Nash Social Welfare, Matrix Permanent, and Stable Polynomials

Nima Anari[1], Shayan Oveis Gharan[2], Amin Saberi[1], and Mohit Singh[3]

1 **Stanford University.**
2 **University of Washington.**
3 **Microsoft Research, Redmond.**

Abstract

We study the problem of allocating m items to n agents subject to maximizing the Nash social welfare (NSW) objective. We write a novel convex programming relaxation for this problem, and we show that a simple randomized rounding algorithm gives a $1/e$ approximation factor of the objective, breaking the $1/2e^{1/e}$ approximation factor of Cole and Gkatzelis [8].

Our main technical contribution is an extension of Gurvits's lower bound on the coefficient of the square-free monomial of a degree m-homogeneous stable polynomial on m variables to all homogeneous polynomials. We use this extension to analyze the expected welfare of the allocation returned by our randomized rounding algorithm.

1998 ACM Subject Classification F.2.1 [Numerical Algorithms and Problems] Computations on Polynomials, G.2.1 [Combinatorics] Counting Problems, G.1.6 [Optimization] Convex Programming, G.3 [Probability and Statistics] Probabilistic Algorithms

Keywords and phrases Nash Welfare, Permanent, Matching, Stable Polynomial, Randomized Algorithm, Saddle Point

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

We study the problem of allocating a set of indivisible items to agents subject to maximizing the Nash social welfare (NSW). We are given a set of m indivisible items and we want to assign them to n agents. An allocation vector is a vector $\mathbf{x} \in \{0,1\}^{n\times m}$ such that for each j, exactly one $x_{i,j}$ is 1. We assume that agents have additive valuations. That is, each agent i has nonnegative value $v_{i,j}$ for an item j and the value that i receives for an allocation $\mathbf{x}$ is

$$v_i(\mathbf{x}) = \sum_{j=1}^{m} x_{i,j} v_{i,j}.$$

The NSW objective is to compute an allocation $\mathbf{x}$ that maximizes the geometric mean of agents' values,

$$\left(\prod_{i=1}^{n} v_i(\mathbf{x})\right)^{\frac{1}{n}}.$$

The above objective naturally encapsulates both fairness and efficiency and has been extensively studied as a notion of fair division (see [14, 5] and references therein).

Recently, there have been a number of results that study the computational complexity of the Nash social welfare objective. For additive valuations it is known that it is NP-hard to approximate the NSW objective within $(1-c)$ [15, 13], for some constant $c > 0$. On the positive side, Nguyen and Rothe [16] designed a $\left(\frac{1}{m-n+1}\right)$ approximation algorithm and Cole and Gkatzelis [8] gave the first constant factor, $\left(\frac{1}{2e^{1/e}}\right)$-approximation. Recently, independent of our work, Cole et al. [7] gave a $\frac{1}{2}$-approximation.

A closely related problem, that captures only fairness, is the Santa-Clause problem where the goal is to find an allocation to maximize the minimum value among all agents, i.e., $\max_{\mathbf{x}} \min_i v_i(\mathbf{x})$ which has also been studied recently [1, 2, 3, 6].

1.1 Our Contributions

Our main contribution is to show an intricate connection between the Nash welfare maximization problem, the theory of real stable polynomials, and the problem of approximating the permanent. We establish this connection in the following manner. We first give a new mathematical programming relaxation for the problem; indeed the standard relaxation has arbitrarily large integrality gap as shown by Cole and Gkatzelis [8]. Our relaxation is a polynomial optimization problem[1] which, despite not being convex in the standard form, can be solved efficiently by a change of variables. We remark that a similar geometric program was used in the context of maximum sub-determinant problem [17].

More precisely, we study a real stable polynomial $p(y_1, \ldots, y_m)$. We give a simple randomized rounding algorithm such that the expected Nash welfare of the allocation returned by the algorithm exactly equals the sum of *square-free* coefficients of $p(\mathbf{y})$. Thus, our program needs to maximize the sum of square-free coefficients of $p(\mathbf{y})$. Unfortunately, such an optimization problem in not convex. Instead, we maximize the following proxy

$$\inf_{\substack{\mathbf{y}>0: \\ \prod_{i\in S} y_i \geq 1, \forall S \in \binom{[m]}{n}}} p(\mathbf{y}).$$

The main part of our analysis is to relate the sum of square-free coefficients of $p(\mathbf{y})$ to the above proxy. This desired inequality is a generalization of an elegant result of Gurvits [11] relating the problem of approximating the permanent of a matrix with the theory of real stable polynomials. We prove this generalization in theorem 1.2. The connection to permanents allows us to use algorithmic results for approximating the permanent due to Jerrum, Sinclair and Vigoda [12] and we obtain the following result.

▶ **Theorem 1.1.** *There is a randomized polynomial time algorithm for the Nash welfare maximization problem that, with high probability, returns a solution with objective at least $1/e$ fraction of the optimum.*

We emphasize that unlike the recent constant factor approximation algorithms by Cole and Gkatzelis [8] and [7], our algorithm and its analysis are purely algebraic and completely oblivious to the structure of the underlying market. In particular, unlike other approaches we are not taking advantage of the combinatorial structure of “spending restricted assignments” in our rounding algorithms (see [8] for more information). This generality makes our approach potentially applicable to a variety of resource allocation problems.

