Contention Resolution, Matrix Scaling and Fair Allocation

Nikhil Bansal[1] and Ilan Reuven Cohen[2]

[1] TU Eindhoven and CWI, Netherlands. n.bansal@tue.nl
[2] Bar-Ilan University, Israel.* ilan-reuven.cohen@biu.ac.il

Abstract. A contention resolution (CR) scheme is a basic algorithmic primitive, that deals with how to allocate items among a random set S of competing players, while maintaining various properties. We consider the most basic setting where a *single* item must be allocated to some player in S. Here, in a seminal work, Feige and Vondrak (2006) designed a *fair* CR scheme when the set S is chosen from a product distribution. We explore whether such fair schemes exist for arbitrary non-product distributions on sets of players S, and whether they admit simple algorithmic primitives. Moreover, can we go beyond fair allocation and design such schemes for all possible achievable allocations.

We show that for any arbitrary distribution on sets of players S, and for any achievable allocation, there exist several natural CR schemes that can be succinctly described, are simple to implement and can be efficiently computed to any desired accuracy. We also characterize the space of achievable allocations for any distribution, give algorithms for computing an optimum fair allocation for arbitrary distributions, and describe other natural fair CR schemes for product distributions. These results are based on matrix scaling and various convex programming relaxations.

* Part of the work has been done while the author was a postdoctoral fellow at CWI Amsterdam and TU Eindhoven.

1 Introduction

Contention resolution (CR) schemes are basic algorithmic primitives that arise naturally, either implicitly or explicitly, in many optimization problems. One of the most basic version of contention resolution was first introduced by Feige and Vondrak [11, 12], in the context of allocating items to players, and it deals with the following setting: there is a single item, and there are n players, out of which some subset S might request the item. However, the item can only be allocated to a single player in S and the goal is to decide how to allocate the item in some fair and optimal way.

More formally, there is an underlying distribution $\mathcal{P}$ over subsets $S \subseteq [n]$ of n players, that specifies the probability p_S that subset S of players request the item. A CR scheme or rule R, specifies for each set S and player $i \in S$, the probability $r_{S,i}$ that player i gets the item. Given such a rule R, player i receives the item with probability $g_i = \sum_{S:i\in S} r_{S,i}\, p_S$. The goal then, is to find a suitable rule R, so that resulting allocation vector $g = (g_1, \ldots, g_n)$ satisfies some desired fairness and optimality properties, depending on the application at hand.

Such questions arise in designing rounding-based algorithms for several problems, like those involving allocation of jobs to machines, or goods to users in combinatorial auctions, labeling problems where a vertex must pick one of several possible labels, assignment problems where a vertex much be matched to one of several neighbors and so on. Here, the solution to the natural linear or convex relaxation gives some distribution $\mathcal{P}$ over possible assignments for each item, out of which exactly one must be chosen. For this reason various CR schemes have received a lot of attention in both offline, online, stochastic and submodular optimization [1, 8, 11, 13, 15, 20].

Here, we focus on the most basic version due to Feige and Vondrak [11] described above.

Feige-Vondrak scheme for product distributions. Feige and Vondrak considered the case where $\mathcal{P}$ is a product distribution corresponding to marginals q_i, i.e., $p_S = \prod_{i\in S} q_i \prod_{i\notin S}(1-q_i)$. The goal is to allocating the item fairly, i.e., $g_i = \alpha q_i$ (each player receives the item in proportion to its marginal probability of requesting it), with α as large as possible.

Perhaps surprisingly, the solution turns out to be quite non-trivial already for product distributions, and the natural strategies such as allocating players in S uniformly ($r_{S,i} = 1/|S|$), or in proportion to q_i ($r_{S,i} = q_i/(\sum_{i\in S} q_i)$) are sub-optimal [11]. Feige and Vondrak give the following elegant rule (that we call FV scheme): For $S = \{i\}$, $r_{S,i} = 1$, otherwise for all S with $|S| > 1$

$$r_{S,i} = \frac{1}{\sum_{j=1}^{n} q_j} \left(\sum_{j\in S\setminus\{i\}} \frac{q_j}{|S|-1} + \sum_{j\notin S} \frac{q_j}{|S|} \right) \qquad \forall i \in S. \tag{1}$$

They show that this gives a *fair* allocation with $g_i = \alpha q_i$, with the best possible

$$\alpha = (1 - p_\phi)/(\sum_i q_i) = (1 - \prod_{i=1}^{n}(1-q_i))/(\sum_i q_i).$$

Notice that for this α, we have $\sum_i g_i = \alpha(\sum_i q_i) = 1 - p_\phi$, so the rule always allocates an item whenever possible, i.e., $\sum_{i \in S} r_{S,i} = 1$ for all $S \neq \emptyset$.

1.1 Extensions and Motivating Questions

The work of Feige and Vondrak [11] raises several natural questions, which motivate our work.

Question 1. (Arbitrary distributions) Can we go beyond product distributions, and design CR schemes with similar guarantees for *arbitrary* distributions $\mathcal{P}$?

Our starting point for exploring this question is that the need for such CR schemes for general distributions arises while rounding fractional solutions to more complex relaxations such as configuration LPs [3, 15, 17] or SDPs [2, 7], where the underlying variables are correlated, and hence the distribution $\mathcal{P}$ on the resulting sets S is not a product distribution.

Exploring general distributions $\mathcal{P}$ raises several other questions: (i) How is the distribution $\mathcal{P}$ specified? (ii) How does one describe the CR scheme (specifying $r_{S,i}$ explicitly for each S, i could use exponential space)? (iii) Do there exist CR schemes with succinct description, even if $\mathcal{P}$ is completely arbitrary? (iv) Can such CR schemes be found efficiently even if the algorithm only has sampling access to $\mathcal{P}$?

Question 2. (General allocation vectors) What is the space of all possible allocation vectors g achievable by CR schemes? Given arbitrary target allocation g and distribution $\mathcal{P}$, can we efficiently determine whether g is achievable, and find such a scheme for g if it is achievable?

In some applications, one may be interested in allocations g that are not necessarily fair, and it is desirable to design a CR scheme that achieves such a g. It is unclear how to do this even for product distributions $\mathcal{P}$, as the FV-scheme only describes the rule for fair allocations of the form $g = \alpha(q_1, \dots, q_n)$. The question becomes even more intriguing for arbitrary distributions $\mathcal{P}$.

Questions 1 and 2 above consider general distributions $\mathcal{P}$ and general allocation vectors g. It is also interesting to explore the production distribution $\mathcal{P}$ further.

Question 3. (Other CR schemes for product distributions) Are there other natural CR schemes for product distributions $\mathcal{P}$ that achieve the same allocation as the FV-scheme?

Finally, for general distributions $\mathcal{P}$, the right generalization of a maximally fair allocation is the widely studied notion of max-min fairness, also referred to as lexicographic max-min fairness [4, 10, 18, 19]. We define this formally later below, but intuitively an allocation is max-min fair if no player's share can be increased without decreasing that of another player with *lower* share.

Question 4. Can we efficiently find an optimum max-min fair allocation for an arbitrary distribution $\mathcal{P}$?

In this work, we answer these questions in the affirmative and give several other structural and algorithmic results. These results are described in Section

1.2 below. Our results reveal a rich general structure in CR schemes and suggest several new directions for further investigation. Before describing our results we give some notation and basic definitions.

Notation There are n players denoted by $N = [n]$, and $\mathcal{P}$ specifies an arbitrary distribution on subset of players $S \subseteq [n]$ requesting the single item. We use $\mathrm{supp}(\mathcal{P})$ to denote the support of $\mathcal{P}$, and p_S to denote the probability of set S under $\mathcal{P}$. We use $q_i = \sum_{S \ni i} p_S$ to denote the marginal probabilities of player i requesting the item, and use $\mathcal{P} = \langle q_1, \ldots, q_n \rangle$ to denote the product distribution on n players with marginals $q_1, \ldots, q_n$ respectively. A CR scheme R, specifies the probabilities $r_{S,i}$ of giving the item to player $i \in S$ upon seeing the set S. We say that an allocation vector $g = (g_1, \ldots, g_n)$ is *achievable* if there exists some CR scheme R such that for each $i \in [n]$

$$g_i = g_i(R, \mathcal{P}) := \sum_{S \subseteq [n]} r_{S,i}\, p_S.$$

The achievable vectors g form a convex polyhedral set $G = G(\mathcal{P})$ (see Section 1.3) and we use $\mathrm{int}(G)$ to denote the interior of G. We call a goal vector *maximal* if an item is assigned whenever possible, i.e., for all $S \neq \emptyset$, $\sum_{i \in S} r_{S,i} = 1$, or equivalently when $\sum_i g_i = 1 - p_\emptyset$. Denote $G^M(\mathcal{P}) = \{g \in G(\mathcal{P}) : \sum_{i \in [n]} g_i = 1 - p_\emptyset\}$ as the set of maximal achievable vectors. While we state our results for maximal achievable vectors $g \in G^M(\mathcal{P})$, they hold for any $g \in G(\mathcal{P})$ by a simple reduction[3].

Max-min Fairness. The FV allocation is both fair and maximal. However, for general distributions $\mathcal{P}$, both these properties need not hold simultaneously[4]. One must allow players to have different $\alpha_i = g_i/q_i$, and the right notion to consider is max-min fairness[5].

