf{x}\|_1$, and the minimum function, i.e., $g(\mathbf{x}) = \min_{i=1,\dots,n} x_i$. Other examples include the p-norms for $p \in (0, 1)$ and the Nash Social Wel-

*These authors contributed equally.

fare (Nash 1950) (i.e., the p-norm for $p = 0$). For a general function g, a natural discrete optimization problem to study is then

$$\max\{g(\mathbf{f}(S)) : S \in \mathcal{S}\}, \tag{1}$$

i.e., we want to find the feasible set that maximizes the aggregation of the n function values.

In the previously described example of influence maximization, U is the set of nodes within the social network and, given $S \subseteq U$ and $i \in [n]$,[1] the value $f_i(S)$ represents the probability that node i will be positively influenced (through the word-of-mouth effect) by nodes in S, assuming that the campaign has been initiated from the nodes in S. The standard influence maximization problem (as it was introduced by Kempe, Kleinberg and Tardos (2015)) is obtained, when g is the sum (or 1-norm), the maximin fair influence maximization problem (see, e.g., the work by Fish et al. (2019)) is obtained when g is the minimum.

More generally, let us illustrate better why *fairness* can be a motivation for studying the general setting: Assume that there are n agents and the functions $f_1, \ldots, f_n$ indicate their valuation for each of the feasible subsets $S \in \mathcal{S}$ of the universe U. For g being the minimum, for example, the problem then aims at finding the feasible subset S that maximizes the minimum valuation of all agents. This is known as the maximin-criterion of fairness. Other fairness notions can be modeled using other functions g, e.g., p-norms with $p \geq 0$. We note that once we assume a continuous welfare function to satisfy four arguably natural properties, namely monotonicity, symmetry, independence of unconcerned agents and common scale, it follows using the Debreu-Gorman Theorem (Debreu 1959; Gorman 1968) that the social welfare function is equivalent to a p-norm (Heidari et al. 2018). Hence, our framework can be argued to cover most interesting cases of welfare functions.

It is evident that the problem in (1) is computational intractable without assuming additional properties on the set functions $\mathbf{f}$. In this work, in the most settings that we study, we assume the functions $\mathbf{f} = (f_1, \ldots, f_n)$ to be submodular, i.e., to satisfy the diminishing returns property. In addition, we study one setting where we make a more indirect assumption on the set functions, namely we assume that a certain oracle problem involving the set functions is approximately solvable. We get back to this below, when formally describing the three studied scenarios. Even when assuming submodularity, the optimization problem in (1) is very hard to solve. A reduction from the NP-hard maximum coverage problem for example directly implies that this problem is not approximable within any finite ratio, unless P = NP. But, not only is the presented problem difficult to study from a computational perspective, it also does not admit very good solutions in the sense of fairness. Consider for example the setting where $n = 2$, $U = \{u_1, u_2\}$, $\mathcal{S} = \{\{u_1\}, \{u_2\}\}$, $f_1(\{u_1\}) = f_2(\{u_2\}) = 1$, $f_1(\{u_2\}) = f_2(\{u_1\}) = 0$, and g is the minimum function. Then clearly the optimal solution to the above problem is 0, that is, there is no feasible solution that achieves any level of fairness.

[1]For a natural number $k \in \mathbb{N}$, we let $[k] := \{1, \ldots, k\}$.

This problem has been addressed in the literature by considering the *ex-ante fairness* notion. The idea is to consider randomized solutions, i.e., distributions over the feasible solutions in $\mathcal{S}$ along with expected values. The key of ex-ante fairness, however, is to consider g applied to the expected values of the functions $f_1(S), \ldots f_n(S)$ when S is sampled according to the distribution. This is in contrast to ex-post fairness, which is obtained when considering the expected value of g applied to the values $f_1(S), \ldots f_n(S)$. Recall the example above, where $n = 2$, $U = \{u_1, u_2\}$, $\mathcal{S} = \{\{u_1\}, \{u_2\}\}$ and g as the minimum function. The optimal ex-ante value over distributions is now $1/2$ (achieved by the uniform distribution that chooses each $S \in \mathcal{S}$ with probability $1/2$) rather than 0 for deterministic as well as for ex-post fairness. Hence, considering the ex-ante objective over distributions allowed us to obtain much fairer solutions. This example goes back to Machina (1989), who illustrated the problem with a parent assigning an indivisible good to one of two children. Conceptually, ex-ante fairness can be understood as fairness as principle of *equal opportunities* that guarantees fairness in the long run in settings of repeated decisions. Formally, we let Π be a set of distributions over the feasible sets $\mathcal{S}$ and consider the optimization problem

$$\max\{g(\mathbb{E}_{S\sim\mathbf{p}}[\mathbf{f}(S)]) : \mathbf{p} \in \Pi\}, \tag{2}$$

where $S \sim \mathbf{p}$ denotes the random experiment of sampling a set $S \in \mathcal{S}$ according to the distribution $\mathbf{p} \in \Pi$.

In the case where the functions $\mathbf{f} = (f_1, \ldots, f_n)$ are additive one could be tempted to solve problem (2) using techniques from convex optimization. The problem with this approach however is that the dimension of $\mathcal{S}$ may be exponential in m and, in general, an optimal solution to problem (2) may have exponential support. Indeed, we allow that the solutions are not given explicitly (which would be problematic in the case of exponential support), but instead can be given as an oracle, i.e., outputting a solution $\mathbf{p}$ can also mean to provide an algorithm that when repeatedly queried for feasible sets from $\mathcal{S}$, outputs them following the distribution $\mathbf{p}$.

Remark 1.1. *Our probabilistic problems* (2) *are* relaxations *of the original discrete problems* (1)*, that is, a feasible deterministic solution $S \in \mathcal{S}$ is a feasible solution to the ex-ante version of the problem through the distribution* $\mathbf{p}$ *with* $\mathbf{p}_S = 1$ *and* $\mathbf{p}_T = 0$ *for all* $T \in \mathcal{S} \setminus \{S\}$*. It is important to note that from an approximation algorithms perspective, the problems are incomparable as optimal solutions to* (2) *are not in general "deterministic". In particular, although the discrete problem* (1) *seems much harder, a corresponding approximation algorithm does not imply the same approximation for the corresponding ex-ante problem.*

2 Settings, Contribution and Techniques

In this work we consider three different instances of the general problem in (2) that result from restrictions to the feasible set of the optimization problem that are consequences of restraining the feasible sets $\mathcal{S}$. In some of the settings considered we in addition assume the functions $\mathbf{f} = (f_1, \ldots, f_n)$ to be submodular. In total we study three different settings.

We next detail these three settings and our contribution for each of them. We summarize our results in Table 1.

Setting 1. The first setting is obtained when restricting the set of feasible solutions to the sets of cardinality at most k, i.e., $\mathcal{S} := \{S \subseteq U : |S| \leq k\}$ and then letting the set of distributions be $\Pi := \{\mathbf{p} \in [0,1]^{\mathcal{S}} : \|\mathbf{p}\|_1 = 1\}$. In addition, we assume $\mathbf{f} = (f_1, \dots, f_n)$ to be n submodular functions and g to be Lipschitz continuous with a Lipschitz constant that is polynomially bounded in n and m.