The crucial ingredient of our analysis is the following general inequality about real stable polynomials that generalizes the result of Gurvits [11] (see theorem 2.5) that provided an elegant proof of the Van-der-Waerden conjecture.

[1] It falls in the broad class of geometric programs, where the mathematical program is convex in logarithms of the variables and not the variables itself.

▶ **Theorem 1.2.** *Let p be a degree n-homogeneous real stable polynomial in $y_1, \ldots, y_m$ with nonnegative coefficients. For any set $S \subseteq [m]$, let c_S denote the coefficient of monomial $y^S := \prod_{i \in S} y_i$. If $\sum_{S \in \binom{[m]}{n}} c_S > 0$, then*

$$\sum_{S \in \binom{[m]}{n}} c_S \geq \frac{m! \cdot (m-n)^{m-n}}{m^m \cdot (m-n)!} \inf_{\substack{\mathbf{y}>0: \\ y^S \geq 1, \forall S \in \binom{[m]}{n}}} p(\mathbf{y}) \tag{1}$$

$$\geq e^{-n} \inf_{\substack{\mathbf{y}>0: \\ y^S \geq 1, \forall S \in \binom{[m]}{n}}} p(\mathbf{y}).$$

Note that second inequality follows by lemma 1.1, $\frac{m!}{m^m} \cdot \frac{(m-k)^{m-k}}{(m-k)!} \geq e^{-k}$. By setting $n = m$ in the above statement, we obtain the result of Gurvits as described in theorem 2.5.

It is not hard to see that the above inequality is (almost) tight. For the stable n-homogeneous polynomial $p(y_1, \ldots, y_m) = (y_1 + \cdots + y_n)^n$, the LHS is $n!$ and the RHS is $(n/e)^n$. This tight example was already studied by Friedland and Gurvits [9] to show tightness of a lower bound for the number of matchings in regular bipartite graphs.

2 Preliminaries

For a vector $\mathbf{y}$, we write $\mathbf{y} \leq 1$ to denote that all coordinates of $\mathbf{y}$ are at most 1. For an integer $n \geq 1$ we use $[n]$ to denote the set of numbers $\{1, 2, \ldots, n\}$. For any m, n, we let $\binom{[m]}{n}$ denote the collection of subsets of $[m]$ of size n.

2.1 Stable Polynomials

Stable polynomials are natural multivariate generalizations of real-rooted univariate polynomials. For a complex number z, let $\text{Im}(z)$ denote the imaginary part of z. We say a polynomial $p(z_1, \ldots, z_m) \in \mathbb{C}[z_1, \ldots, z_m]$ is *stable* if whenever $\text{Im}(z_i) > 0$ for all $1 \leq i \leq m$, $p(z_1, \ldots, z_m) \neq 0$. We say $p(.)$ is real stable, if it is stable and all of its coefficients are real. It is easy to see that any univariate polynomial is real stable if and only if it is real rooted.

We say a polynomial $p(z_1, \ldots, z_m)$ is degree n-homogeneous, or n-homogenous, if every monomial of p has degree exactly n. Equivalently, p is n-homogeneous if for all $a \in \mathbb{R}$, we have

$$p(a \cdot z_1, \ldots, a \cdot z_m) = a^n p(z_1, \ldots, z_m).$$

We say a monomial $z_1^{\alpha_1} \ldots z_m^{\alpha_m}$ is *square-free* if $\alpha_1, \ldots, \alpha_m \in \{0, 1\}$. For a set $S \subset 2^{[m]}$ we write

$$z^S = \prod_{i \in S} z_i.$$

Thus, we can represent a square-free monomial with the set of indices of variables in that monomial.

▶ **Fact 2.1.** *If $p(z_1, \ldots, z_m)$ and $q(z_1, \ldots, z_m)$ are stable then $p \cdot q$ is stable.*

▶ **Fact 2.2.** *The polynomial $\sum_i a_i z_i$ is stable if $a_i \geq 0$ for all i.*

Polynomial optimization problems involving real stable polynomials with nonnegative coefficients can often be turned into concave/convex programs. Such polynomials are log-concave in the positive orthant:

▶ **Theorem 2.3** ([10]). *For any n-homogeneous stable polynomial $p(x_1, \ldots, x_n)$ with non-negative coefficients, $\log p(\mathbf{x})$ is concave in the positive orthant, $\mathbb{R}^n_{++}$.*

It is also an immediate corollary of Hölder's inequality that a polynomial with nonnegative coefficients is log-convex in terms of the log of its variables (for more details on log-convex functions see [4]).

▶ **Fact 2.4.** *For any polynomial $p(y_1, \ldots, y_m)$ with nonnegative coefficients, $\log p(\mathbf{y})$ is convex in terms of $\log \mathbf{y}$. In other words, $\log p(e^{z_1}, \ldots, e^{z_m})$ is convex in terms of $\mathbf{z}$.*

The following theorem is proved by Gurvits [11].