For a vector $\alpha = (\alpha_1, \ldots, \alpha_n)$, let $\alpha^\uparrow$ be the vector with entries of α sorted in non-decreasing order. Given vectors $\alpha, \beta \in \mathbb{R}^n$ we say that α is *fairer* than β, denoted as $\alpha \succeq \beta$, if $\alpha^\uparrow$ is lexicographically at least as large $\beta^\uparrow$ (either $\alpha^\uparrow = \beta^\uparrow$, or there is an index $k \in [n]$ such that $\alpha^\uparrow(i) = \beta^\uparrow(i)$ for $i < k$ and $\alpha^\uparrow(k) > \beta^\uparrow(k)$). For an achievable goal vector $g \in G(\mathcal{P})$, let $\alpha(g)$ denote the vector with entries $\alpha_i(g) = g_i / q_i$. Then we say that g is a max-min fair allocation if $\alpha(g) \succeq \alpha(g')$ than any other feasible allocation g'. It is not hard to see that such an allocation is also maximal.

[3] Define a distribution $\tilde{\mathcal{P}}$ on $[n+1]$ players, where for $S \subseteq [n+1]$, we set $\tilde{p}_S = p_{S \setminus \{n+1\}}$ if $n+1 \in S$ and $\tilde{p}_S = 0$ otherwise, and $g_{n+1} = 1 - p_\emptyset - \sum_{i \in [n]} g_i$. Then $\sum_{i \in [n+1]} g_i = 1 - p_\emptyset$ and whenever the item is assigned to player $n+1$ in the CR scheme for $\tilde{\mathcal{P}}$ we do not assign it at all in the scheme for $\mathcal{P}$.

[4] Suppose $n > 3$ and $S = \{1\}$ or $S = \{1, \ldots, n\}$, each with probability $1/2$. Then $q_1 = 1$ and $q_i = 1/2$ for $i \geq 2$. For a *balanced* allocation with $g = \alpha q$, the best g is $(1/(n-1), 1/2(n-1), \ldots, 1/2(n-1))$, but here $\sum_i g_i < 1 - p_\emptyset$.

[5] In the example above, this allows one to achieve the allocation $(1/2, 1/2(n-1), \ldots, 1/2(n-1))$, which is clearly better.

More generally, we will consider max-min fair allocations with respect to any priority values of the players $V = \langle v_1, \ldots, v_n \rangle$, where $v_i \in R^+$ and $\alpha_i^V(g) = g_i/v_i$, and define the fairness relation ($\succeq_V$) accordingly (the special case above corresponds to $V = q$). That is, an achievable goal vector $g \in G(\mathcal{P})$ is max-min fair with respect to V, if $g \succeq_V g'$ for any achievable vector $g' \in G(\mathcal{P})$.

1.2 Our Results

We first show that there exist very natural and succinct to describe CR schemes, even under the most general setting of arbitrary distribution $\mathcal{P}$ and any feasible allocation vector g.

Weight-based schemes A CR scheme is *weight-based* if it has the following form: Each player $i \in [n]$ has a (fixed) weight $w_i \geq 0$, and for every non-empty set S and player $i \in S$,

$$r_{S,i} = w_i / \big(\sum_{j \in S} w_j \big).$$

That is, for each set S the rule simply assigns the item to $i \in S$ with probability proportional to w_i. Note that as the weights w_i are independent of S, the CR rule is extremely simple and is succinctly described by only n numbers. Besides simplicity this rule also has several other useful properties, and is widely studied in social discrete choice and econometrics and is referred to as the Multinomial Logit Model [14, 22, 26].

Theorem 1. *For any distribution $\mathcal{P}$ on the players, and any maximal achievable allocation vector in $g \in int(G^M(\mathcal{P}))$, there exists a weight-based scheme achieving g.*

We remark that the condition $g \in \text{int}(G^M)$ is necessary. E.g., suppose $n = 2$, and $S = \{1\}$ or $\{1, 2\}$ with probability 1/2 each. Then the allocation $g = (1/2, 1/2)$ is achievable (choose player 1 if $S = \{1\}$ and player 2 otherwise). But in a weight-based scheme as player 2 is only chosen with probability $w_2/(w_1 + w_2)$ when $S = \{1, 2\}$, and as $p_{1,2} = 1/2$, its allocation approaches 1/2 only as $w_2/w_1 \to \infty$.

We give two proofs of Theorem 1. The first is based on formulating a suitable max-entropy convex program [16, 27] and using convex programming duality [5]. The second proof is based on applying the classic matrix-scaling algorithm [21, 23, 25] to a suitably chosen matrix.

These proofs give algorithms that take $\mathcal{P}$ and g as input and some error parameter $\varepsilon > 0$, and return weights w corresponding to some achievable $\hat{g}$ with $\|g - \hat{g}\| \leq \varepsilon$ (if it exists) in time $\text{poly}(n, \text{supp}(\mathcal{P}), \log(1/\varepsilon))$. However, this is not so useful as the support of $\mathcal{P}$ is exponentially large typically. Using standard sampling based techniques, these can be adapted to run with only access to samples from $\mathcal{P}$. This removes the dependence on $\text{supp}(\mathcal{P})$ at the expense of increasing the dependence on ε from $\log(1/\varepsilon)^{O(1)}$ to $1/\varepsilon^{O(1)}$.

Theorem 2. *Given sampling access to $\mathcal{P}$, there are algorithms based on convex programming and on matrix-scaling, that given a target allocation g and $\varepsilon > 0$, with high probability, find some $\hat{g} \in G(\mathcal{P})$ satisfying $\|g - \hat{g}\|_\infty \leq \varepsilon$, or certify that no such $\hat{g}$ exists. Both algorithms use $O(\varepsilon^{-2} \log n)$ samples and run in time* $\mathrm{poly}(n, \varepsilon^{-1})$.

Permutation schemes and characterizing the set G^M Next, we consider another natural class of CR schemes based on what we call the *permutation scheme*. In a permutation scheme, the players are ordered according to some fixed permutation π, and given a set of players S, the item is assigned to the first player in S according to π.

We show the following general result.

Theorem 3. *For any arbitrary distribution $\mathcal{P}$ and given any feasible goal vector $g \in G^m(\mathcal{P})$, there exists a CR scheme of the following form: A random permutation π is chosen from a (fixed) distribution, depending only on g and $\mathcal{P}$, over at most $n+1$ permutations. Then, given a set S, the item is all allocated according to the permutation scheme given by π.*

Notice that these schemes are succinctly described, requiring only $n + 1$ permutations on the n players, and exist in the most general setting possible. The Theorem above follows from the result below which gives an explicit characterization of the set $G^M(\mathcal{P})$ of achievable allocations in the most general setting using permutation schemes.

Theorem 4. *For any arbitrary distribution on players $\mathcal{P}$, the convex set $G^M(\mathcal{P})$ is the convex hull of the goal vectors $g(\pi)$ of permutation schemes.*

This result is based on a non-trivial connection between the allocations achievable by convex combination of permutation schemes and weight-based schemes. Apriori, it is unclear why such schemes exist even for product distributions.

Finally, we give an efficient algorithm to checking whether some given g is achievable, and obtain the corresponding allocation rule.

Theorem 5. *If the distribution $\mathcal{P}$ has the property that the allocation vector $g(\pi)$ can be computed[6] for any permutation π, then for any vector g there is an efficient algorithm to test if $g \in int(G(P))$, and to compute the corresponding CR-scheme.*

The Feige-Vondrak setting and Sequential Schemes. While the above results hold for arbitrary $\mathcal{P}$ and g, we now revisit the Feige Vondrak setting, where $\mathcal{P}$ is a product distribution and $g = \alpha q$ is the fair allocation. We give another very natural scheme that we call the *Sequential scheme*.

A sequential scheme has a very simple form. We fix an order of players (say $1, \ldots, n$) and compute numbers $\gamma_1, \ldots, \gamma_n$. When S is realized, we go over the

[6] It can always be computed efficiently to any desired accuracy $\varepsilon > 0$ using $\mathrm{poly}(n, 1/\varepsilon)$ samples from $\mathcal{P}$.

players sequentially in the fixed order, and give the item to player i in S with probability γ_i, unless it is the last player in S, in which case it gets the item with probability 1.

As the γ_i are independent of the set S, this gives a succinct scheme with only n parameters. The following result shows that sequential schemes can achieve the same allocation as the FV rule. The proof of this result also gives an explicit expression for the γ_i as a function of $q_1, \ldots, q_n$.

Theorem 6. *For any product distribution $\mathcal{P} = \langle q_, \ldots, q_n \rangle$, and for any ordering of the players, there exists a sequential scheme, that achieves the maximally fair allocation $g_i = \alpha q_i$ for all $i \in [n]$.*

Unfortunately, sequential schemes are not general enough, and we give two examples which show their limitations in a strong way: (i) there exist non-product distributions $\mathcal{P}$, for which the fair allocation g is feasible, but no sequential scheme can achieve it, and (ii) even for $\mathcal{P}$ a product distribution, there exist (non-fair) feasible vectors g, that cannot be achieved by a sequential scheme.

Variance Minimizing Program Another natural question is whether the FV-scheme in (1), can be viewed more systematically, and obtained as an optimum solution to some natural convex program. We show that this is indeed the case, and it corresponds to a convex program that minimizes a certain weighted squared error objective.