In Section 3, we present an algorithm for this setting that we call *probabilistic greedy algorithm*. For this algorithm, we assume access to an oracle that, for every arbitrarily small $\delta > 0$, computes an additive δ-approximate solution to the concave optimization problem

$$\max\Big\{g\Big(\sum\nolimits_{j\in U} x_j \cdot \mathbf{m}_j\Big) : \|\mathbf{x}\| = 1, \mathbf{x} \geq \mathbf{0}\Big\}, \quad (3)$$

where, for every $j \in U$, $\mathbf{m}_j \in \mathbb{R}^n_{\geq 0}$ is a non-negative vector. We note that this is a comparatively mild assumption that, for many natural choices of g, follows from the theory of self-concordant functions by Nesterov and Nemirovski (1994). Under this assumption, our probabilistic greedy algorithm achieves a multiplicative $1 - 1/e - \varepsilon$-approximation in polynomial time and is thus optimal up to an arbitrarily small error. Our algorithm can be seen as a probabilistic version of the classical greedy algorithm for maximizing a submodular set function subject to a cardinality constraint (Nemhauser, Wolsey, and Fisher 1978). This classical algorithm, for k steps, adds the element of maximum marginal increment to an initially empty set that is output at the end. Our algorithm, instead computes a distribution over subsets that can be represented by k non-negative vectors $\mathbf{x}^1, \dots, \mathbf{x}^k$ each of one-norm 1. The set S of size at most k is sampled from these vectors by independently sampling k elements, the t-th element being distributed according to the vector $\mathbf{x}^t$. Our algorithm in its t-th iteration computes a vector $\mathbf{x}^t$ that maximizes the expected increment to the random set of size at most $t-1$ obtained by independently sampling from the vectors $\mathbf{x}^1, \dots, \mathbf{x}^{t-1}$. We show that (after a straightforward application of a concentration bound), this problem can be solved using an oracle for a concave optimization problem of the form stated above. We stress that our algorithm outputs a distribution that samples at most k elements independently, but our proofs show that it actually achieves an approximation factor of $1 - 1/e - \varepsilon$ with respect to an optimal distribution that is not restricted to sample elements independently. Furthermore, our approximation ratio even holds with respect to the larger set of feasible solutions, where distributions are not restricted to be over sets of size at most k, but have to satisfy a cardinality constraint only in expectation.

One specific problem that can be modeled using this setting and can thus be solved using our algorithm is the ex-ante fair influence maximization problem that was considered by Becker et al. (2022) and is basically the ex-ante version of the maximin fair influence maximization problem mentioned previously and considered by Fish et al. (2019). A similar setting was studied by Becker et al. (2023). Our algorithm achieves essentially the same approximation ratio as the one from (Becker et al. 2022).[2] In this case, where g is the minimum, the concave optimization problem actually simplifies to a linear one.

To complement the above results, we show that our algorithm achieves the best possible polynomial-time approximation. Indeed, in the full version of this work we give a simple reduction from the max-coverage problem that shows that the considered problem cannot be approximated within a factor better than $1 - 1/e + \varepsilon$ for any $\varepsilon > 0$, unless $\mathrm{RP} = \mathrm{NP}$, where RP is the class of problems with 1-sided error polynomial-time Monte Carlo algorithms.

Setting 2. The second setting allows an arbitrary set of feasible solutions, i.e., $\mathcal{S}$ can be any subset of 2^U, while the set of distributions is again $\Pi := \{\mathbf{p} \in [0,1]^{\mathcal{S}} : \|\mathbf{p}\|_1 = 1\}$. Problems from this setting are a superset of problems from Setting 1 and thus the problem again cannot be approximated within a factor better than $1 - 1/e + \varepsilon$, even if the functions $\mathbf{f}$ are submodular. We give an approximation algorithm also for this setting. Again we assume to have access to an oracle for a specific optimization problem. The oracle problem in this case however takes the form

$$\max\Big\{\sum\nolimits_{i=1}^{n} \alpha_i \cdot f_i(S) : S \in \mathcal{S}\Big\}, \quad (4)$$

where $\boldsymbol{\alpha} \geq \mathbf{0}$ is a non-negative weight vector, and we assume to have a multiplicative approximation algorithm. In the common case where the functions $\mathbf{f}$ are submodular, also the objective function of (4) is submodular and hence multiplicative approximation algorithms exist for several choices of the feasible set $\mathcal{S}$ (e.g., matroid constraints).

Our algorithm is inspired by the continuous greedy algorithm for submodular optimization under matroid constraints (Vondrák 2008; Călinescu et al. 2011). Its analysis for our modified setting however requires further non-trivial arguments. We call our algorithm *continuous greedy* as well. The algorithm achieves a $1 - 1/e^{\gamma} - \varepsilon$-approximation, when equipped with a γ-approximate oracle for the problem in (4). In cases where the oracle problem can be solved optimally, also this algorithm achieves an approximation ratio that is optimal up to an arbitrarily small error. Our continuous greedy algorithm is a discretization of a continuous process that moves a vector $\mathbf{p}$ from $\mathbf{p}(0) = \mathbf{0}$ towards a solution $\mathbf{p}(1)$ while moving into the direction $\mathbb{1}_{S(t)}$, where $S(t) \in \mathcal{S}$ is the γ-approximate solution to the oracle problem for a suitable choice of the weight vector α that is defined through the gradient of g at the current solution $\mathbf{p}(t)$.

An application of this scenario is an ex-ante version of the well-studied *maximum submodular matching (MSM)* problem. In this problem we are given a graph $G = (V, E)$ together with a submodular set function $f : 2^E \to \mathbb{R}_{\geq 0}$ on the edges and the goal is to find a matching $M \subseteq E$ in G (i.e., set of disjoint edges) that maximizes $f(M)$. In the more general *maximum submodular b-matching problem (MSbM)*, a vector $b \in \mathbb{N}^v$ is given in addition, and every node v should

[2] The algorithm of Becker et al. (2022) incurs an additonal additive error due to the approximation of the information spread. Also when using our algorithm for this scenario, the functions f_i cannot be computed exactly and instead have to be approximated.

Setting	Algorithm	Approx. Factor	Hardness	Example
Setting 1	probabilistic greedy	$1 - \frac{1}{e} - \varepsilon$	$1 - \frac{1}{e} + \varepsilon$	Fair Influence Maximization
Setting 2	continuous greedy	$1 - \frac{1}{e^{\gamma}} - \varepsilon$	$1 - \frac{1}{e} + \varepsilon$	Submodular b-matching
Setting 3	probabilistic greedy assignment	$\frac{1}{2} - \varepsilon$	$1 - \frac{1}{e} + \varepsilon$	Submodular Nash social welfare

Table 1: Summary of our results: We give three different greedy algorithms for three different settings. Hardness results hold under the assumption $\mathrm{RP} \neq \mathrm{NP}$. The listed examples are each ex-ante versions of the respective problems.

be adjacent to at most b_v edges in M. Several approximation algorithms are known for these problems, the most classical being that the greedy algorithm is a 3-approximation if f is monotone (Fisher, Nemhauser, and Wolsey 1978). Several works consider problems related to fairness or diversity in this context of submodular matching, see, e.g., the work of Ahmed et al. (2017). Our framework can be used directly to model a setting with n agents with different preferences (i.e., set functions f_i) over matchings or b-matchings. In this case the oracle problem for our algorithm is the standard MSM problem with monotone set functions. The currently best result for this problem is $1/2 - \varepsilon$ for an arbitrarily small $\varepsilon > 0$ (Feldman et al. 2011). Using this result in our framework gives a $1 - e^{-1/2} - \varepsilon$-approximation for the ex-ante version of the maximum submodular b-matching problem.