▶ **Theorem 2.5** ([11]). *For any degree m-homogeneous stable polynomial $p(z_1, \ldots, z_m)$ with nonnegative coefficients, let $c_{[m]}$ denote the coefficient of the multilinear monomial $z_1 \cdots z_m$. If $c_{[m]} > 0$, then*

$$c_{[m]} \geq \frac{m!}{m^m} \inf_{\mathbf{z}>0} \frac{p(z_1, \ldots, z_m)}{z_1 \ldots z_m}.$$

2.2 Counting Matchings in Bipartite Graphs

Given a bipartite graph $G = (X, Y, E)$ with weights $w : E \to \mathbb{R}$, the weight of a perfect matching M is defined as follows:

$$w(M) = \prod_{e \in M} w_e.$$

Jerrum, Sinclair, and Vigoda in their seminal work designed a FPRAS to count the sum of (weighted) perfect matchings of an arbitrary bipartite graph with nonnegative weights. This problem is also equivalent to the computation of the permanent of a nonnegative matrix.

▶ **Theorem 2.6** ([12]). *There exists a randomized polynomial time algorithm that for any arbitrary bipartite graph G with n vertices and nonnegative weights and $\epsilon > 0$ in time polynomial in the size of G and $1/\epsilon$ approximates the sum of weights of all perfect matchings of G within a $1 + \epsilon$ multiplicative error, with high probability.*

A k-matching of a bipartite graph $G = (X, Y, E)$ is a set $M \subseteq E$ of size $|M| = k$ such that no two edges share an endpoint. The following corollary follows immediately from the above theorem. For completeness, we prove it in the appendix.

▶ **Corollary 2.7.** *There is a randomized polynomial time algorithm that for any arbitrary bipartite graph G with nonnegative edge weights and for any given $\epsilon > 0$ and integer $k \leq n$ in time polynomial in the size of G and $1/\epsilon$ approximates the sum of the weights of all k-matchings of G within $1 + \epsilon$ multiplicative error, with high probability.*

3 Approximation Algorithm for NSW Maximization

In this section, we give an approximation algorithm for the NSW maximization problem. We begin by giving a mathematical programming relaxation that can be efficiently solved. For convenience, we will aim to optimize

$$\left(\prod_{i=1}^{n} v_i(\mathbf{x}) \right),$$

which is the n^{th} power of the NSW objective. Thus, it is enough to give an e^{-n}-approximation to the above objective. With a slight abuse of notation, we will also refer to problem of maximizing the new objective as the Nash welfare problem. In section 3.2, we give a rounding algorithm that proves the guarantee claimed in Theorem 1.1.

3.1 Mathematical Programming Relaxation

We use the following mathematical program.

$$
\begin{aligned}
\max_{\mathbf{x}} \quad & \inf_{\mathbf{y}>0: y^S \geq 1, \forall S \in \binom{[m]}{n}} \quad \prod_{i=1}^{n} \left(\sum_{j=1}^{m} x_{i,j} v_{i,j} y_j \right), \\
\text{s.t.} \quad & \sum_{i=1}^{n} x_{i,j} \leq 1 \qquad \forall 1 \leq j \leq m, \\
& x_{i,j} \geq 0 \qquad \forall i, j.
\end{aligned} \tag{2}
$$

First, we show that (2) is a relaxation of the Nash welfare problem and can be optimized in polynomial time to an arbitrary accuracy.

▶ **Lemma 3.1.** *The mathematical program* (2) *is a relaxation of the Nash welfare problem and can be optimized in polynomial time.*

Proof. Let $x^* \in \{0,1\}^{n \times m}$ be an optimal solution of the Nash welfare problem and let $\sigma : [m] \to [n]$ denote the assignment, i.e., $\sigma(j) = i$ if and only if $x^*_{ij} = 1$. We show that x^* is a feasible solution (2) of objective $\prod_{i=1}^n v_i(x^*)$. Consider any $\mathbf{y} > 0$ such that $y^S \geq 1$ for each $S \subseteq \binom{[m]}{n}$. Moreover let $\mathcal{S} = \{S \in \binom{[m]}{n} : \forall i \in [n], \exists j \in S \text{ such that } x^*_{ij} = 1\}$. We have

$$
\begin{aligned}
\prod_{i=1}^{n} \left(\sum_{j=1}^{m} x^*_{i,j} v_{i,j} y_j \right) &= \textstyle\sum_{S \in \mathcal{S}} y^S \prod_{j \in S} v_{\sigma(j),j} \\
&\geq \textstyle\sum_{S \in \mathcal{S}} \prod_{j \in S} v_{\sigma(j),j} \\
&= \textstyle\prod_{i=1}^{n} \left(\sum_{j=1}^{m} x^*_{i,j} v_{i,j} \right)
\end{aligned}
$$

as required, where we use the fact that $y^S \geq 1$ for each $S \in \mathcal{S}$. To show that the objective of the mathematical program equals $\prod_{i=1}^n v_i(x^*)$, we consider the solution $y^*_j = 1$ for each $j \in [m]$.