However, this convex program also does not seem too useful for more general settings. In particular, we give examples of (i) general $\mathcal{P}$ for which the fair allocation is feasible, and (ii) where $\mathcal{P}$ is a product distribution for g is not the fair allocation, for which the program does not have a nice structured solution. Finally, we also consider a general class of such convex programs and show their solutions can have a form similar to that of (1).

Efficiently computing the max-min fair allocation For general $\mathcal{P}$, while Theorems 1 and 2 give an efficient test to determine if an allocation vector g is feasible, they do not give an algorithm for computing the max-min fair allocation. Despite extensive work on fairness and in particular on computing max-min fair allocations in both continuous and discrete settings, we are not aware of any work in our setting.

We show the following result for max-min fair allocation with respect to a general *priority* values.

Theorem 7. *For any distribution $\mathcal{P}$ and for any priority vector V, the max-min allocation $g \in G^M(P)$ can be computed exactly in time poly$(n, |supp(\mathcal{P})|)$ using a liner program. If an ϵ additive error is allowed, the time is poly$(n, 1/\varepsilon)$. Finally, an exact computation in poly(n) time is also possible if $g(\pi)$ can be computed efficiently for any permutation π.*

Our algorithm follows a natural approach of finding and removing the most *critical* subset of players and iterating on the residual instance. To do this, we extend an LP-based algorithm of Charikar for computing the densest subgraph in a graph [6], to hypergraphs. For the setting where $g(\pi)$ can be computed efficiently for any permutation, we consider a different LP to compute to the most critical set.

1.3 Preliminaries

We describe some basic properties of the set $G^M(\mathcal{P})$ of maximal achievable goal vectors, that will be useful later. First, we note that $G^M(\mathcal{P})$ is a convex set for an underlying distribution over players $\mathcal{P}$. Let $g, g' \in G^M(\mathcal{P})$ with corresponding CR schemes $R = \{r_{S,i}\}_{S,i}$ and $R' = \{r'_{S,i}\}_{S,i}$. Then for $\alpha \in [0,1]$, the goal vector $g^{(\alpha)} = \alpha g + (1-\alpha) g'$ is also valid, as the CR scheme $R^{(\alpha)}$ with $r^{(\alpha)}_{S,i} = \alpha r_{S,i} + (1-\alpha) r'_{S,i}$ gives that for any $i \in [n]$,

$$g^{(\alpha)}_i = \alpha g_i + (1-\alpha) g'_i = \alpha \sum_S p_S\, r_{S,i} + (1-\alpha) \sum_S p_S\, r'_{S,i} = \sum_S p_S\, r^{(\alpha)}_{S,i}.$$

The following gives a test to check whether a given allocation vector g is feasible or not. While the test is not efficient as it checks for exponentially many inequalities, it will be useful for our proofs later. For $S \subseteq [n]$, let $\mathcal{E}(S) = \{T \subseteq [n] : T \cap S \neq \emptyset\}$ be the collection of sets with non-empty intersection with S.

Lemma 1. *$g \in G(\mathcal{P})$ iff for all $S \subseteq [n]$, $P(\mathcal{E}(S)) \geq \sum_{i \in S} g_i$, where $P(\mathcal{E}(S)) = \sum_{T \in \mathcal{E}(S)} p_T$.*

Proof. For a goal vector g, consider the flow network $F(\mathcal{P}, g) = (V, E)$, with vertices $V = \{s\} \cup \mathrm{supp}(\mathcal{P}) \cup [n] \cup \{t\}$, and edges $E = E_1 \cup E_2 \cup E_3$ with capacity function c, where $E_1 = \{(s, S) : S \in \mathrm{supp}(\mathcal{P})\}$, $c(s, S) = p_S$, $E_2 = \{(i, t) : i \in [n]\}$, $c(i, t) = g_i$ and $E_3 = \{(S, i) : i \in S\}$, $c(S, i) = \infty$.

We claim that g is achievable iff there exists an s-t flow of value $\sum_{i \in N[n]} g_i$. Indeed, given a CR scheme given by $r_{S,i}$ that achieves g, the flow f with $f(s, S) = \sum_{i \in S} r_{S,i} p_S$ for $S \in \mathrm{supp}(\mathcal{P})$, $f(S, i) = r_{S,i} p_S$ for $i \in S$, and $f(i, t) = g_i$ for $i \in [n]$, is feasible and has value $\sum_{i \in N} g_i$. Conversely, given a flow f with value $\sum_i g_i$, setting $r_{S,i} = f(S, i)/p_S$ would achieve g.

Applying max-flow min-cut theorem to this network directly gives the result.

2 Weight-based schemes

Recall that in a weight based scheme, $r_{S,i} = w_i / (\sum_{j \in S} w_j)$, for each set S and each $i \in S$.

We now prove Theorems 1 and 2. For clarity of exposition, we first focus on showing the existence of weights and assume that $\mathcal{P}$ is given explicitly. The algorithmic version follows from known results for max-entropy convex programs and matrix scaling, and standard ideas based on sampling that are described in Section 2.3.

2.1 Convex Programming based Algorithm

Consider the following convex program with variables $x_{S,i} \geq 0$ for the probability that player $i \in S$ receives the item on outcome S.

$$\max \sum_{S,i} x_{S,i} \ln \frac{e}{x_{S,i}}$$

$$s.t. \sum_{S:i\in S} x_{S,i} = g_i \qquad \forall i \in [n] \tag{2}$$

$$\sum_{i\in S} x_{S,i} = p_S \qquad \forall S \in \mathrm{supp}(\mathcal{P}) \tag{3}$$

$$x_{S,i} \geq 0 \qquad \forall S \in \mathrm{supp}(\mathcal{P}),\ i \in S \tag{4}$$

As $g \in \mathrm{int}(G^M)$, $g_i > 0$ for each i, and as the objective function is strictly convex, the optimum solution x^* to the program satisfies $x^*_{S,i} > 0$ for all i, S. As $g \in \mathrm{int}(G^M)$, the program is feasible and also satisfies Slater's condition [5], and by strong duality there is a tight dual solution with the same objective value.

Dual Let us consider the dual. It has variables $\beta_i \in \mathbb{R}$ for each $i \in [n]$ for the constraints (2), and $\alpha_S \in \mathbb{R}$ for each $S \in \mathrm{supp}(\mathcal{P})$ for the constraints (3). Writing the objective as $\sum_{S,i} x_{S,i}(1-\ln x_{s,i})$, its partial derivative with respect to $x_{S,i}$ is $(1-\ln x_{S,i})-1 = -\ln x_{S,i}$. As $x^*_{S,i} > 0$ in the primal, by complementary slackness the dual variables for constraints (4) have value 0 and the KKT conditions give that,

$$\beta_i - \alpha_S = \ln x_{S,i} \qquad \forall S, i. \tag{5}$$

Or equivalently, $x_{S,i} = e^{\beta_i}/e^{\alpha_S}$. For a fixed S, summing up (5) over $i \in S$ and using $\sum_{i\in S} x_{i,S} = p_S$ gives $\sum_{i\in S} e^{\beta_i} = e^{\alpha_S} p_S$ and hence

$$x_{S,i} = p_S e^{\beta_i} / (\sum_{i\in S} e^{\beta_i}).$$

As $r_{S,i} = x_{S,i}/p_S$, this gives that $r_{S,i}$ has the form $w_i/(\sum_{i\in S} w_i)$ for each s, i with $w_i = e^{\beta_i}$.

Running time and accuracy The condition $g \in \mathrm{int}(G^M)$ ensures that g has distance at least ϵ from G^M for some $\epsilon > 0$. The arguments in [24] on the bit-length of solutions for max-entropy convex programs, imply that the bit length of w_i depends as $O(\log 1/\varepsilon)$ on ε and the convex program runs in time polynomial in n, $\mathrm{supp}(\mathcal{P})$ and $\log(1/\varepsilon)$.

2.2 Matrix Scaling based Algorithm

We now show how to obtain Theorem 1 using matrix scaling. We begin by briefly describing the relevant background on matrix scaling.

Let A be a given $m \times n$ matrix with non-negative entries, and let $r \in (\mathbb{R}^+)^m$ and $c \in (\mathbb{R}^+)^n$ be positive vectors satisfying $\sum_{i\in[m]} r_i = \sum_{j\in[n]} c_j$.

Definition 1 (**(r,c)-scalable).** *A non-negative matrix A is said to be (r,c)-scalable, if there exists non-negative vectors $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$ such that $B = \mathrm{Diag}(x) A\, \mathrm{Diag}(y)$ with entries $b_{ij} = x_i a_{ij} y_j$ has row sums r and column sums c. That is, $\sum_j b_{ij} = r_i$ for $i \in [m]$ and $\sum_i b_{ij} = c_j$ for $j \in [n]$.*

Given an $\epsilon > 0$, we say that A is ε-(r,c) scalable if there is some scaling of A with row sums (exactly) r, and columns sums c' satisfying $\sum_{j=1}^{n}(c'_j - c_j)^2 \leq \varepsilon$. We say that A is *almost* (r,c)-scalable if it is ϵ-(r,c) scalable for any $\epsilon > 0$.

The following proposition from [23] exactly characterizes approximate scalability for a matrix.