Setting 3. In the third setting, the universe U is the cross-product of $[n]$ and a set of M *items* $J := [M]$, i.e., $U := [n] \times J$ and thus $m = n \cdot M$. The feasible sets are $\mathcal{S} := \{S \subseteq U \ : \ \forall r \in J : \exists! i \in [n] : (i, r) \in S\}$, i.e., we restrict to sets that correspond to *assignments* of the set J to the set $[n]$. As before $\Pi := \{\mathbf{p} \in [0,1]^{\mathcal{S}} : \|\mathbf{p}\|_1 = 1\}$ and we again assume each f_i to be submodular.

In Section 3, we then present the *probabilistic greedy assignment algorithm*. Here, we assume to have access to an oracle that, for every $\delta > 0$, computes an additive δ-approximate solution to the concave optimization problem

$$\max \Big\{ g\Big(\sum\nolimits_{i \in [n]} x_i \cdot \mathbf{m}_i \Big) : \|\mathbf{x}\| = 1, \mathbf{x} \geq \mathbf{0} \Big\}, \quad (5)$$

where, for every $i \in [n]$, $\mathbf{m}_i \in \mathbb{R}^n_{\geq 0}$ is a non-negative vector. This oracle problem is similar to that of Setting 1; the only difference is that the convex combination is taken over the n agents rather than over the m items. Consequently, the algorithms for the corresponding oracle problem are identical to those used in Setting 1. The probabilistic greedy assignment algorithm is also similar to the probabilistic greedy algorithm for Setting 1. However, instead of computing distributions over the m items in k iterations, we now compute distributions over the n agents in M iterations, i.e., in the j-th iteration, we greedily compute a distribution that characterizes to which agent the j-th item is assigned.

Specific problems that can be modeled using this setting are ex-ante variants of all classical item assignment problems with indivisible goods and submodular valuation functions. Maybe most prominently, our framework applies to item assignment problems where the welfare function, i.e., the aggregation function g is the Nash social welfare function (i.e., the geometric mean). Here, the concavity of the geometric mean allows us to apply our framework. These item assignment problems have recently received a lot of attention (Nguyen and Rothe 2014; Cole et al. 2017; Cole and Gkatzelis 2018; Caragiannis et al. 2019; Amanatidis et al. 2021; Akrami et al. 2022; Banerjee et al. 2022; Bilò et al. 2022). Perhaps most interestingly, for the ex-ante version of the Nash social welfare problem with submodular valuation functions, our framework yields a $1/2 - \varepsilon$-approximation. Furthermore, in the full version we show via a reduction from the submodular welfare problem that, unless $\mathrm{RP} = \mathrm{NP}$, this problem cannot be approximated within $1 - 1/e + \varepsilon$ for any $\varepsilon > 0$. The deterministic versions of the considered problem and similar variants have recently been studied among others by (Li and Vondrák 2021a,b; Garg et al. 2023; Garg, Kulkarni, and Kulkarni 2023; Dobzinski et al. 2024).

Some Preliminaries. We use bold fonts for vectors (of functions). For a set $S \subseteq U$, χ_S is the indicator function, i.e., $\chi_S(T) = 1$ if $S = T$ and $\chi_S(T) = 0$ otherwise. With $\mathbb{1}_S$ we indicate the unit vector indexed by sets, with entry 1 at position S and 0 elsewhere. For two vectors $\mathbf{a}$ and $\mathbf{b}$, we let $\mathbf{a} \vee \mathbf{b}$ be the vector with $(a \vee b)_i = \max\{a_i, b_i\}$. A set function $f : 2^U \to \mathbb{R}$ is *submodular* if for all $S \subseteq T \subseteq U$ and $e \in U \setminus T$, $f(S \cup \{e\}) - f(S) \geq f(T \cup \{e\}) - f(T)$. We use $S \cup e$ instead of $S \cup \{e\}$ if it is clear from the context that e is an element rather than a set.

Assumptions on the Input. We make the most basic assumption regarding the set functions $\mathbf{f}$, namely we assume them to be given in form of a value oracle (Vondrák 2008), that is, for a set $S \subseteq U$, our algorithms can query the oracle for $f_i(S)$ for any $i \in [n]$. We furthermore globally make the following assumptions on the functions $\mathbf{f}$ and g:

1. For all $i \in [n]$, there exists a set $S_i \in \mathcal{S}$ such that $f_i(S_i) \geq 1$. Furthermore, for all $i \in [n]$ and $S \in \mathcal{S}$, it holds that $f_i(S) \leq \Phi_1$ for some $\Phi_1 = \mathrm{poly}(n, m)$.
2. For all $\mathbf{x} \in \mathbb{R}^n_{\geq 0}$, it holds that $\min_{i \in [n]} x_i \leq g(\mathbf{x}) \leq \Phi_2 \cdot \|x\|_1$ for some $\Phi_2 = \mathrm{poly}(n, m)$.

These assumptions are mainly made in order for the following claim to hold (we defer its proof to the full version).

Claim 2.1. *Let* $\mathrm{opt} := \max\{g(\mathbb{E}_{S \sim p}[\mathbf{f}(S)]) : \mathbf{p} \in \Pi\}$ *and assume the above assumptions to hold. Then,* $\frac{1}{n} \leq \mathrm{opt} \leq n \cdot \Phi_1 \Phi_2 = \mathrm{poly}(n, m)$.

The lower bound on the optimum also immediately yields that an arbitrarily small additive error in the approximation can be turned into a multiplicative one:

Claim 2.2. *Let* $\text{opt} := \max\{g(\mathbb{E}_{S\sim p}[\mathbf{f}(S)]) : \mathbf{p} \in \Pi\}$ *and let* $\mathbf{p}$ *be such that* $g(\mathbb{E}_{S\sim p}[\mathbf{f}(S)]) \geq \rho \cdot \text{opt} - \frac{\varepsilon}{n}$ *for some* $\rho \in (0, 1]$ *and* $\varepsilon > 0$*, then* $g(\mathbb{E}_{S\sim p}[\mathbf{f}(S)]) \geq (\rho - \varepsilon) \cdot \text{opt}$.

3 Probabilistic Greedy for Setting 1

Recall Setting 1 described above. Specifically, recall that here we assume that $\mathbf{f} : 2^{[m]} \to \mathbb{R}^n_{\geq 0}$ are n non-negative monotone and submodular set functions. We consider the problem of finding a probability distribution $\mathbf{p} \in \Pi$ maximizing $g\left(\mathbb{E}_{S\sim \mathbf{p}}\left[\mathbf{f}(S)\right]\right)$, where $\Pi := \{\mathbf{p} \in [0, 1]^{\mathcal{S}} : \|\mathbf{p}\|_1 = 1\}$ and $\mathcal{S} := \{S \subseteq U : |S| \leq k\}$. Formally,

$$\text{opt} := \max\{g(\mathbb{E}_{S\sim \mathbf{p}}[\mathbf{f}(S)]) : \mathbf{p} \in \Pi\}. \quad (6)$$

For our algorithm, we need to make the following additional assumption. The function g satisfies the Lipschitz continuity condition $|g(\mathbf{x}) - g(\mathbf{y})| \leq L \cdot \|\mathbf{x} - \mathbf{y}\|_1$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n_{\geq 0}$, for some $L = \text{poly}(m, n)$. As mentioned above, we assume an oracle that, for every $\delta > 0$, computes an additive δ-approximation to the optimization problem in (3).