To solve the mathematical program, we observe that the function $\log \prod_{i=1}^n \sum_{j=1}^m x_{i,j} v_{i,j} y_j$ is concave in x and convex in $\log \mathbf{y}$, where $\log \mathbf{y}$ is the vector defined by taking logarithms of the vector $\mathbf{y}$ coordinate-wise. These follow from theorem 2.3 and fact 2.4. Moreover, the constraints on $\mathbf{x}$ and $\log \mathbf{y}$ are linear. Thus the above program can be formulated as a convex program and solved to an arbitrary accuracy. ◀

3.2 Randomized Algorithm I

We now give a rounding algorithm that proves the required guarantee. Algorithm 1 will only satisfy the guarantee in expectation. Later, we show how to give a randomized algorithm that gives essentially the same guarantee with high probability.

Algorithm 1 An Algorithm for NSW Maximization

Check whether the optimal solution has weight strictly more than zero using the bipartite matching algorithm. Return zero if answer is false.
Find an optimal solution $\mathbf{x}^*$ to the mathematical program (2).
Independently for each item $j \in [m]$, assign item j to one agent where agent $i \in [n]$ is chosen with probability x^*_{ij}.

The first step of the algorithm can be implemented by a bipartite matching problem. Indeed consider the bipartite graph with one side as agents and other as items. We have an edge (i,j) for agent i and item j if $v_{ij} > 0$. The optimal solution to the NSW maximization problem is strictly positive if and only if this bipartite graph has a matching that includes an edge at every agent. Thus, we can check in polynomial time whether the optimal solution has weight zero. For the remainder of the section, we assume that the optimal solution is strictly positive.

We say $\mathbf{x} \in \mathbb{R}_+^{n\times m}$ is a fractional allocation vector if for each $j \in [m]$, $\sum_{i=1}^n x_{i,j} = 1$. Given any fractional allocation $\mathbf{x}$, consider the following polynomial in variables $y_1, \dots, y_n$,

$$p_{\mathbf{x}}(y_1,\dots,y_n) = \prod_{i=1}^n \left(\sum_{j=1}^m x_{i,j} v_{i,j} y_j \right).$$

Observe that $p_{\mathbf{x}}(\mathbf{y})$ is a degree n-homogenous polynomial in m variables for any $\mathbf{x}$ or the identically 0 polynomial.

▶ **Lemma 3.2.** *We have the following.*

1. *For $S \subseteq [m]$ of size n, let c_S denote the coefficient of y^S in $p_{\mathbf{x}^*}(\mathbf{y})$. Then, the expected value of algorithm 1 equals*
$$\sum_{S \in \binom{[m]}{n}} c_S.$$
2. *The optimal value of the relaxation* (2) *is*
$$\inf_{\mathbf{y}: y^S \geq 1, \forall S \in \binom{[m]}{n}} p_{\mathbf{x}^*}(\mathbf{y}).$$

Proof. Let $X_{i,j}$ be the random variable indicating that j is assigned to i. Then, the value that i receives is $\sum_{j=1}^m X_{i,j} v_{i,j}$. So, the expected value of the algorithm is

$$\mathbb{E}\left[\prod_{i=1}^n \sum_{j=1}^m X_{i,j} v_{i,j}\right] = \sum_{\sigma:[n]\to[m]} \mathbb{E}\left[\prod_{i=1}^n X_{i,\sigma(i)} v_{i,\sigma(i)}\right] =$$
$$\sum_{\sigma:[n]\to[m]} \mathbb{P}\left[\forall i : X_{i,\sigma(i)} = 1\right] \prod_{i=1}^n v_{i,\sigma(i)}.$$

where σ is summed over all functions from $[n]$ to $[m]$. Observe that $\mathbb{P}\left[\forall i : X_{i,\sigma(i)=1}\right] \neq 0$ only if σ is a one-to-one function. In such a case, we have $\mathbb{P}\left[\forall i : X_{i,\sigma(i)} = 1\right] = \prod_{i=1}^n x_{i,\sigma(i)}$ where we use the fact that each item is assigned independently. Therefore,

$$\mathbb{E}\left[\prod_{i=1}^n \sum_{j=1}^m X_{i,j} v_{i,j}\right] = \sum_{\sigma:[n] \underset{\text{one-to-one}}{\to} [m]} \prod_{i=1}^n x_{i,\sigma(i)} v_{i,\sigma(i)}.$$

The lemma follows by the fact that for any one-to-one σ, the term $\prod_{i=1}^n x_{i,\sigma(i)} v_{i,\sigma(i)}$ on the RHS appears in the coefficient of the (square-free) monomial $\prod_{i=1}^n y_{\sigma(i)}$ of $p_{\mathbf{x}^*}(\mathbf{y})$. For any $S \in \binom{[m]}{n}$ the coefficient of y^S in $p_{\mathbf{x}^*}(\mathbf{y})$ is the sum of all such terms where $\sigma([n]) = S$.

The proof of the second claim is immediate by definition. ◀

We are now ready to apply theorem 1.2 and obtain the following immediate corollary.