Proposition 1. *A non-negative matrix A is almost (r,c)-scalable iff for every zero minor $Z \times L$ of A (i.e., a submatrix of A with all zero entries), the vectors r and c satisfy $\sum_{i\in[n]\setminus Z} r_i \geq \sum_{j\in L} c_j$.*

Relation to weight-based CR schemes We now relate matrix scaling to contention resolution.

Given $\mathcal{P}$, let A be a $|\mathrm{supp}(\mathcal{P})| \times n$, incidence matrix for the support of $\mathcal{P}$, defined as $A_{S,i} = 1$ if $i \in S$ and 0 otherwise. The key observation is the following.

Lemma 2. *Let $g \in G^M(\mathcal{P})$, and suppose there exist vectors x and y such that the matrix $\mathrm{diag}(x) A\, \mathrm{diag}(y)$ has row sums p_S for each S and column sums g_j for each j. Then the vector y gives the weights for a weight-based CR scheme that achieves g.*

Proof. By the definition of A, and the properties of the scaling, the sum for row S satisfies

$$p_S = \sum_{j\in[n]} x_S A_{S,j} y_j = \sum_{j\in S} x_S y_j,$$

and hence $x_S = p_S/(\sum_{j\in S} y_j)$. Similarly, the sum for each column j satisfies

$$g_j = \sum_{S} x_S A_{S,j} y_j = \sum_{S:j\in S} x_S y_j = \sum_{S:j\in S} p_S \frac{y_j}{\sum_{j\in S} y_j}.$$

So setting $r_{S,j} = y_j/(\sum_{j\in S} y_j)$ achieves $g_j = \sum_S p_S r_{S,j}$ and $\sum_{j\in S} r_{S,j} = 1$ for all S, giving the claimed weight-based scheme.

Interestingly, the condition for the existence of the scaling of A in Lemma 2, turns out to be identical to the condition for feasibility of a goal vector $g \in G^M(\mathcal{P})$ as in Lemma 1.

Lemma 3. *For any $\mathcal{P}$, the incidence matrix A is almost (p_S, g)-scalable iff $g \in G^M$.*

Proof. We will show that the condition in Proposition 1 holds for each zero minor $Z \times L$ of A. Fix some $L \subset [n]$. Then it suffices to consider the maximal $Z \times L$ minor of A, which corresponds to $Z = [n] \setminus \mathcal{E}(L)$, the collection all subsets disjoint from L.

For $r = \{p_S\}_S$ and $c = g$, the condition $\sum_{i\in[n]\setminus Z} r_i \geq \sum_{j\in L} c_j$ thus becomes $\sum_{T\in\mathcal{E}(L)} p_T \geq \sum_{i\in L} g_i$, which holds by Lemma 1 for every $L \subseteq [n]$ iff g is achievable.

It is also known that if A is almost (r, c)-scalable, then an ϵ-(r, c) scaling can be found after poly$(n, \log(1/\varepsilon))$ iterations [9], which gives the claimed running time.

2.3 Handling g at the boundary and Sampling

Interestingly, the matrix-scaling based algorithm is quite robust and works even if g lies at the boundary of $G^M(\mathcal{P})$. In particular, as the algorithm iteratively computes some row and column scaling x and y, using y as weights at any intermediate step gives some valid and achievable allocation. More precisely, the weights y after poly$(n, \log 1/\varepsilon)$ iterations will correspond to a goal vector $\hat{g}$ with $\|g - \hat{g}\|_2 \leq \varepsilon$ [9], even for g on the boundary of G^M. The algorithms above assumed access to the entire distribution $\mathcal{P}$. This can be removed using standard arguments that we now sketch. Let q_i be the marginal probability of player i in $\mathcal{P}$. By standard tail bounds, for any $\mathcal{P}$, each q_i can be estimated to within $\pm\varepsilon$ error with high probability, using $O((\log n)/\epsilon^2)$ independent samples from $\mathcal{P}$.

Suppose g is achievable for $\mathcal{P}$ by some CR scheme R given by $r_{S,i}$, i.e., $g_i = \sum_S p_S\, r_{S,i}$. Take $m = O(\log n/\varepsilon^2)$ samples $\hat{S}$ from $\mathcal{P}$, and consider the (empirical) allocation $\hat{g}_i = \frac{1}{m}\sum_{\hat{S}} r_{\hat{S},i}$ obtained by R on these samples. As $0 \leq r_{S.i} \leq 1$, and as $|p_i - \sum_{\hat{S}} \mathbf{1}(i \in \hat{S})/m| \leq \varepsilon$ for all i, this gives that $\|\hat{g} - g\|_\infty \leq \varepsilon$ with high probability. As $\hat{g}$ is achievable for the sampled distribution $\hat{S}$, running the convex program in proof of Theorem 1 on $\hat{S}$ with goal g (and by using an additional error term ϵ in (2)) will find some $\tilde{g}$, with $\|g - \tilde{g}\|_\infty \leq \|g - \hat{g}\|_\infty + \|\hat{g} - \tilde{g}\| \leq 2\varepsilon$. In particular, this implies the following.

Theorem 8. *There are efficient algorithms, using only sampling access to $\mathcal{P}$, that given g, find an achievable $\hat{g}$ satisfying $\|g - \hat{g}\| \leq \varepsilon$ or certify that no such $\hat{g}$ exists and run in time poly$(n, 1/\varepsilon)$.*

3 Permutation Schemes and Achievable Goal Vectors

Recall that a given a permutation π of the players, the corresponding permutation scheme allocates the item to $i \in S$ appearing earliest in the ordering π, i.e., for every $S \neq \emptyset$, $r^\pi_{S,i} = 1$ for $i = \arg\min\{\pi(j) : j \in S\}$, and $r_{S,j}(\pi) = 0$ otherwise.

Let Π_n denote the set of all $n!$ permutations on $[n]$. Given a distribution $\mathcal{P}$ on subsets of n players, and a permutation $\pi \in \Pi_n$, let $g(\pi, \mathcal{P})$ denote allocation vector achieved by the permutation scheme given by π.

Theorem 9. *For any $\mathcal{P}$, any maximal achievable allocation $g \in int(G^M(\mathcal{P}))$ is a convex combination of $g(\pi, \mathcal{P})$ for $\pi \in \Pi_n$. Equivalently, the closure of $G^M(\mathcal{P})$ is the convex hull of the vectors $g(\pi, \mathcal{P})$.*

Proof. As all quantities depend on $\mathcal{P}$, we drop it to simplify notation. Fix some $g \in \mathrm{int}(G^M)$. It suffices to show that g is some convex combination of $g(\pi)$ for permutations π in Π_n, i.e., $g = \sum_{\pi \in S_n} \alpha(\pi)\, g(\pi)$ with $\sum_{\pi \in S_n} \alpha(\pi) = 1$ and $\alpha(\pi) \geq 0$ for all π.

As $g \in \mathrm{int}(G^M)$, by Theorem 1, g can be achieved by a weight-based scheme given by some weight vector w, i.e., $g_i = \sum_{i \ni S} p_S\, w_i/(\sum_{j\in S} w_j)$ for all $i \in [n]$.

We define $\alpha(\pi)$ using w as follows. Equip each player i with an independent exponentially distributed random variable X_i with rate $\lambda_i = w_i$ (i.e., with mean $1/w_i$). For each instantiation of these variables, let π be the ordering of the players in the increasing of their X_i values i.e., $X_{\pi^{-1}(1)} < X_{\pi^{-1}(2)} < \ldots < X_{\pi^{-1}(n)}$ (we ignore ties, as this is a measure zero event). Let α_π denote the probability for the permutations $\pi \in \Pi_n$, and let α denote the resulting distribution on the permutations.

By the memoryless property of exponential random variables, for any set S of players, the probability that player i has minimum X_i value among all the X_j for $j \in S$ is exactly $w_i/\sum_{j\in S} w_j$. In other words,

$$\mathrm{Pr}_{\pi\sim\alpha}[\arg\min\{\pi(j) : j \in S\} = i] = \frac{w_i}{\sum_{j\in S} w_j}.$$

This implies that choosing π according to the distribution α and applying the permutation scheme based on π exactly achieves the goal g.

While the proof may use g exponentially many permutations to write g as a convex combination, as $g \in \mathbb{R}^n$, by Caratheodory's theorem at most $n+1$ vectors $g(\pi)$ always suffice.

Tightness. We remark that there exist distributions $\mathcal{P}$ for which each of the $n!$ permutations π actually corresponds to a unique extreme point $g(\pi)$ of $G^M(\mathcal{P})$. Consider the product distribution $\mathcal{P}$ with marginals $q_1 = \ldots = q_n = 1/2$. For any permutation π, the player $\pi^{-1}(k)$, which is at position k in the ordering, gets the item with probability exactly 2^{-k}. So $g(\pi) = (2^{-\pi(1)}, \ldots 2^{-\pi(n)})$.

For any π, we claim that $g(\pi)$ cannot be expressed as a convex combination of other $g(\pi')$. Without loss of generality, suppose π is the identity permutation. Then, as $g(1) = 1/2$ and the marginal $q_1 = 1/2$, every permutation π' in the support of the convex combination must have 1 as its first element. Conditioning out the element 1, and applying the argument repeatedly gives that the only permutation in the support of $g(\pi)$ is $(1, 2, \ldots, n)$.