In this section, we give the probabilistic greedy algorithm (ε-`prob_greedy`), that, for a given $\varepsilon > 0$, computes a distribution over random subsets S of U with at most k elements that is optimal up to a multiplicative factor of $1 - 1/e - \varepsilon$. Formally, we prove the following theorem.

Theorem 3.1. *Let* $\varepsilon > 0$ *and* $\alpha \in \mathbb{N}$*. Assume an oracle that, in* $\text{poly}(n, m, \varepsilon^{-1})$ *time and with high probability, outputs an additive* $\varepsilon/(nk)$*-approximation to the problem* (3)*. Then Algorithm* ε-`prob_greedy` *in* $\text{poly}(n, m, \varepsilon^{-1})$ *time computes vectors* $\mathbf{x}_1, \ldots, \mathbf{x}_k$ *satisfying*

$$g(\mathbb{E}_{S\sim \mathbf{x}^1,\ldots,\mathbf{x}^k}[\mathbf{f}(S)]) \geq \left(1 - \frac{1}{e} - \varepsilon\right) \cdot \text{opt}$$

with probability at least $1 - \mu^{-\alpha}$*, where* $\mu = \max\{n, m\}$.

Here, for k vectors $\mathbf{x}_1, \ldots, \mathbf{x}_k$ each non-negative and of one-norm 1, $\mathbb{E}_{S\sim \mathbf{x}^1,\ldots,\mathbf{x}^k}[\mathbf{f}(S)]$ is the expected value resulting from generating a set S of size at most k by independently sampling the t-th element according to $\mathbf{x}^t$ for $t \in [k]$. Our algorithm returns a distribution that always samples at most k elements, but the returned distribution still achieves a good approximation with respect to the stronger benchmark of choosing at most k items in expectation.

Algorithm 1: ε-`prob_greedy`

1: Let $\delta := \varepsilon/(nk)$ and $\mu := \max\{n, m\}$.
2: **for** $t = 1, \ldots, k$ **do**
3: Compute $\mathbf{x}^t \geq \mathbf{0}$ with $\|\mathbf{x}^t\|_1 = 1$ s.t. $\mathbf{x}^t$ maximizes $g(\mathbb{E}_{j\sim \mathbf{x}^t}[\mathbb{E}_{S^{t-1}\sim \mathbf{x}^1,\ldots,\mathbf{x}^{t-1}}[\mathbf{f}(S^{t-1} \cup j)]])$ up to additive error δ with probability at least $1 - \mu^{-\alpha}/k$.
4: **return** $\mathbf{x}^1, \ldots, \mathbf{x}^k$

3.1 Algorithm

The algorithm works in k iterations. In its t-th iteration, for $t = 1, \ldots, k$, the algorithm greedily computes a distribution $\mathbf{x}^t := (x^t_j)_{j\in U}$ over all m elements of U that maximizes, up to a certain additive error $\delta := \delta(\varepsilon)$ depending on ε, the expected value of g achieved when sampling the t-th element according to $\mathbf{x}^t$, and the previous $t-1$ elements according to the vectors $\mathbf{x}^1, \ldots, \mathbf{x}^{t-1}$ computed in the previous steps. At the end, the algorithm returns the distribution $\mathbf{p}(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ over sets $S \subseteq U$ defined in the exact same way: for any $t \in [k]$ we independently sample the element j_t according to the distribution $\mathbf{x}^t$, resulting in a set $S := \{j_1, \ldots, j_k\}$ made of at most k sampled elements (if an element is sampled multiple times, only one copy of the element is added). As distribution $\mathbf{p}(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ is fully characterized by the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_k$, hence it is sufficient that the algorithm returns those k vectors. We refer the reader to Algorithm 1 for the pseudo-code of ε-`prob_greedy`.

3.2 Approximation Guarantee

Our goal is now to prove Theorem 3.1, i.e., the approximation guarantee for ε-`prob_greedy`.

Progress per Iteration. We start with some preliminary definitions and lemmas with the aim to quantify the progress in objective value that the algorithm is guaranteed to do per iteration. For $t \in [k]$, let $\mathbf{x}^t$ be the probability vector computed by ε-`prob_greedy` in iteration t and let $\mathbf{v}^t := \mathbb{E}_{S^t\sim \mathbf{x}^1,\ldots,\mathbf{x}^t}[\mathbf{f}(S^t)]$. Observe that $g(\mathbf{v}^k)$ is the objective value achieved by the distribution computed by ε-`prob_greedy`. We further extend this notation by $\mathbf{v}^0 = 0$.

Lemma 3.2. *Let* $\varepsilon > 0$*,* $t \in [k]$*, and let* $\mathbf{v}^t = \mathbb{E}_{S^t\sim \mathbf{x}^1,\ldots,\mathbf{x}^t}[\mathbf{f}(S^t)]$*, where* $\mathbf{x}^1, \ldots, \mathbf{x}^t$ *are the vectors computed by* ε-`prob_greedy` *up to iteration t. Then, it holds that* $g(\mathbf{v}^t) - g(\mathbf{v}^{t-1}) \geq \frac{1}{k} \cdot (\text{opt} - g(\mathbf{v}^{t-1})) - \frac{\varepsilon}{nk}$.

We defer the proof of this and the following two lemmata to the full version. This lemma is the main ingredient for proving Theorem 3.1. It remains to show how to compute the vector $\mathbf{x}^t$ in each of the k iterations.

Solving Oracle Problem. Recall that the subproblem in iteration t consists of finding $\mathbf{x}$ that maximizes

$$g\Big(\sum\nolimits_{j\in U} x_j \cdot \mathbb{E}_{S^{t-1}\sim \mathbf{x}^1,\ldots,\mathbf{x}^{t-1}}[\mathbf{f}(S^{t-1} \cup j)]\Big)$$

over linear constraints, namely $\mathbf{x} \geq 0$ and $\mathbb{1}^T\mathbf{x} = 1$. As the function g is concave, this problem can be solved using techniques from convex optimization with the slight caveat that the coefficients $\mathbb{E}_{S^{t-1}\sim \mathbf{x}^1,\ldots,\mathbf{x}^{t-1}}[\mathbf{f}(S^{t-1} \cup j)]$ are not computable exactly and thus have to be approximated using sampling. We discuss this issue next. The proof follows with a straightforward concentration bound. We recall that $\max_{S\subseteq[m]} \|\mathbf{f}(S)\|_\infty \leq \Phi_1$ and that $\mu := \max\{n, m\}$.

Lemma 3.3. *Let* $\Delta > 0$*,* $\alpha \in \mathbb{N}$*,* $t \in [k]$*,* $j \in U$ *and let* $S_1, \ldots, S_T$ *be* $T := \frac{\Phi_1^2}{2\Delta^2} \log(\frac{2knm}{\mu^{-\alpha}})$ *random sets sampled according to* $\mathbf{x}^1, \ldots, \mathbf{x}^{t-1}$ *and let* $\mathbf{m}_j := \frac{1}{T}\sum_{\ell=1}^T \mathbf{f}(S_\ell \cup j)$*. Then, with probability at least* $1 - \mu^{-\alpha}/k$*, we have* $\|\mathbf{m}_j - \mathbb{E}_{S^{t-1}\sim \mathbf{x}^1,\ldots,\mathbf{x}^{t-1}}[\mathbf{f}(S^{t-1} \cup j)]\|_\infty \leq \Delta$ *for all* $j \in U$.