▶ **Corollary 3.3.** *The expected objective of algorithm 1 is at least*

$$e^{-n} \cdot \mathrm{OPT}$$

where OPT *is the optimal NSW objective.*

Proof. From fact 2.1 and fact 2.2, it follows that $p_{\mathbf{x}^*}(\mathbf{y})$ as defined above is real stable with nonnegative coefficients. Moreover, it is an m-variate polynomial that is degree n-homogenous. Let c_S denote the coefficient of square-free monomial y^S for any $S \in \binom{[m]}{n}$. Since, we assume that there is at least one assignment that has strictly positive NSW objective, the sum of coefficients $\sum_{S\in\binom{[m]}{n}} c_S > 0$. Thus, from theorem 1.2, we have

$$\sum_{S\in\binom{[m]}{n}} c_S \quad \geq \quad e^{-n} \min_{\mathbf{y}: y^S \geq 1, \forall S \in \binom{[m]}{n}} p_{\mathbf{x}^*}(\mathbf{y}).$$

Now the proof is immediate using lemma 3.2. ◀

3.3 Randomized Algorithm II

From corollary 3.3, the expected NSW of the allocation returned by algorithm 1 is at least $1/e^n$ fraction of the optimum. Repeated applications of the algorithm to obtain a high probability bound is not possible since the output of algorithm 1 may have an exponentially large variance. In this section, we prove Theorem 1.1 by giving an algorithm that returns the same guarantee as algorithm 1 with high probability.

Proof of theorem 1.1. We use the method of conditional expectations to prove the theorem. We iteratively assign one item at a time, making sure that conditional expectation over the random assignment of the remaining items does not decrease (substantially). We now claim that for any assignment x, the expected value of the objective as given by randomized algorithm 1 equals the number of weighted n-matchings of a bipartite graph. Consider the weighted bipartite graph $G = ([n], [m], E)$ where for any $1 \leq i \leq n$ and $1 \leq j \leq m$, $w_{i,j} = x_{i,j} v_{i,j}$. Then, for one-to-one mapping $\sigma : [n] \to [m]$, the coefficient of the monomial $\prod_{i=1}^n x_{i,\sigma(i)} v_{i,\sigma(i)}$ is equal to the weight of the n-matching $\{(1, \sigma(1)), (2, \sigma(2)), \ldots, (n, \sigma(n))\}$. Therefore, the sum of square-free monomials of $p_{\mathbf{x}}(\mathbf{y})$ is equal to the sum of the weights of all n-matchings of G.

Now, pick any item $j \in [m]$ and any fractional assignment x. Consider the following n assignments, $x^1, \ldots, x^n$. Assignment x^i assigns item j to i and rest of the items identically to the fractional assignment x. Thus $x^i_{ij} = 1$, $x^i_{i',j} = 0$ for all $i \neq i'$ and $x^i_{i'j'} = x_{i'j'}$ for any $j' \neq j$. Let ALG^i denote the objective value of the output of algorithm 1 on solution x^i and ALG on x. Since the objective value of the algorithm 1 is linear in $\{x_{ij} : i \in [n]\}$ for fixed j, we have

$$\mathrm{ALG} = \sum_{i=1}^n x_{ij} \, \mathrm{ALG}^i$$

Thus ALG is the expected value of the conditional expected value of the output of the algorithm 1 when we assign item j to one of the agents; it is assigned to agent i with probability x_{ij}.

By corollary 2.7, we can estimate ALG and ALG^i within a factor of $1 + 1/m^3$ factor in polynomial time. Therefore, using the method of conditional expectations, we obtain an allocation of NSW of value at least $\frac{\mathrm{OPT}}{e^n} \cdot (1 - 1/m^3)^m \geq \frac{\mathrm{OPT}}{\left((1+\frac{1}{n})e\right)^n}$ where OPT denotes the objective of the optimal allocation. ◀

4 A Generalization of Gurvits's Theorem

In this section we prove theorem 1.2. Let

$$q(y_1,\dots,y_m)=(y_1+\cdots+y_m)^{m-n}$$

be a degree $(m-n)$-homogenous polynomial. It is straightforward to see that it is real stable. Consider the polynomial $p(\mathbf{y})q(\mathbf{y})$. Observe that this is a degree m-homogeneous stable polynomial with nonnegative coefficients. Since from the assumption of theorem 1.2, at least one of the square-free monomials in $p(\mathbf{y})$ has a non-zero coefficient, the coefficient of the square-free monomial in $p(\mathbf{y})q(\mathbf{y})$ is non-zero. Let $\alpha_{[m]}$ be the coefficient of the square-free monomial $y_1\cdots y_m$ in $p(\mathbf{y})q(\mathbf{y})$. Thus, from theorem 2.5, we have

$$\alpha_{[m]}\geq\frac{m!}{m^m}\inf_{\mathbf{y}>0}\frac{p(\mathbf{y})q(\mathbf{y})}{y_1\dots y_m}. \tag{3}$$

To prove theorem 1.2 it is enough to relate the LHS and the RHS of (3) to the two sides of (1). This is done in lemma 4.1 and proposition 4.2.