3.1 Alternate Feasibility Test

The characterization above gives another useful test for testing if g is feasible, provided the distribution $\mathcal{P}$ has the property that the goal vector $g(\pi)$ can be efficiently computed for any permutation π. This is true for most natural distributions, or in general $g(\pi)$ can be computed efficiently to desired accuracy by sampling. E.g., for the product distribution $\mathcal{P} = \langle q_1, \ldots, q_n\rangle$, we have the have the explicit expression $g_k(\pi) = q_k \prod_{j:\pi(j)<\pi(k)} (1 - q_j)$.

Theorem 10. *If $\mathcal{P}$ has the property that $g(\pi)$ can be computed for any permutation π, then for any g there is an efficient algorithm to test if $g \in int(G^M(P))$. The algorithm also computes a corresponding CR scheme.*

The proof is based on linear programming. Given a goal vector g, consider the following LP with (exponentially many) variables x_π for each permutation $\pi \in \Pi_n$.

$$\min \sum_{\pi \in \Pi_n} x_\pi \qquad \text{s.t.} \qquad \sum_{\pi \in \Pi_n} x_\pi g_i(\pi) \geq g_i, \quad \forall i \in [n], \qquad x_\pi \geq 0, \quad \forall \pi \in \Pi_n.$$

Then g is achievable iff the objective $\sum_{\pi \in \Pi_n} x_\pi \leq 1$. As this LP has exponentially many variables, let us consider the following dual with variables y_i for $i \in [n]$.

$$\max \sum_{i \in [n]} g_i \, y_i \qquad \text{s.t.} \qquad \sum_{i \in [n]} y_i \, g_i(\pi) \leq 1, \quad \forall \pi \in \Pi_n, \qquad y_i \geq 0, \quad \forall i \in [n].$$

While the dual has exponentially many constraints, it can be solved efficiently using a *separation oracle*. Recall that for the separation oracle, given a candidate feasible solution y we need to find some permutation π such that $\sum_{i\in[n]} y_i \, g_i(\pi) > 1$, provided such a permutation exists. The following shows that the such a permutation π is easily obtained by sorting the coordinates of y.

Claim 4 *Given y, let σ be the permutation such that $y_{\sigma(1)} \geq \ldots \geq y_{\sigma(n)}$. Then for any $\mathcal{P}$, the quantity $\sum_i y_i g_i(\pi)$ is maximized for $\pi = \sigma^{-1}$. That is $\sigma^{-1} = \arg\max_\pi(\sum_{i\in[n]} y_i g_i(\pi))$.*

Proof. Let π be some permutation that maximizes $\sum_{i\in[n]} y_i \, g_i(\pi)$ and $\sum_{i\in[n]} y_i \, g_i(\pi) > \sum_{i\in[n]} y_i \, g_i(\sigma^{-1})$. Then $\pi^{-1}(k)$ is the index of player that appears at position k in the ordering π. Suppose for the sake of contradiction that $y_{\pi^{-1}(k)} < y_{\pi^{-1}(k+1)}$. Then, consider the ordering obtained by swapping the players at positions k and $k+1$, i.e. $\pi'(\pi^{-1}(k)) = k+1$ and $\pi'(\pi^{-1}(k+1)) = k$ and $\pi'(i) = \pi(i)$ otherwise. We will show that $(\sum_{i\in[n]} y_i g_i(\pi'))$ is not smaller than $(\sum_{i\in[n]} y_i g_i(\pi))$, giving the desired contradiction. Let

$$\tilde{\mathcal{S}} = \{S \subseteq N : \ \pi^{-1}(k), \pi^{-1}(k+1) \in S, \ k = \min_i \{\pi(i) : i \in S\}\}$$

be the collection of sets containing both $\pi^{-1}(k), \pi^{-1}(k+1)$ and where the player $\pi^{-1}(k)$ is the earliest player in S according to the ordering in π.

The crucial observation is that for any S and $i \in S$, we have $r_{S,i}(\pi) = r_{S,i}(\pi')$, unless $S \in \tilde{\mathcal{S}}$ and $i \in \{\pi^{-1}(k), \pi^{-1}(k+1)\}$. Moreover, in this case, $r_{S,\pi^{-1}(k)}(\pi) = 1$, $r_{S,\pi^{-1}(k+1)}(\pi) = 0$, and $r_{S,\pi^{-1}(k)}(\pi') = 0$, $r_{S,\pi^{-1}(k+1)}(\pi') = 1$. This gives that

$$\sum_{i\in[n]} y_i g_i(\pi') - \sum_{i\in[n]} y_i g_i(\pi) = \sum_{S \in \tilde{\mathcal{S}}} p_S \cdot (y_{\pi^{-1}(k+1)} - y_{\pi^{-1}(k)})$$

which is non-negative by our assumption.

4 Sequential schemes for the FV setting

Recall that a sequential scheme has the following form. There is some fixed order π on the players, and we compute $\gamma_i \in [0,1]$ for each i. When S is realized, we go over the players in the order given by π, and give item to $i \in S$ with probability γ_i, unless i is the last player in S, in which it gets the item with probability 1.

Clearly, this scheme is maximal. We show that it can achieve the same allocation as the FV scheme for any production distribution and for any order on players π.

Theorem 11. *For any product distribution $\mathcal{P} = \langle q_1, \dots, q_n \rangle$, and for any ordering π of the players, there exists a sequential scheme, that achieves the maximal fair allocation. Moreover, the γ_i are explicitly given as $\gamma_i = (\alpha_i - R_{i+1})/(1 - R_{i+1})$, where $R_k = \prod_{i \ge k}(1-q_i)$, $\alpha_k = (1-R_k)/Q_k$ and $Q_k = \sum_{i\ge k} q_i$.*

Proof. Without loss of generality, we can assume π is the identity permutation, and the players are considered in the order $1,\dots,n$. Also, given the definition of α_k and R_k, the fairness factor $\alpha = (1 - \prod_{i=1}^n(1-q_i))/(\sum_{i=1}^n q_i)$ is simply $(1-R_1)/Q_1 = \alpha_1$. We will show that the desired $g = \alpha q$ is attained by setting $\gamma_k = (\alpha_k - R_{k+1})/(1-R_{k+1})$ for $k \in [n]$.

We first show that γ_k are well-defined probabilities. Let us recall the Weierstrass' product inequalities.

Fact 5 *Let $a_1, ..., a_n \in [0,1]$ and $S = \sum_i a_i$. Then $\prod_i(1-a_i) \ge 1 - S$ and $\prod_i(1+a_i) \ge 1 + S$.*

As $\alpha_k \le 1$, we clearly have $\gamma_k \le 1$. To show that $\gamma_k \ge 0$, note that the denominator is non-negative, and the numerator satisfies

$$\begin{aligned}\alpha_k - R_{k+1} &= \frac{1-R_k}{Q_k} - R_{k+1} = \frac{1}{Q_k}\Big(1-(1-q_k)R_{k+1} - Q_k R_{k+1}\Big)\\ &= \frac{1}{Q_k}(1-(1+Q_{k+1})R_{k+1}) \ge 0,\end{aligned}$$

where the second equality uses that $R_k = (1-q_k)R_{k+1}$ and the third equality uses that $Q_k - q_k = Q_{k+1}$. The inequality follows, as by Fact 5, $1 + Q_{k+1} \le \prod_{i\ge k+1}(1+q_i)$, and hence $(1+Q_{k+1})R_{k+1} \le \prod_{i \ge k+1}(1-q_i^2) \le 1$.

For $i \in [n]$, let $p^{(i)}$ denote the product distribution $\langle q_i, \dots, q_n\rangle$ on $i, \dots, n$. That is, for a set $T \subseteq [i,n]$, $p_T^{(i)} = \prod_{j\in T} q_j \prod_{j \ge i, j \notin T}(1-q_j)$.

Given $\gamma_1, \dots, \gamma_n$, we define $\tilde{p}_T^{(i)}$ for $i \in [n], T \subseteq [i,n], T \neq \emptyset$, as the probability that each player in T requested the item and that the item is not allocated to any of the players in $[1, i-1]$ in the sequential scheme.

Claim. For any non-empty subset T of $\{i,\dots,n\}$, it holds that $\tilde{p}_T^{(i)} = (\alpha_1/\alpha_i)p_T^{(i)}$.

Proof. We first note that $\tilde{p}_T^{(i)} = \prod_{j=1}^{i-1}\Big((1-q_j) + q_j(1-\gamma_j)\Big)p_T^{(i)} = \prod_{j=1}^{i-1}(1 - q_j\gamma_j)p_T^{(i)}$.

This follows as the set T is 'left over' at step i, if and only if, for each player $j \in \{1, \ldots, i-1\}$, either it did not appear in S or it did not pick the item which happens with probability $1-\gamma_j$ as T is non-empty and hence j was not the last player in S.

So it suffices to show that $1 - q_j\gamma_j = \alpha_j/\alpha_{j+1}$ for each $j \in \{1, \ldots, i-1\}$. To this end, we have

$$\begin{aligned} 1 - q_j\gamma_j &= 1 - \frac{q_j(\alpha_j - R_{j+1})}{1 - R_{j+1}} = \frac{1 - R_{j+1} - q_j\alpha_j + q_jR_{j+1}}{1 - R_{j+1}} \\ &= \frac{1 - R_j - q_j\alpha_j}{1 - R_{j+1}} = \frac{\alpha_j(Q_j - q_j)}{1 - R_{j+1}} = \frac{\alpha_j Q_{j+1}}{1 - R_{j+1}} = \frac{\alpha_j}{\alpha_{j+1}}, \end{aligned}$$

where the third equality uses that $R_j = (1-q_j)R_{j+1}$, and the fourth (resp. sixth) that $1 - R_j = \alpha_j Q_j$ (resp. $1 - R_{j+1} = \alpha_{j+1}Q_{j+1}$).