This lemma establishes that we can approximate the expected values that occur in the concave optimization problems. It remains to analyze how this approximation error influences the maximizing distributions of the optimization problems at hand. This is formalized in the following lemma. In order to obtain the respective bound, we use that the function g satisfies a Lipschitz condition of the form $|g(\mathbf{x}) - g(\mathbf{y})| \leq L \cdot \|\mathbf{x} - \mathbf{y}\|_1$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n_{\geq 0}$ (see the assumptions stated at the beginning of the section).

Lemma 3.4. *Let $t \in [k]$, $\delta > 0$, and let $\mathbf{m}_j$ be such that $\|\mathbf{m}_j - \mathbb{E}_{S^{t-1} \sim \mathbf{x}^1, \ldots, \mathbf{x}^{t-1}}[\mathbf{f}(S^{t-1} \cup \{j\})]\|_\infty \leq \frac{\delta}{4Ln}$ for all $j \in U$, where L is the Lipschitz constant of g. Furthermore, assume that $\mathbf{x}$ is an additive $\frac{\delta}{2}$-approximation to*

$$\max\Big\{g\Big(\sum\nolimits_{j \in U} x_j \cdot \mathbf{m}_j\Big) : \|\mathbf{x}\|_1 = 1, \mathbf{x} \geq 0\Big\}. \tag{7}$$

Then, $\mathbf{x}$ is an additive δ-approximation to

$$\max_{\substack{\mathbf{x} \in [0,1]^m: \\ \|\mathbf{x}\|_1 = 1}} \Big\{g(\mathbb{E}_{j \sim \mathbf{x}^t}[\mathbb{E}_{S^{t-1} \sim \mathbf{x}^1, \ldots, \mathbf{x}^{t-1}}[\mathbf{f}(S^{t-1} \cup j)]])\Big\}. \tag{8}$$

We are now ready to prove Theorem 3.1.

Proof of Theorem 3.1. Let $\delta = \varepsilon/(nk)$ as in the algorithm. Apply Lemma 3.3 with $\Delta := \delta/(4Ln)$. Then, a union bound over all k iterations gives that, with probability at least $1 - \mu^{-\alpha}$, it holds that $\|\mathbf{m}_j - \mathbb{E}_{S^{t-1} \sim \mathbf{x}^1, \ldots, \mathbf{x}^{t-1}}[\mathbf{f}(S^{t-1} \cup \{j\})]\|_\infty \leq \delta/(4Ln)$ for all $j \in U$, where the vectors $\mathbf{m}_j$ are as in Lemma 3.3 and can be computed in polynomial time as $T = \varepsilon^{-2} \operatorname{poly}(n, m)$. Now Lemma 3.4 yields that we can compute an additive δ-approximation to the respective optimization problem involving the expected values via an additive $\delta/2$-approximation to the problem involving the vectors $\mathbf{m}$. This problem takes the exact form of the oracle problem (3). Thus, denoting $\mathbf{v}^t := \mathbb{E}_{S^t \sim \mathbf{x}^1, \ldots, \mathbf{x}^t}[\mathbf{f}(S^t)]$, Lemma 3.2 yields $g(\mathbf{v}^t) \geq \frac{1}{k} \cdot \operatorname{opt} + (1 - \frac{1}{k}) \cdot g(\mathbf{v}^{t-1}) - \frac{\varepsilon}{nk}$ for all $t \in [k]$. Applying this k times, we get that, with probability at least $1 - \mu^{-\alpha}$, it holds that

$$g(\mathbf{v}^k) \geq \frac{1}{k} \operatorname{opt} \cdot \sum_{t=0}^{k-1} \Big(1 - \frac{1}{k}\Big)^t - \frac{\varepsilon}{n} \geq \Big(1 - \frac{1}{e}\Big) \operatorname{opt} - \frac{\varepsilon}{n},$$

where the last step uses that $(1 - \frac{1}{k})^k \leq \frac{1}{e}$ for all $k \geq 1$. Using Claim 2.2 completes the proof. □

4 Continuous Greedy for Setting 2

We now turn to the second setting. Recall that here we allow for an arbitrary restriction of the set of feasible solutions, i.e., $\mathcal{S}$ can be any arbitrary subset of 2^U, while the set of distributions are not further restricted. Our goal is to solve

$$\operatorname{opt} := \max\{g(\mathbb{E}_{S \sim \mathbf{p}}[\mathbf{f}(S)]) : \mathbf{p} \in \Pi\}. \tag{9}$$

In this section, we make the further assumption that g is continuously differentiable and that, for some fixed $\xi \in (0, 1]$ and $n \in \mathbb{N}$, there exists $L = \operatorname{poly}(\xi^{-1}, m, n)$ such that the gradient of g satisfies the Lipschitz condition $\|\nabla g(\mathbf{y}) - \nabla g(\mathbf{y} + \mathbf{d})\|_1 \leq L \cdot \|\mathbf{d}\|_1$ for any $\mathbf{y} \geq \xi \cdot \mathbf{1}$ and $\mathbf{d} \geq \mathbf{0}$.[3]

[3]Requiring Lipschitz continuity outside a small region around the origin enables us to employ also functions g with Lipschitz discontinuity in the origin (e.g., Nash social welfare). To mitigate the impact of this discontinuity, our algorithm initially adds a small probability $\varepsilon/\operatorname{poly}(m, n)$ to a feasible set for each player, without meaningfully affecting the resulting approximation ratio (i.e., by at most an arbitrarily small constant). A similar strategy can be used to relax the respective assumption in Setting 1, and is, in fact, used in Setting 3.

We also assume an oracle that computes a multiplicative γ-approximation for the optimization problem in (4).

In this section, we present the continuous greedy algorithm (ε-`cont_greedy`) that, for a given $\varepsilon > 0$, computes a distribution $\mathbf{p} \in \Pi$ with multiplicative approximation factor $1 - 1/e^\gamma - \varepsilon$. Formally, we prove the following theorem.

Theorem 4.1. *Let $\varepsilon > 0$, $\gamma < 1$, and let $\alpha \in \mathbb{N}$. Assume a polynomial-time oracle that computes a γ-approximate solution to the optimization problem stated in (4). Then Algorithm ε-`cont_greedy`, after $\operatorname{poly}(n, m, \varepsilon^{-1})$ iterations in each of which it calls the oracle once, computes a distribution $\mathbf{p} \in \Pi$ satisfying*

$$g(\mathbb{E}_{S \sim \mathbf{p}}[\mathbf{f}(S)]) \geq \Big(1 - \frac{1}{e^\gamma} - \varepsilon\Big) \cdot \operatorname{opt}.$$

As mentioned before our algorithm is similar to the continuous greedy algorithm by Vondrák (2008). We first give a continuous process called ε-`cont_greedy_process` (refer to Algorithm 2), that we will then discretize to obtain the final algorithm ε-`cont_greedy` and to prove Theorem 4.1.