▶ **Lemma 4.1.** *We have*

$$(m-n)!\sum_{S\in\binom{[m]}{n}}c_S=\alpha_{[m]}.$$

Proof. The RHS is the coefficient of the square-free monomial $y_1\dots,y_m$ in $p(\mathbf{y})q(\mathbf{y})$. The square-free monomial of $p(\mathbf{y})q(\mathbf{y})$ is obtained whenever we multiply a square-free monomial y^S of $p(\mathbf{y})$ with the square-free monomial $y^{\overline{S}}$ of $q(\mathbf{y})$ for some $S\in\binom{[m]}{n}$. Lemma's statement follows by the fact that the coefficient of $y^{\overline{S}}$ in $q(\mathbf{y})$ is $(m-n)!$ for every $S\in\binom{[m]}{n}$ and the coefficient of y^S in $p(\mathbf{y})$ is c_S. ◀

The proof of theorem 1.2 is now immediate from the following proposition which relates the RHS of (3) and (1).

▶ **Proposition 4.2.**

$$\inf_{\mathbf{y}>0}\frac{p(\mathbf{y})q(\mathbf{y})}{y_1\dots y_m}\geq(m-n)^{m-n}\inf_{\mathbf{y}>0:y^S\geq1,\forall S\in\binom{[m]}{n}}p(\mathbf{y}).$$

In the rest of this section we prove the above proposition. We do the proof in two steps. First, we use convex duality to simplify the RHS, and then we prove the proposition.

▶ **Lemma 4.3.**

$$\inf_{\mathbf{y}>0:y^S\geq1,\forall S\in\binom{[m]}{n}}p(\mathbf{y})=\sup_{0\leq\theta\leq1:\sum_{j=1}^m\theta_j=n}\ \inf_{\mathbf{y}>0}\frac{p(\mathbf{y})}{y_1^{\theta_1}\dots y_m^{\theta_m}}.$$

Proof. The proof follows by convex duality. By taking logarithm of $p(\mathbf{y})$ and the change of variable $z_j=e^{y_j}$, we obtain the following equivalent convex program to the LHS of the above inequality.

$$\begin{aligned}&\inf\quad \log p(e^{z_1},\dots,e^{z_m})\\&\text{s.t.}\ \sum_{i\in S}z_i\geq0\qquad\qquad\forall S\in\binom{[m]}{n}.\end{aligned}\tag{4}$$

Let λ_S be the Lagrange dual variable associated to the constraint corresponding to the set $S \in \binom{[m]}{n}$. The Lagrangian of the above convex program is defined as follows:

$$L(\mathbf{z}, \lambda) = \log p(e^{z_1}, \ldots, e^{z_m}) - \sum_{S \in \binom{[m]}{n}} \lambda_S \sum_{i \in S} z_i.$$

The Lagrange dual to (4) is

$$\sup_{\lambda \geq 0} \inf_{\mathbf{z}} L(\mathbf{z}, \lambda).$$

Since $p(\mathbf{y})$ has a non-zero coefficient for at least one of the square-free monomials, the objective of the convex program (4) is finite for any $\mathbf{z}$ and it is easy to see that the Slater conditions are satisfied. Thus the optimum value of the Lagrange dual is exactly equal to the optimum of (4).

Let $\mathbf{z}^*, \lambda^*$ be an optimum of the above program. We claim that $\sum_S \lambda^*_S = 1$. This simply follows from first order optimality conditions. If $\sum_S \lambda^*_S < 1$, then

$$\begin{aligned} L(\mathbf{z}^* - \epsilon, \lambda^*) &= \log p(e^{z_1^* - \epsilon}, \ldots, e^{z_m^* - \epsilon}) - \sum_{S \in \binom{[m]}{n}} \lambda^*_S \sum_{j \in S} (z^*_j - \epsilon) \\ &= L(\mathbf{z}^*, \lambda^*) - n \cdot \epsilon + \sum_{S \in \binom{[m]}{n}} n \lambda^*_S \epsilon < L(\mathbf{z}^*, \lambda^*). \end{aligned}$$

Similarly, if $\sum_{S \in \binom{[m]}{n}} \lambda_S > 1$, $L(\mathbf{z}^* + \epsilon, \lambda^*) < L(\mathbf{z}^*, \lambda^*)$. So, λ^* is a probability distribution on sets of size n. We let $L'(\mathbf{z}, \theta) = \log p(e^{\mathbf{z}}) - \sum_{j=1}^m z_j \theta_j$ Thus, we obtain that

$$\sup_{0 \leq \theta \leq 1: \sum_{j=1}^m \theta_j = n} \inf_{\mathbf{z}} L'(\mathbf{z}, \theta) \geq \sup_{\lambda \geq 0} \inf_{\mathbf{z}} L(\mathbf{z}, \lambda).$$

by setting $\theta^*_j = \sum_{S \in \binom{[m]}{n} : j \in S} \lambda^*_S$ to be the marginal probability of the element j.