We can express probability of player i getting the item as follows.

Claim. The probability the player i gets the item in the sequential scheme is

$$g_i = \tilde{p}^{(i)}_{\{i\}} + \gamma_i \sum_{T \subseteq \{i,\ldots,n\}, i \in T, T \neq \{i\}} \tilde{p}^{(i)}_T.$$

Proof. Consider the set of players T left after the first $i-1$ steps. Then either player i gets the item with probability 1 if $T = \{i\}$, otherwise it gets it with probability γ_i if $i \in T$ and $j \in T$ for some $j > i$.

Noting that $p^{(i)}_{\{i\}} = q_iR_{i+1}$ and that $\sum_{T \subseteq \{i,\ldots,n\}, i \in T, T \neq \{i\}} p^{(i)}_T = q_i(1 - R_{i+1})$, by Claim 4 and Claim 4,

$$g_k = \frac{\alpha_1}{\alpha_k} \cdot (q_kR_{k+1} + \gamma_k q_k(1 - R_{k+1})) = \frac{\alpha_1}{\alpha_k} \cdot (q_kR_{k+1} + q_k(\alpha_k - R_{k+1})) = \alpha_1 q_k.$$

Noting that $\alpha_1 = \alpha$ gives the desired result.

4.1 Limitations

Unfortunately, sequential schemes are not general enough to work for arbitrary $\mathcal{P}$, or even when $\mathcal{P}$ is a product distribution but g is arbitrary. For $n = 3$, there exists a general distribution ($\Pr[A = \{1\}] = \Pr[A = \{1,2,3\}] = 0.34$, $\Pr[A = \{2\}] = \Pr[A = \{1,2\}] = 0.16$) such that a fair allocation that is achievable, but it is not achievable by a sequential scheme (for the order $1,2,3$). In addition, there exists a product distribution and achievable vector ($\mathcal{P} = \langle 0.4, 0.1, 0.2 \rangle$, $g = (0.356, 0.077, 0.135)$) which cannot be achieved by a sequential scheme (for the order $1,2,3$). Understanding the class of allocations g, and the class of distributions $\mathcal{P}$ for which such schemes work might be an interesting question for further investigation.

5 Convex Program for the FV Scheme

Recall that the FV scheme considers the product distribution $\mathcal{P} = \langle q_1, \ldots, q_n \rangle$ with $g = \alpha q$ where $\alpha = (1 - \prod_i (1 - q_i))/(\sum_i q_i)$. Consider the following program with variables $r_{S,i}$, that tries to finds a rule $r_{S,i}$ to minimize the (weighted) quadratic variation about $1/|S|$, while satisfying the global constraints.

$$\min_r \sum_{i \in S} (|S| - 1)\, p_S \left(r_{S,i} - \frac{1}{|S|} \right)^2$$

$$\text{s.t.} \quad \sum_{S \ni i} p_S\, r_{S,i} = \alpha\, q_i \qquad \forall i \in [n] \tag{6}$$

$$\sum_{i \in S} r_{S,i} = 1 \qquad \forall S \tag{7}$$

$$r_{S,i} \geq 0 \qquad \forall S, \forall i \in S \tag{8}$$

Theorem 12. *For product distributions, the optimum solution to the program above is the FV scheme* (1).

Proof. We first note that after rearranging the terms, the FV scheme can be written as

$$r_{S,i} = \frac{1}{|S|} + \frac{(\overline{q}_S - q_i)}{(|S| - 1)(\sum_{i=1}^n q_i)} \qquad \text{for} |S| > 1, i \in S \tag{9}$$

where $\overline{q}_S = \sum_{i \in S} q_i / |S|$. Moreover, $r_{S,i} = 1$ for $|S| = 1$.

We wish to show that this is the optimum solution to the convex program. To do this, we construct a feasible dual solution and show that the primal-dual pair satisfies the KKT conditions, and use strong duality for convex programs (and that Slater's condition is satisfied).

Let us define the dual variables $\beta_i \in \mathbb{R}$ for $i \in [n]$ for constraints (6), $\gamma_S \in \mathbb{R}$ for all S in (7), and $\delta_{S,i} \geq 0$ for (8). The complementary slackness conditions for (8) give $r_{s,i}\delta_{i,S} = 0$. Taking the Lagrangian and the partial derivatives with respect to $r_{S,i}$ gives the condition

$$2(|S| - 1)\; p_S \cdot (r_{S,i} - 1/|S|) - \gamma_S - \beta_i\, p_S - \delta_{i,S} = 0 \qquad \forall S, \forall i \in S. \tag{10}$$

Consider the following dual solution:

$$\beta_i = -2 q_i / (\sum_{i=1}^n q_i), \quad \gamma_S = (\sum_{i \in S} \beta_i)\, p_S / |S| = -2\overline{q}\, p_S / (\sum_{i=1}^n q_i) \quad \text{and} \quad \delta_{S,i} = 0.$$

We show that this primal-dual pair satisfies the KKT conditions. First, $r_{s,i}\delta_{i,S} = 0$ holds trivially as $\delta_{S,i} = 0$. So (10) becomes

$$2(|S| - 1)\; p_S \cdot (r_{S,i} - 1/|S|) - \gamma_S - \beta_i\, p_S = 0. \tag{11}$$

For S with $|S| = 1$, this holds easily as the first term above becomes 0, and our choice satisfies $\gamma_S = -\beta_i p_S$ for $S = \{i\}$.

For S with $|S| > 1$, plugging the values of γ_S and β_i, cancelling p_S and re-arranging, (11) simplifies to

$$r_{S,i} = \frac{1}{S} + \frac{1}{2(|S|-1)}(\beta_i - \gamma_S) = \frac{(\overline{q}_S - q_i)}{(|S|-1)(\sum_{i=1}^n q_i)},$$

which exactly corresponds to the FV scheme. The primal feasibility of $r_{S,i}$ follows from the (somewhat tedious) calculations in [11], which show that $r_{S,i}$ is a valid and maximally fair scheme.

6 Extending the Variance Minimization Program

More generally, for any distribution $\mathcal{P}$ and any goal vector g, one can consider a more general family of convex programs with arbitrary weights w_S for set S in the objective (the program for the FV scheme has $w_S = |S| - 1$).

$$\min \sum_{i \in S} w_S\, p_S\, (r_{S,i} - 1/|S|)^2 \qquad \text{s.t.} \qquad \sum_{S \ni i} p_S \cdot r_{S,i} = g_i, \forall i; \qquad \sum_{i \in S} r_{S,i} = 1, \forall S.$$

Suppose there is some natural choice of weights w_S (possibly depending of $\mathcal{P}, g$), so that the constraints $r_{S,i} \geq 0$ were automatically satisfied by the optimum solution to this convex program. Then, we claim that the resulting CR rule given by $r_{S,i}$ has a very succinct representation, similar to FV scheme.

Dual. As before, consider the dual of this program with dual variables $\beta_i, \gamma_S \in \mathbb{R}$. Then taking the Lagrangian and partial derivatives, gives the KKT conditions:

$$2 w_S\, p_S\, (r_{S,i} - \frac{1}{|S|}) - \gamma_S - \beta_i\, p_S = 0, \forall S, i.$$

For a fixed S, summing this up over all $i \in S$ and using that $\sum_{i \in S} r_{S,i} = 1$, the first term becomes 0 and we get $\gamma_S = -\sum_{i \in S} \beta_i p_S / |S|$. Let us denote $\overline{\beta}_S = \sum_{i \in S} \beta_i / |S|$. Then, this gives that

$$r_{S,i} = \frac{1}{|S|} + \frac{\beta_i - \overline{\beta}_S}{2 w_S}. \tag{12}$$

Let us call a CR scheme a β-scheme, if there is some natural choice of weights w_S, such that the optimum solution to the convex program has the form above. By our discussions above, the FV scheme is a β-scheme for $w_S = |S| - 1$ with $\beta_i = -2q_i / (\sum_{i=1}^n q_i)$. It is an interesting question to explore if β-schemes exists for more general distributions or even for more general allocation vectors and product distributions. The following two examples show that this is not true for the choice $w_S = |S| - 1$, that worked for the FV scheme.

Computing β efficiently. We remark that if a β-scheme exists, for some choice of w_S, then given any goal vector g, the corresponding $\beta_i(g)$ can be computed in time polynomial in n, assuming suitable access to $\mathcal{P}$. In particular, as $g_i =$

$\sum_{S\ni i} r_{S,i}\, p_S$, and $r_{S,i}$ is given by (12), g is linear in β and hence $g = J\beta + v_0$ where v_0 is some affine shift and J is the $n \times n$ Jacobian matrix with entries $J_{ij} = \partial g_i/\partial \beta_j$. Suppose J (the entries of which only depend on $\mathcal{P}$) and w_S can be computed explicitly, and that the goal vector $g(0)$ for $\beta = \mathbf{0}$ (that has entries $g_i(0) = \sum_{S\ni i} p_S/|S|$) can be computed. Then $\beta(g) = J^{+}(g - g(0))$, where J^{+} is the pseudoinverse of J.
Limitations. For $n = 3$, there exists a general distribution ($\Pr[\{1\}] = 0.55$, $\Pr[\{3\}] = 0.2$, $\Pr[\{1,3\}] = 0.05$, $\Pr[\{1,2,3\}] = 0.25$) such that a a fair allocation is achievable, but is not achievable using a β-scheme with the choice of weights $w_S = |S| - 1$. And there exists a product distributions and an achievable goal vector g ($\mathcal{P} = \langle 0.28, 0.59, 0.52\rangle$, and $g = (0.25437, 0.29448, 0.309446)$) which cannot be achieved using a β-scheme with the choice of weights $w_S = |S| - 1$.