The Continuous Process. We first, with a slight abuse of notation, extend the definition of Π to $\Pi := \{\mathbf{p} \in [0, 1]^{\mathcal{S}} : \|\mathbf{p}\|_1 \leq 1\}$ and note that it contains all probability distributions over $\mathcal{S}$ as the vectors where the constraint is tight. We notice that, due to monotonicity of g and $\mathbf{f}$, this does not change opt. We furthermore write $\mathbb{E}_{S \sim \mathbf{p}}[\mathbf{f}(S)] := \sum_{S \subseteq [m]} p_S \cdot \mathbf{f}(S)$ also for vectors $\mathbf{p}$ with $\|\mathbf{p}\|_1 < 1$. We also define a function $G : \Pi \to \mathbb{R}_{\geq 0}$ as $G(\mathbf{p}) := g(\mathbb{E}_{S \sim \mathbf{p}}[\mathbf{f}(S)])$.

Remark 4.2. *We observe that G can be written as $g \circ \mathbf{L}$, where $\mathbf{L} : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^n$ is defined as $\mathbf{L}(\mathbf{p}) := \mathbb{E}_{S \sim \mathbf{p}}[\mathbf{f}(S)]$. Since $\mathbf{L}$ is a linear function with non-negative coefficients, that is, $\mathbf{L}$ is of the form $\mathbf{L}(\mathbf{p}) = (\sum_{S \in \mathcal{S}} \beta_{j,S} \cdot p_S)_{j \in [n]}$ with $\beta_{j,S} = f_j(S) \geq 0$, it follows that the composition $G = g \circ \mathbf{L}$ inherits the following properties from g: The function G satisfies $G(\mathbf{0}) = 0$, is concave, non-decreasing in every component, and continuously differentiable.*

For a vector $\mathbf{p} \in \Pi$, we define $F_\mathbf{p} : \mathcal{S} \to \mathbb{R}_{\geq 0}$ as the function obtained when differentiating G for p_S, i.e., $F_\mathbf{p}(S) := \frac{\partial}{\partial p_S} G(\mathbf{p}) = \boldsymbol{\alpha}_\mathbf{p}^T \cdot \mathbf{f}(S)$, where $\boldsymbol{\alpha}_\mathbf{p} := \nabla g(\mathbf{y})\big|_{\mathbf{y} = \mathbb{E}_{S' \sim \mathbf{p}}[\mathbf{f}(S')]}$.

Remark 4.3. *Assume that all functions f_j are submodular monotone set functions. As g is non-decreasing in every component, it holds that $\boldsymbol{\alpha}_\mathbf{p} \geq \mathbf{0}$ for all $\mathbf{p} \in \Pi$, where the inequality is component-wise. Thus, $F_\mathbf{p}$ is a submodular monotone set function for any $\mathbf{p} \in \Pi$ in this case.*

ε-`cont_greedy_process` implements a well-defined continuous time process $\mathbf{p}(t)$ with $t \in [\eta, 1]$, as shown below (the proof is deferred to the full version).

Proposition 4.4. *Let $\gamma \in (0, 1]$, $\varepsilon > 0$, and $\eta \in (0, 1]$. Then, the process in ε-`cont_greedy_process`(γ) admits a solution $\mathbf{p}(t)$ for $t \in [\eta, 1]$.*

Then, we establish the approximation factor guaranteed by the continuous process.

Proposition 4.5. *Let $\gamma \leq 1$, $\varepsilon > 0$, and let $\mathbf{p}$ be a solution computed by ε-`cont_greedy_process`(γ). Then, $\mathbf{p}(1) \in \Pi$, $\|\mathbf{p}(1)\|_1 = 1$, and $G(\mathbf{p}(1)) \geq (1 - e^{-\gamma}) \cdot \operatorname{opt} - \frac{\varepsilon}{n}$.*

Algorithm 2: ε-cont_greedy_process(γ)

Set $\delta := \frac{\varepsilon\gamma}{2n(1-e^{-\gamma})}$ and $\eta := \frac{\varepsilon}{2\Phi_1\Phi_2 n}$.
Let S_i be s.t. $f_i(S_i) \geq 1$ for $i \in [n]$ and let $\mathbf{p}(\eta)$ be s.t. $\mathbf{p}(\eta)_{S_i} = \frac{\eta}{n}$ for $i \in [n]$ and $\mathbf{p}(\eta)_S = 0$ else.
for $t \in [\eta, 1]$ **do**
  $\frac{\partial \mathbf{p}(t)}{\partial t^+} := \mathbb{1}_{S(t)}$, where $S(t) \in \mathcal{S}$ is such that $F_{\mathbf{p}(t)}(S(t)) \geq \gamma \cdot \max_{S\in\mathcal{S}} F_{\mathbf{p}(t)}(S) - \delta$.
return $\mathbf{p}(1)$

To show this proposition we rely on the following lemma (the proof is deferred to the full version), that is proven similarly to a claim by Vondrák (2008).

Lemma 4.6. *For every $t \in [\eta, 1]$, it holds that:*

1. $\mathbf{p}(t) \in \Pi$ *and then* $\|\mathbf{p}(1)\|_1 = 1$*;*

2. $\operatorname{opt} - G(\mathbf{p}(t)) \leq \frac{1}{\gamma} \cdot \left(\frac{\partial \mathbf{p}(t)}{\partial t^+}^T \cdot \nabla G(\mathbf{p})\big|_{\mathbf{p}=\mathbf{p}(t)} + \delta \right)$.

Proof of Proposition 4.5. Fix δ, η as in the algorithm. From part (i) of Lemma 4.6, we get $\mathbf{p}(1) \in \Pi$ and $\|\mathbf{p}(1)\|_1 = 1$. Part (ii) of Lemma 4.6 says that, for any $t \in [\eta, 1]$,

$$\begin{aligned} \operatorname{opt} - G(\mathbf{p}(t)) &\leq \frac{1}{\gamma} \cdot \left(\frac{\partial \mathbf{p}(t)}{\partial t^+}^T \cdot \nabla G(\mathbf{p})\big|_{\mathbf{p}=\mathbf{p}(t)} + \delta \right) \\ &= \frac{1}{\gamma} \cdot \left(\frac{\partial}{\partial t^+} G(\mathbf{p}(t)) + \delta \right). \end{aligned}$$

Applying the above statement yields that $\frac{\partial}{\partial t^+}\left(e^{\gamma t} G(\mathbf{p}(t))\right) = \gamma e^{\gamma t} G(\mathbf{p}(t)) + e^{\gamma t} \frac{\partial}{\partial t^+} G(\mathbf{p}(t)) \geq \left(\operatorname{opt} - \frac{\delta}{\gamma}\right) \cdot \gamma e^{\gamma t}$. Integrating this inequality in $t \in [\eta, 1]$ yields $\left[e^{\gamma t} \cdot G(\mathbf{p}(t))\right]_\eta^1 \geq \int_\eta^1 \left(\operatorname{opt} - \frac{\delta}{\gamma}\right) \cdot \gamma e^{\gamma t} dt = \left(\operatorname{opt} - \frac{\delta}{\gamma}\right) \cdot (e^{\gamma} - e^{\gamma\eta})$. Together with $G(\mathbf{p}(\eta)) \geq 0$, $\delta = \frac{\varepsilon\gamma}{2n(1-e^{-\gamma})}$, $\eta = \frac{\varepsilon}{2\Phi_1\Phi_2 n}$ and $\operatorname{opt} \leq n \cdot \Phi_1\Phi_2$ (see Claim 2.1) we obtain $G(\mathbf{p}(1)) \geq (1 - e^{-\gamma(1-\eta)}) \cdot \left(\operatorname{opt} - \frac{\delta}{\gamma}\right) \geq (1 - e^{-\gamma}) \cdot \operatorname{opt} - \frac{\varepsilon}{n}$. Using Claim 2.2 completes the proof. □

Finally, we design ε-cont_greedy as a polynomial-time discretization of the continuous process, ensuring that its solutions are also valid solutions for the continuous version, thereby preserving the approximation guarantee established in Proposition 4.5. Further details on the discretization and approximation are deferred to the full version.