We now claim that equality must hold in the above. This follows since given any $\theta \in \{0 \leq \theta \leq 1 : \sum_{j=1}^m \theta_j = n\}$, there exists a probability distribution over sets of size n such that marginal of every element is exactly θ_j. Setting λ'_S to be the probability of set $S \in \binom{[m]}{n}$, we obtain that for any $\mathbf{z}$ and θ, we have $L'(\mathbf{z}, \theta) = L(\mathbf{z}, \lambda')$. Putting this together we have

$$\inf_{\sum_{j \in S} z_j \geq 0, \forall S \in \binom{[m]}{n}} \log p(e^{\mathbf{z}}) = \sup_{0 \leq \theta \leq 1: \sum_{j=1}^m \theta_j = n} \inf_{\mathbf{z}} \left(\log p(e^{\mathbf{z}}) - \sum_{j=1}^m z_j \theta_j \right).$$

Substituting e^{z_j} with y_j and taking the exponential of the objective functions we have

$$\inf_{\mathbf{y} > 0 : y^S \geq 1, \forall S \in \binom{[m]}{n}} p(\mathbf{y}) = \sup_{0 \leq \theta \leq 1: \sum_{j=1}^m \theta_j = n} \inf_{\mathbf{y} > 0} \frac{p(\mathbf{y})}{y_1^{\theta_1} \ldots y_m^{\theta_m}}$$

as desired. ◀

Now we give the proof of proposition 4.2.

Proof of proposition 4.2. By lemma 4.3, it is enough to show that

$$\inf_{\mathbf{y} > 0} \frac{p(\mathbf{y}) q(\mathbf{y})}{y_1 \ldots y_m} \geq (m - n)^{m-n} \sup_{0 \leq \theta \leq 1: \sum_{j=1}^m \theta_j = n} \inf_{\mathbf{y} > 0} \frac{p(\mathbf{y})}{y_1^{\theta_1} \ldots y_m^{\theta_m}}.$$

Let θ be any vector such that $0 \leq \theta \leq 1$ and $\sum_i \theta_i = n$. It is enough to show for any such θ,

$$\inf_{\mathbf{y}>0} \frac{p(\mathbf{y})q(\mathbf{y})}{y_1 \dots y_m} \geq (m-n)^{m-n} \inf_{\mathbf{y}>0} \frac{p(\mathbf{y})}{y_1^{\theta_1} \dots y_m^{\theta_m}}.$$

We prove a stronger statement,

$$\inf_{\mathbf{y}>0} \frac{p(\mathbf{y})}{y_1^{\theta_1} \dots y_m^{\theta_m}} \cdot \inf_{\mathbf{y}>0} \frac{q(\mathbf{y})}{y_1^{1-\theta_1} \cdot y_m^{1-\theta_m}} \geq (m-n)^{m-n} \inf_{\mathbf{y}>0} \frac{p(\mathbf{y})}{y_1^{\theta_1} \dots y_m^{\theta_m}}.$$

Equivalently, we show that

$$\inf_{\mathbf{y}>0} \frac{q(\mathbf{y})}{y_1^{1-\theta_1} \dots y_m^{1-\theta_m}} \geq (m-n)^{m-n}$$

Taking $(m-n)$-th root of both sides it is enough to show that

$$\inf_{\mathbf{y}>0} \frac{y_1 + \dots + y_m}{y_1^{\alpha_1} \dots y_m^{\alpha_m}} \geq m-n, \tag{5}$$

where $\alpha_j = \frac{1-\theta_j}{m-n}$ for all $j \in [m]$. Note that by the definition of θ, we have $0 \leq \alpha_j \leq \frac{1}{m-n}$ and that

$$\sum_i \alpha_j = \frac{m - \sum_{j=1}^m \theta_j}{m-n} = 1.$$

Therefore, the ratio on the LHS of (5) is homogeneous in $\mathbf{y}$. Thus, to prove (5), it is enough to prove the following

$$\sup_{\mathbf{y}>0: \sum_{j=1}^m y_j = 1} y_1^{\alpha_1} \dots y_m^{\alpha_m} \leq \frac{1}{m-n}. \tag{6}$$

Next, we use the weighted AM-GM inequality. We let $\alpha_1, \dots, \alpha_m$ be the weights, and recall that α_j's sum to 1. Weighted AM-GM implies that

$$\sum_{j=1}^m \alpha_j \frac{y_j}{\alpha_j} \geq \prod_{j=1}^m \left(\frac{y_j}{\alpha_j}\right)^{\alpha_j} = \prod_{j=1}^m \alpha_j^{-\alpha_j} \prod_{j=1}^m y_j^{\alpha_j}$$

Therefore,

$$\sup_{\mathbf{y}>0: \sum_{j=1}^m y_j = 1} \prod_{j=1}^m y_j^{\alpha_j} \leq \prod_{j=1}^m \alpha_j^{\alpha_j}.$$

To prove (6), it is enough to show that

$$\prod_{j=1}^m \alpha_j^{\alpha_j} \leq \frac{1}{m-n}.$$

Or equivalently,

$$\sum_{j=1}^m -\alpha_j \log \alpha_j \geq \log(m-n).$$

Since $\alpha_j \leq \frac{1}{m-n}$ and that $\sum_{j=1}^m \alpha_j = 1$, we have

$$\sum_{j=1}^m -\alpha_j \log \alpha_j \quad \geq \quad \sum_{j=1}^m -\alpha_j \log \frac{1}{m-n} = \log(m-n) \sum_{j=1}^m \alpha_j = \log(m-n),$$

as required. ◀

References

1 Arash Asadpour, Uriel Feige, and Amin Saberi. Santa claus meets hypergraph matchings. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 10–20. Springer, 2008.