7 Computing the Max-min Allocation

Recall that given $\mathcal{P}, V$ our goal is to compute a feasible $g \in G(\mathcal{P})$, such that for any $g' \in G(\mathcal{P})$ we have $g \succeq_V g'$. In other words, the vector $\alpha^V(g)$ with entries g_i/v_i satisfies that $\alpha^V(g)^{\uparrow}$ is lexicographically the largest among all other feasible allocations g'.

For any set S, the maximum probability that the item is allocated to players in S is $\sum_{T:T\cap S\neq\emptyset} p_T$. So by averaging, under any allocaton g, some player $i \in S$ always has $\alpha_i^V(g) \leq (\sum_{T:T\cap S\neq\emptyset} p_T)/(\sum_{i\in S} v_i)$. The algorithm proceeds by finding the set S will the smallest such value, called the critical value, and ensures that g_i/v_i equals this value for players in S. It then iterates on the residual instance. Below, we show that the critical probabilities can only increase, and that the resulting allocation in the fairest with respect to V. Later we show how to compute the critical set S.

We omit the superscript V in α^V for convenience, whenever it is clear from the context. For $S \subseteq [n]$, let $V(S) = \sum_{i\in S} v_i$ and let $E^A(S) = \{T \subseteq A, T \cap S \neq \emptyset\}$ be the collection of subsets of A intersecting S. For a collection of sets $\mathcal{S}$, let $P(\mathcal{S}) = \sum_{S\in\mathcal{S}} p_S$. Consider the following algorithm.

Algorithm ComputeFair$(\mathcal{P}, V)$:

1. Init: $A_1 \leftarrow [n]$, $k \leftarrow 1$
2. while $A_k \neq \emptyset$
   (a) $S_k \leftarrow \arg\min_{S\subseteq A_k} \left\{\frac{P(E^{A_k}(S))}{V(S)}\right\}$, $\alpha_k \leftarrow \frac{P(E^{A_k}(S_k))}{V(S_k)}$
   (b) $g_i \leftarrow v_i \cdot \alpha_k$, for $i \in S_k$, $A_{k+1} \leftarrow A_k \setminus S_k$, $k \leftarrow k+1$

Theorem 13. *For any distribution $\mathcal{P}$ and for any priority vector V, the fairest allocation $g \in G(P)$ can be computed exactly in time poly$(n, |supp(\mathcal{P})|)$ using a liner program. If an ϵ additive error is allowed, the running time is poly$(n, 1/\epsilon)$. An exact computation in poly(n) time is also possible if $g(\pi)$ can be computed efficiently for any permutation π.*

First, we prove the correctness of the algorithm, let K be the number of iterations. The following simple claim shows that α_i can only increase over the iterations.

Claim. For all $i \in [1, K-1]$, $\alpha_i \leq \alpha_{i+1}$.

Proof. For the sake of contradiction, suppose that $\alpha_i > \alpha_{i+1}$ for some $i \in [K-1]$. Then for $S' = S_i \cup S_{i+1}$,

$$\frac{P(E^{A_i}(S'))}{V(S')} = \frac{P(E^{A_i}(S_i)) + P(E^{A_{i+1}}(S_{i+1}))}{V(S_i) + V(S_{i+1})} < \frac{P(E^{A_i}(S_i))}{V(S_i)} = \alpha_i,$$

where the first equality follows from $E^{A_i}(S_i) \cap E^{A_{i+1}}(S_{i+1}) = \emptyset$, and the inequality follows from $\frac{a+x}{b+y} < \frac{a}{b}$ if $\frac{x}{y} < \frac{a}{b}$, for $a, b, x, y > 0$. This contradicts that S_i is the critical subset at step i.

Next, we prove that the output of **ComputeFair**$(\mathcal{P}, V)$ is achievable.

Claim 6 *For any distribution $\mathcal{P}$ and priority V, the output g of* **ComputeFair**$(\mathcal{P}, V)$ *lies in $G(\mathcal{P})$.*

Proof (Proof of Claim 6). By Lemma 1, we need to show that for all S we have $P(\mathcal{E}(S)) \geq \sum_{i\in S} g_i$, where $\mathcal{E}(S) = \{T \subseteq [n] : T \cap S \neq \emptyset\}$. This follows as

$$\begin{aligned}\sum_{i\in S} g_i &= \sum_{j\in[K]} \sum_{i\in S\cap S_j} g_i = \sum_{j\in[K]} \sum_{i\in S\cap S_j} \alpha_j v_i \\ &\leq \sum_{j\in[K]} \sum_{i\in S\cap S_j} v_i \cdot \frac{P(E^{A_j}(S\cap S_j))}{V(S\cap S_j)} = \sum_{j\in[K]} P(E^{A_j}(S\cap S_j)) \leq P(\mathcal{E}(S)),\end{aligned}$$

where the first inequality follows from the minimality of S_j, and the second inequality follows from $E^{A_j}(S \cap S_j) \cap E^{A_{j'}}(S \cap S_{j'}) = \emptyset$, for $j \neq j'$.

Claim 7 *For any $\mathcal{P}$ and V, let g the output of* **ComputeFair**$(\mathcal{P}, V)$*, then $g \succeq_V g'$ for all $g' \in G(\mathcal{P})$.*

Proof. For the sake of contradiction, consider some $\mathcal{P}$,V and $g' \in G(\mathcal{P})$ such that $g' \succ_V g$, with the fewest number of players n. Let r' be the rule corresponding to g', i.e., $g'_i = \sum_{S\subseteq[n]} r'_{S,i} p_S$. Consider the following two cases depending on the first critical set S_1.

Case 1. $g'_i = g_i$ for all $i \in S_1$. In this case, we claim that for all $S \in E(S_1)$, we have $\sum_{i\in S_1} r'_{S,i} = 1$ as

$$\sum_{S\in E(S_1)} p_S \cdot \sum_{i\in S_1} r'_{S,i} = \sum_{i\in S_1} g'_i = \alpha_1 \cdot \sum_{i\in S_1} v_i = \sum_{S\in E(S_1)} p_S.$$

So, for all $S \in E(S_1)$, we have $\sum_{i\in S\setminus S_1} r'_{S,i} = 0$.

Consider the distribution $\hat{\mathcal{P}}, \hat{V}$ on players $A_2 = [n] \setminus S_1$, defined as $\hat{p}_S = p_S$ for $S \subseteq A_2$ and $\hat{v}_i = v_i$ for $i \in A_2$. Let $\hat{g}$ be the output of **ComputeFair**$(\hat{\mathcal{P}}, \hat{V})$, and note that by the design of the algorithm $\hat{g}_i = g_i$ for all $i \in A_2$. Moreover, let $\tilde{g}$ the vector which corresponds to the assignment rule $\tilde{r}$, where $\tilde{r}_{S,i} = r'_{S,i}$, for $S \subseteq A_2$. Since for all $S \in E(S_1)$ we have $\sum_{i \in S \setminus S_1} r'_{S,i} = 0$, therefore $\tilde{g}_i = g'_i$ for all $i \in A_2$. Hence, $\tilde{g} \succ_{\hat{V}} \hat{g}$ as well, with a smaller number of players, which contradicts the minimality of $\mathcal{P}$.

Case 2. $g'_j \neq g_j$ for some $j \in S_1$. Suppose $g'_i < g_i$ for some $i \in S_1$, then as $g'_i/v_i < g_i/v_i = \alpha_1$, and as α_k are non-decreasing by Claim 7, it follows that $g'_i/v_i < g_r/v_r$ for all $r \in [n]$, contradicting that $g' \succeq_V g$. Thus, we have $g'_j > g_j = \alpha_1 \cdot v_j$, and $g'_i \geq g_i = \alpha_1 \cdot v_i$ for all $i \in S_1$, but this is impossible as

$$\alpha_1 \cdot V(S_1) = \alpha_1 \cdot v_j + \sum_{i \in S_1 \setminus \{j\}} \alpha_1 \cdot v_i \leq \alpha_1 \cdot v_j + \sum_{i \in S_1 \setminus \{j\}} g'_i < \sum_{i \in S_1} g'_i \leq P(E(S_1)) = \alpha_1 \cdot V(S_1).$$

7.1 Computing the critical subset

We now give an algorithm that given $\mathcal{P}$, computes the critical subset of players. For clearer exposition, we assume that $\mathcal{P}$ is given explicitly and show that the algorithm runs in $\text{poly}(n, |\text{supp}(\mathcal{P})|)$. The sampling argument in Section 2 directly gives the ε-approximate version, which runs in time $\text{poly}(n, 1/\varepsilon)$.

The algorithm is based on a result of Charikar [6] for computing dense subgraphs in graphs, but for our setting we need to extend it to hypergraphs and to handle the weights v_i.