5 Probabilistic Greedy for Setting 3

Regarding the assignments we now adopt a notation involving partitions. That is, we let $A := [n]$ be a set of n *agents* and $J := [M]$ be a set of M items. For any $i \in A$, let $f_i : 2^J \to \mathbb{R}_{\geq 0}$ be the *utility function* of agent i, that returns the utility $f_i(S)$ that agent i gets when receiving a subset of items $S \subseteq J$. We assume the functions f_i to be submodular. An *assignment* of items in J to agents in A can equivalently be defined as a sequence $\mathbf{S} = \langle S_1, \ldots, S_n \rangle$ that corresponds to a partition of J, i.e., $\bigcup_{i\in[n]} S_i = J$ and $S_i \cap S_{i'} = \emptyset$ for all $i \neq i'$. We write $f_i(\mathbf{S}) := f_i(S_i)$ to denote the utility achieved by i when receiving the set of items S_i assigned to him by assignment $\mathbf{S}$. We also write $\mathbf{f}(\mathbf{S})$ for the vector $(f_1(\mathbf{S}), \ldots, f_n(\mathbf{S}))$ and let $\Pi := \{\mathbf{p} \in [0,1]^{n^M} : \|\mathbf{p}\|_1 = 1\}$ be the set of probability distributions over all possible n^M assignments $\mathbf{S} = \langle S_1, \ldots, S_n \rangle$. We then want to solve

$$\operatorname{opt} := \max\{g(\mathbb{E}_{\mathbf{S}\sim\mathbf{p}}[\mathbf{f}(\mathbf{S})]) : \mathbf{p} \in \Pi]\}. \qquad (10)$$

In this section, we make the additional assumption that, for any $\xi \in (0, 1]$, there exists $L = \operatorname{poly}(\xi^{-1}, n, M)$ such that the function g satisfies the Lipschitz condition $|g(\mathbf{y} + \mathbf{d}) - g(\mathbf{y})| \leq L \cdot \|\mathbf{d}\|_1$ for any $\mathbf{y} \geq \xi \cdot \mathbf{1}$ and $\mathbf{d} \geq \mathbf{0}$. Again, we assume that there exists an oracle that, for every $\delta > 0$, computes an additive δ-approximation to the optimization problem in (3). Formally, we prove the following theorem.

Theorem 5.1. *Let $\varepsilon > 0$ and let $\alpha \in \mathbb{N}$. Assume an oracle that, in $\operatorname{poly}(n, M, \varepsilon^{-1})$ time and with high probability, computes an additive ε/M-approximation to the optimization problem stated in (5). Then we can compute in $\operatorname{poly}(\varepsilon^{-1}, n, M)$ time M vectors $\mathbf{x}_1, \ldots, \mathbf{x}_M$ satisfying*

$$g(\mathbb{E}_{\mathbf{S}\sim\mathbf{x}^1,\ldots,\mathbf{x}^M}[\mathbf{f}(\mathbf{S})]) \geq \left(\frac{1}{2} - \varepsilon\right) \cdot \operatorname{opt}$$

with probability at least $1 - \mu^{-\alpha}$, where $\mu = \max\{n, M\}$.

Here $\mathbb{E}_{\mathbf{S}\sim\mathbf{x}^1,\ldots,\mathbf{x}^M}[\mathbf{f}(\mathbf{S})]$ denotes the expected value that results from generating an assignment $\mathbf{S}$ of the M elements to the n agents by independently sampling the agent to which item $t \in [M]$ is assigned according to $\mathbf{x}^t$. The algorithm in Theorem 5.1 first assigns each agent a specific item with probability $1/\operatorname{poly}(n, M, \varepsilon^{-1})$, and then proceeds for M iterations. In the t-th iteration, it greedily computes a distribution $\mathbf{x}^t := (x_i^t)_{i\in A}$ over n agents to assign item t. This distribution maximizes, up to an additive error $\delta := \delta(\varepsilon)$, the expected value of g when sampling the agent for item t according to $\mathbf{x}^t$, and previous items according to $\mathbf{x}^1, \ldots, \mathbf{x}^{t-1}$. Full details and the proof are deferred to the full version.

6 Conclusion

We presented a general framework for computing fair solutions to settings with a set of agents whose valuations f_i are aggregated using a function g (e.g., the minimum, Nash social welfare, p-norm). Besides its generality, the strength of our framework lies in the power of ex-ante fairness, a concept that has recently gained much attention in the scope of fair allocation and fairness in AI in general.

An open problem left by our work is closing the gap on the approximation achieved by Settings 2 and 3. Another interesting research direction would be to design ad-hoc algorithms for other combinatorial structures, e.g., matroids, as in (Călinescu et al. 2011) or to consider problem variants where the functions are subject to observable stochastic events, as in adaptive optimization (Dean, Goemans, and Vondrák 2005; Asadpour and Nazerzadeh 2016; Golovin and Krause 2011; Chen and Peng 2019; D'Angelo, Poddar, and Vinci 2023) or multi-armed bandits (Auer, Cesa-Bianchi, and Fischer 2002; Gabillon et al. 2013; Chen et al. 2016; Wen et al. 2017). Finally, it would be interesting to experimentally evaluate the quality of solutions produced by our approximation algorithms on random or real-world instances of relevant optimization problems.

Acknowledgements

This work was partially supported by: the PNRR MIUR project FAIR - Future AI Research (PE00000013), Spoke 9 - Green-aware AI; the MUR - PNRR IF Agro@intesa; the Project SERICS (PE00000014) under the NRRP MUR program funded by the EU – NGEU; GNCS-INdAM. We thank the anonymous reviewers for their insightful comments.

References

Ahmed, F.; Dickerson, J. P.; and Fuge, M. D. 2017. Diverse Weighted Bipartite b-Matching. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*, 35–41.

Akrami, H.; Chaudhury, B. R.; Hoefer, M.; Mehlhorn, K.; Schmalhofer, M.; Shahkarami, G.; Varricchio, G.; Vermande, Q.; and van Wijland, E. 2022. Maximizing Nash Social Welfare in 2-Value Instances. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*, 4760–4767.

Amanatidis, G.; Birmpas, G.; Filos-Ratsikas, A.; Hollender, A.; and Voudouris, A. A. 2021. Maximum Nash welfare and other stories about EFX. *Theor. Comput. Sci.*, 863: 69–85.

Amoruso, M.; Anello, D.; Auletta, V.; Cerulli, R.; Ferraioli, D.; and Raiconi, A. 2020. Contrasting the Spread of Misinformation in Online Social Networks. *J. Artif. Intell. Res.*, 69: 847–879.