2 Arash Asadpour and Amin Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. In *STOC*, pages 114–121, 2007.

3 N Bansal and M Sviridenko. The santa claus problem. In *Proceedings of Symposium on Theory of Computing*, pages 31–40, 2006.

4 S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2006.

5 Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum nash welfare. In *EC*, 2016.

6 Deeparnab Chakrabarty, Julia Chuzhoy, and Sanjeev Khanna. On allocating goods to maximize fairness. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 107–116. IEEE, 2009.

7 Richard Cole, Nikhil R. Devanur, Vasilis Gkatzelis, Kamal Jain, Tung Mai, Vijay V. Vazirani, and Sadra Yazdanbod. Convex program duality, fisher markets, and nash social welfare. submitted, 2016.

8 Richard Cole and Vasilis Gkatzelis. Approximating the nash social welfare with indivisible items. In *STOC*, pages 371–380. ACM, 2015.

9 Shmuel Friedland and Leonid Gurvits. Lower bounds for partial matchings in regular bipartite graphs and applications to the monomer–dimer entropy. *Combinatorics, Probability and Computing*, 17(03):347–361, 2008.

10 Osman Güler. Hyperbolic polynomials and interior point methods for convex programming. *MOR*, 22(2):350–77, 1997.

11 Leonid Gurvits. Hyperbolic polynomials approach to van der waerden/schrijver-valiant like conjectures: Sharper bounds, simpler proofs and algorithmic applications. In *STOC*, STOC '06, pages 417–426, 2006.

12 Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, July 2004.

13 Euiwoong Lee. Apx-hardness of maximizing nash social welfare with indivisible items. arXiv preprint arXiv:1507.01159, 2015.

14 Hervé Moulin. *Fair division and collective welfare*. MIT press, 2004.

15 Nhan-Tam Nguyen, Trung Thanh Nguyen, Magnus Roos, and Jörg Rothe. Computational complexity and approximability of social welfare optimization in multiagent resource allocation. *Autonomous agents and multi-agent systems*, 28(2):256–289, 2014.

16 Trung Thanh Nguyen and Jörg Rothe. Minimizing envy and maximizing average nash social welfare in the allocation of indivisible goods. *Discrete Applied Mathematics*, 179:54–68, 2014.

17 Aleksandar Nikolov and Mohit Singh. Maximizing determinants under partition constraints. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 192–201, 2016.

A Miscellaneous Lemmas

▶ **Lemma 1.1.** *For any $k \leq m$, we have*

$$\frac{m!}{m^m} \cdot \frac{(m-k)^{m-k}}{(m-k)!} \geq e^{-k}.$$

Proof. We prove by induction on k. The claim obviously holds for $k = 0$. For the induction step, it is sufficient to show that

$$\frac{1}{e} \cdot \frac{m!}{m^m} \cdot \frac{(m-k)^{m-k}}{(m-k)!} \leq \frac{m!}{m^m} \cdot \frac{(m-(k+1))^{m-(k+1)}}{(m-(k+1))!}.$$

Equivalently, it is enough to show that

$$\left(\frac{m-k}{m-(k+1)}\right)^{m-(k+1)} \leq e.$$

The above can be written as $(1 + \frac{1}{m-k-1})^{m-k-1} \leq e$. The latter follows by the fact that $1 + x \leq e^x$. ◀

Proof of corollary 2.7. Suppose that we are given a bipartite graph $G = (X, Y, E)$ where $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$. Note that m is not necessarily equal to n. We construct another graph $G' = (X', Y', E')$ such there is a one-to-$(m-k)!(n-k)!$ and onto mapping between the k-matchings of G and the perfect matchings of G'. That is each k-matching of G is mapped to a unique set of $(m-k)!(n-k)!$ perfect matchings of G', and for each perfect matching M' of G' there is a k-matching of G that has M' in its image.

Let $X' = X \cup \{x_{m+1}, \ldots, x_{m+n-k}\}$ and $Y' = Y \cup \{y_{n+1}, \ldots, y_{m+n-k}\}$. The set of edges E' is the union of E and the following edges: Connect all vertices of $X' \setminus X$ to all vertices of Y with weight 1, and connect all vertices of $Y' \setminus Y$ to all vertices of X with weight 1. Observe that for any k-matching M of G there are exactly $(m-k)!(n-k)!$ perfect matchings in G' that contain M; for any such prefect matching M', we have $M' \setminus M \subseteq E' \setminus E$. So, this mapping is one-to-$(m-k)!(n-k)!$. Furthermore, any perfect matching M' of G' has exactly k edges in E, i.e., $|M' \cap E| = k$. So, this mapping is onto.

It follows that a $1 + \epsilon$ approximation to the sum of the weights of all perfect matchings of G' is a $1 + \epsilon$ approximation to the sum of the weights of all k-matchings of G. ◀