Consider the following LP with variables x_S for each set S in the support of $\mathcal{P}$ and y_i for $i \in [n]$.

$$\min \sum_S p_S x_S \qquad \text{s.t.} \qquad x_S \geq y_i, \forall i \in S; \qquad \sum_{i \in [n]} v_i y_i \geq 1; \qquad x_S, y_i \geq 0, \forall i, S. \tag{LPα}$$

Given a solution, let us define $T(r) = \{i \in [n] : y_i \geq r\}$. The algorithm solves LPα, and outputs $T(r^*)$ where $r^* = \arg\min_r \frac{P(E(T(r)))}{V(T(r))}$. Note that it suffices to only check for $r \in \{y_1, \ldots, y_n\}$.

Lemma 8. *The optimal value for LPα is $\min_T \frac{P(E(T))}{V(T)}$. The algorithm also outputs the critical set T.*

Proof. First, note that for any set T, there exists a feasible solution with value of $P(E(T))/V(T)$, by setting $y_i = 1/V(T)$ for all $i \in T$, and $x_S = 1/V(T)$ for all $S \in E(T)$, and all the other variables to 0. This satisfies all the constraints and has the claimed objective value.

Given a solution to LPα with value u, we show how to find an integral solution T with value u. First, we can assume that for all sets S we have $x_S = \max\{y_i : i \in S\}$. Consider a set T and a collection of sets $\mathcal{S}$ parameterized by $r \geq 0$ as follows: $T(r) = \{i : y_i \geq r\}$ and $\mathcal{S}(r) = \{S : x_S \geq r\}$.

We claim that $E(T(r)) = \mathcal{S}(r)$ for all $r \geq 0$. Fix some r. As $x_S \geq y_i$ for all $i \in S$, if $i \in T(r)$, then $S \in \mathcal{S}(r)$ for all S containing i. This implies that $\mathcal{S}(r)$

contains $E(T(r))$. Conversely, as $x_S = \max\{y_i : i \in S\}$, if $S \in \mathcal{S}(r)$ then there exists some $i \in S$ such that $y_i \geq r$ and hence $i \in T(r)$.

We claim that there exists some r such that $\frac{P(E(T(r)))}{V(T(r))} \leq u$. Indeed if not, then using $\int_0^\infty \mathbf{1}_{\{x \geq r\}} dr = x$, we have the following contradiction

$$u = \sum_S p_S\, x_S \int_0^\infty \Big(\sum_{S \in \mathcal{S}(r)} p_S\Big) dr > u \cdot \int_0^\infty \Big(\sum_{i \in T(r)} v_i\Big) dr = u\Big(\sum_i v_i y_i\Big) \geq u.$$

7.2 Computing the critical subset using an oracle

One can also compute the critical subset using oracle access to the allocation vectors $g(\pi)$ for permutations, using the linear program in Theorem 9, and thus avoid knowing $\mathcal{P}$ exactly. Consider the following LP.

$$\min \sum_{\pi \in \Pi} x_\pi \qquad \text{s.t.} \qquad \sum_{\pi \in \Pi} x_\pi g_i(\pi), \forall i \in [n]; \qquad x_\pi \geq 0, \forall \pi \in \Pi. \tag{LP-Perm-α}$$

The dual of this LP is the following.

$$\max \sum_{i \in [n]} v_i \cdot y_i \qquad \text{s.t.} \qquad \sum_{i \in [n]} y_i g_i(\pi) \leq 1, \forall \pi \in \Pi; \qquad y_i \geq 0, \forall i \in [n]. \tag{LP-α-Perm-Dual}$$

As before, let $T(r) = \{i \in [n] : y_i \geq r\}$. The algorithm solves LP-α-Perm-Dual by using a separation oracle as defined in Claim 4, and outputs $T(r^*)$ where $r^* = \arg\max_r \frac{V(T(r)}{P(E(T(r)))}$. As previously, it suffices to consider $r \in \{y_1, \ldots, y_n\}$.

Lemma 9. *The optimal value of the LP is* $\max_T \frac{V(T)}{P(E(T))}$. *The algorithm also outputs the critical set* T.

Proof. For any $S \subseteq [n]$ and $\pi \in \Pi$, note that $\sum_{i \in S} g_i(\pi) \leq P(E(S))$, as $P(E(S))$ is the total probability mass of all sets that contain some element of S. Moreover, for S is a prefix of π, $\sum_{i \in S} g_i(\pi) = P(E(S))$.

Let u the optimal value of LP-α-Perm-Dual, and let $T = \arg\max_S(V(S)/P(E(S)))$. First we show that $v \geq V(T)/P(E(T))$. Setting $y_i = 1/P(E(T))$, for any $\pi \in \Pi$, we have $\sum_{i \in [n]} y_i g_i(\pi) = (\sum_{i \in T} g_i(\pi))/P(E(T)) \leq 1$, and the objective is $(\sum_{i \in T} v_i)/P(E(T)) = V(T)/P(E(T))$. On the other hand, given a solution y, let π_y be order according to non-decreasing values of y (i.e., $y_{\pi_y(i)} \geq y_{\pi_y(i+1)}$ for all $i \in [n-1]$). Let $S(r) = \{i \in N : y_i \geq r\}$, note that $S(r)$ is a prefix of π_y for any r. By the constraint for π_y in the dual we have: $1 \geq \sum_{i \in [n]} y_i g_i(\pi_y) = \int_0^\infty \Big(\sum_{i \in S(r)} g_i(\pi_y)\Big) dr = \int_0^\infty P(E(S(r))) dr$ Now, an r such that $V(S(r))/P(E(S(r))) \geq u$ exists, otherwise we get the contradiction that

$$u = \sum_{i \in [n]} v_i \cdot y_i = \int_0^\infty V(S(r)) dr < u \int_0^\infty P(E(S(r))) dr \leq u.$$

References

1. Marek Adamczyk and Michał Włodarczyk. Random order contention resolution schemes. In *Foundations of Computer Science (FOCS)*, pages 790–801, 2018.
2. Nikhil Bansal, Aravind Srinivasan, and Ola Svensson. Lift-and-round to improve weighted completion time on unrelated machines. In *Symposium on Theory of Computing, STOC*, pages 156–167, 2016.
3. Nikhil Bansal and Maxim Sviridenko. The santa claus problem. In *Symposium on Theory of Computing, STOC*, pages 31–40. ACM, 2006.
4. D. Bertsekas and R. Gallagher. *Data Networks*. Prentice Hall, 1992.
5. Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.
6. Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *APPROX*, pages 84–95, 2000.
7. Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for unique games. In *Symposium on Theory of Computing, STOC*, pages 205–214, 2006.
8. Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM J. Comput.*, 43(6):1831–1879, 2014.
9. Michael B. Cohen, Aleksander Madry, Dimitris Tsipras, and Adrian Vladu. Matrix scaling and balancing via box constrained newton's method and interior point methods. In *Symposium on Foundations of Computer Science, FOCS*, pages 902–913, 2017.
10. Melvin Dresher. *Theory and Applications of Games of Strategy*. RAND Corporation, Santa Monica, CA, 1951.
11. Uriel Feige and Jan Vondrák. Approximation algorithms for allocation problems: Improving the factor of 1 - 1/e. In *Symposium on Foundations of Computer Science, FOCS*, pages 667–676, 2006.
12. Uriel Feige and Jan Vondrák. The submodular welfare problem with demand queries. *Theory of Computing*, 6(1):247–290, 2010.
13. Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes. In *Symposium on Discrete Algorithms, SODA*, pages 1014–1033, 2016.
14. David A. Hensher, John M. Rose, and William H. Greene. *Applied Choice Analysis*. Cambridge University Press, 2015.
15. Sungjin Im and Maryam Shadloo. Weighted completion time minimization for unrelated machines via iterative fair contention resolution. In *Symposium on Discrete Algorithms, SODA*, pages 2790–2809, 2020.
16. E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, 1957.
17. Narendra Karmarkar and Richard M. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. In *Foundations of Computer Science, FOCS*, pages 312–320, 1982.
18. F P Kelly, A K Maulloo, and D K Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3):237–252, 1998.
19. Elon Kohlberg. On the nucleolus of a characteristic function game. *SIAM Journal on Applied Mathematics*, 20(1):62–66, 1971.
20. Euiwoong Lee and Sahil Singla. Optimal online contention resolution schemes via ex-ante prophet inequalities. In *European Symposium on Algorithms, ESA*, volume 112 of *LIPIcs*, pages 57:1–57:14, 2018.

21. Nathan Linial, Alex Samorodnitsky, and Avi Wigderson. A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents. *Combinatorica*, 20(4):545–568, 2000.
22. R. D. Luce. *Individual Choice Behavior: A Theoretical Analysis.* Wiley & Sons, New York, 1959.
23. Uriel G Rothblum and Hans Schneider. Scalings of matrices which have prespecified row sums and column sums via optimization. *Linear Algebra and its Applications*, 114:737–764, 1989.
24. Mohit Singh and Nisheeth K. Vishnoi. Entropy, optimization and counting. In *Symposium on Theory of Computing, STOC*, pages 50–59, 2014.
25. Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statist.*, 35(2):876–879, 06 1964.
26. Kenneth E. Train. *Discrete Choice Methods with Simulation.* Cambridge University Press, 2009.
27. Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.