Asadpour, A.; and Nazerzadeh, H. 2016. Maximizing stochastic monotone submodular functions. *Management Science*, 62(8): 2374–2391.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. *Mach. Learn.*, 47(2-3): 235–256.

Banerjee, A.; Chandrasekhar, A. G.; Duflo, E.; and Jackson, M. O. 2013. The Diffusion of Microfinance. *Science*, 341(6144): 1236498.

Banerjee, S.; Gkatzelis, V.; Gorokh, A.; and Jin, B. 2022. Online Nash Social Welfare Maximization with Predictions. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA*, 1–19. SIAM.

Becker, R.; D'Angelo, G.; and Ghobadi, S. 2023. On the Cost of Demographic Parity in Influence Maximization. In Williams, B.; Chen, Y.; and Neville, J., eds., *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023*, 14110–14118. AAAI Press.

Becker, R.; D'Angelo, G.; Ghobadi, S.; and Gilbert, H. 2022. Fairness in Influence Maximization through Randomization. *J. Artif. Intell. Res.*, 73: 1251–1283.

Bilò, V.; Monaco, G.; Moscardelli, L.; and Vinci, C. 2022. Nash Social Welfare in Selfish and Online Load Balancing. *ACM Trans. Economics and Comput.*, 10(2): 8:1–8:41.

Călinescu, G.; Chekuri, C.; Pá, M.; and Vondrák, J. 2011. Maximizing a Monotone Submodular Function Subject to a Matroid Constraint. *SIAM J. Comput.*, 40(6): 1740–1766.

Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A. D.; Shah, N.; and Wang, J. 2019. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Trans. Economics and Comput.*, 7(3): 12:1–12:32.

Chen, W.; and Peng, B. 2019. On Adaptivity Gaps of Influence Maximization Under the Independent Cascade Model with Full-Adoption Feedback. In *ISAAC*, 24:1–24:19.

Chen, W.; Wang, Y.; Yuan, Y.; and Wang, Q. 2016. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *The Journal of Machine Learning Research*, 17(1): 1746–1778.

Cole, R.; Devanur, N. R.; Gkatzelis, V.; Jain, K.; Mai, T.; Vazirani, V. V.; and Yazdanbod, S. 2017. Convex Program Duality, Fisher Markets, and Nash Social Welfare. In *Proceedings of the 2017 ACM Conference on Economics and Computation (EC)*, 459–460.

Cole, R.; and Gkatzelis, V. 2018. Approximating the Nash Social Welfare with Indivisible Items. *SIAM J. Comput.*, 47(3): 1211–1236.

D'Angelo, G.; Poddar, D.; and Vinci, C. 2023. Better bounds on the adaptivity gap of influence maximization under full-adoption feedback. *Artif. Intell.*, 318: 103895.

Dean, B. C.; Goemans, M. X.; and Vondrák, J. 2005. Adaptivity and approximation for stochastic packing problems. In *SODA*, 395–404.

Debreu, G. 1959. Topological methods in cardinal utility theory. *Mathematical Methods in Social Sciences*, 16–26.

Dobzinski, S.; Li, W.; Rubinstein, A.; and Vondrák, J. 2024. A Constant-Factor Approximation for Nash Social Welfare with Subadditive Valuations. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC*, 467–478. ACM.

Feldman, M.; Naor, J.; Schwartz, R.; and Ward, J. 2011. Improved Approximations for k-Exchange Systems - (Extended Abstract). In Demetrescu, C.; and Halldórsson, M. M., eds., *Algorithms - ESA 2011 - 19th Annual European Symposium, Proceedings*, 784–798.

Fish, B.; Bashardoust, A.; danah boyd; Friedler, S. A.; Scheidegger, C.; and Venkatasubramanian, S. 2019. Gaps in Information Access in Social Networks? In *The World Wide Web Conference, WWW*, 480–490. ACM.

Fisher, M. L.; Nemhauser, G. L.; and Wolsey, L. A. 1978. *An analysis of approximations for maximizing submodular set functions—II*, 73–87. Berlin, Heidelberg: Springer Berlin Heidelberg.

Gabillon, V.; Kveton, B.; Wen, Z.; Eriksson, B.; and Muthukrishnan, S. 2013. Adaptive submodular maximization in bandit setting. *NeurIPS*, 26.

Garg, J.; Husic, E.; Li, W.; Végh, L. A.; and Vondrák, J. 2023. Approximating Nash Social Welfare by Matching and Local Search. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC*, 1298–1310. ACM.

Garg, J.; Kulkarni, P.; and Kulkarni, R. 2023. Approximating Nash Social Welfare under Submodular Valuations through (Un)Matchings. *ACM Trans. Algorithms*, 19(4): 36:1–36:25.

Golovin, D.; and Krause, A. 2011. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42: 427–486.

Gorman, W. 1968. The Structure of Utility Functions. *The Review of Economic Studies*, 35(4): 367–390.

Heidari, H.; Ferrari, C.; Gummadi, K. P.; and Krause, A. 2018. Fairness Behind a Veil of Ignorance: A Welfare Analysis for Automated Decision Making. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS*, 1273–1283.

Kempe, D.; Kleinberg, J. M.; and Tardos, É. 2015. Maximizing the Spread of Influence through a Social Network. *Theory Comput.*, 11: 105–147.

Li, W.; and Vondrák, J. 2021a. A constant-factor approximation algorithm for Nash Social Welfare with submodular valuations. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021*, 25–36. IEEE.

Li, W.; and Vondrák, J. 2021b. Estimating the Nash Social Welfare for coverage and other submodular valuations. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA*, 1119–1130. SIAM.

Machina, M. J. 1989. Dynamic consistency and non-expected utility models of choice under uncertainty. *Journal of Economic Literature*, 27(4): 1622–1668.

Nash, J. 1950. The bargaining problem. *Econometrica*, 18(2): 155–162.

Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.*, 14(1): 265–294.

Nesterov, Y. E.; and Nemirovskii, A. 1994. *Interior-point polynomial algorithms in convex programming*, volume 13 of *Siam studies in applied mathematics*. SIAM.

Nguyen, T. T.; and Rothe, J. 2014. Minimizing envy and maximizing average Nash social welfare in the allocation of indivisible goods. *Discrete Applied Mathematics*, 179: 54–68.

Valente, T. W.; and Pumpuang, P. 2007. Identifying Opinion Leaders to Promote Behavior Change. *Health Education & Behavior*, 34: 881 – 896.

Vondrák, J. 2008. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC*, 67–74.

Wen, Z.; Kveton, B.; Valko, M.; and Vaswani, S. 2017. Online influence maximization under independent cascade model with semi-bandit feedback. *NeurIPS*, 30.

Wilder, B.; Onasch-Vera, L.; Hudson, J.; Luna, J.; Wilson, N.; Petering, R.; Woo, D.; Tambe, M.; and Rice, E. 2018. End-to-End Influence Maximization in the Field. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS*, 1414–1422.

Yadav, A.; Wilder, B.; Rice, E.; Petering, R.; Craddock, J.; Yoshioka-Maxwell, A.; Hemler, M.; Onasch-Vera, L.; Tambe, M.; and Woo, D. 2018. Bridging the Gap Between Theory and Practice in Influence Maximization: Raising Awareness about HIV among Homeless Youth. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI*, 5399–5403.