Approximations for Indivisible Concave Allocations with Applications to Nash Welfare Maximization

Nathaniel Kell[1], **Kevin Sun**[2]

[1] Denison University
[2] Elon University
kelln@denison.edu, ksun4@elon.edu

Abstract

We study a general allocation setting where agent valuations are *concave additive*. In this model, a collection of items must be uniquely distributed among a set of agents, where each agent-item pair has a specified utility. The objective is to maximize the sum of agent valuations, each of which is an arbitrary non-decreasing concave function of the agent's total additive utility. This setting was studied by Devanur and Jain (STOC 2012) in the online setting for divisible items. In this paper, we obtain *both multiplicative and additive approximations in the offline setting for indivisible items.* Our approximations depend on novel parameters that measure the local multiplicative/additive curvatures of each agent valuation, which we show correspond directly to the integrality gap of the natural assignment convex program of the problem. Furthermore, we extend our additive guarantees to obtain *constant multiplicative approximations for Asymmetric Nash Welfare Maximization* when agents have *smooth valuations*. This algorithm also yields an interesting tatonnement-style interpretation, where agents adjust uniform prices and items are assigned according to maximum *weighted* bang-per-buck ratios.

Introduction

In recent years the study of *indivisible* allocation has received increasing attention: given a collection of indivisible items and a set of agents, each with a specified valuation function, how should items be distributed among the agents as to maximize a specified measure of overall welfare or fairness? Many classic allocation and market models consider divisible goods that can be split fractionally among agents. From an optimization perspective, there are several well-known methods for computing optimal fractional assignments in polynomial time (e.g., ellipsoid methods), while indivisible variants are often NP-hard. Ensuring fairness also becomes more complex in the indivisible setting, especially when agents exhibit diminishing returns in their valuations.

In this paper, we study a general yet natural allocation model that lies at the intersection of these algorithmic challenges, which we call *Indivisible Allocation with Concave-Additive Valuations* (ICA). As input, we are given a set of n agents and m indivisible items, where each agent i has a specified utility $u_{i,j}$ for each item j. An algorithm must partition the items into disjoint sets $(A_1, A_2, \dots A_n)$, one for each agent. The overall valuation agent i has for her set is $v_i(u_i)$, where her valuation $v_i\colon \mathbf{R}_+ \to \mathbf{R}$ is permitted to be any monotone (non-decreasing) concave function of her total additive utility $u_i = \sum_{j \in A_i} u_{i,j}$. The objective is to maximize the welfare of the allocation, i.e., the sum of agent valuations $\sum_i v_i(u_i)$. For ease of comparison to prior work, we refer to such valuations $v_i(\cdot)$ as *concave additive*.

The focus of this paper is on obtaining efficient allocations for ICA, i.e., designing approximation algorithms. However, several recent works have established close connections between approximability and fairness guarantees, e.g., see (Barman, Krishnamurthy, and Vaish 2018), (McGlaughlin and Garg 2020). Indeed, when optimizing over a concave objective, an efficient allocation must ultimately strike the correct balance between a *utilitarian allocation* (i.e., assign each item j to agent $\arg\max_i u_{i,j}$) and an *egalitarian allocation* (one that maximizes $\min_i u_i$), where the trade off between these two extremes depends on the degree of concavity. Thus, a primary technical goal of our work is to provide a precise and unified characterization for how to achieve this balance for the general class of concave additive functions.

Model Motivation. ICA is an indivisible variant of the fractional online problem studied by Devanur and Jain (2012). This model was primarily motivated by applications in internet advertising, where agents correspond to advertisers and items correspond to so-called "impressions" (opportunities to show ads to users). In this setting, $u_{i,j}$ translates to the bid value an advertiser i is willing pay for impression j. The concave objective is then used to capture common contract features in internet ad systems such as under-delivery penalties and soft budgets.

However, given the special role concave functions have played in the economics literature, we believe ICA is a natural problem to consider in its own right. For example, a standard class of valuations is that of *separable concave valuations* (Chen et al. 2009; Vazirani and Yannakakis 2011; Anari et al. 2018; Chaudhury et al. 2018), which can be decomposed into the sum of monotone concave functions over the amount received from each good (or in the indivisible settings, the numbers of *copies* of a single good). Thus, such valuations can express varying degrees of diminishing returns for receiving more of the same item.

But in many applications, agent valuations are in fact *inseparable*, since how much an agent values an additional item will likely depend on the other items she has already received. A canonical example of inseparable valuations are *budget-additive* functions, i.e., each valuation $v_i(u_i) = \min(u_i, c_i)$ where c_i denotes the utility cap for agent i. Thus, one can view concave-additive valuations as a general class of inseparable functions that can express diminishing returns beyond just a global cap on an agent's overall utility. There is an extensive line of work that has studied approximation algorithms for budget-additive valuations (Garg, Kumar, and Pandit 2001; Andelman and Mansour 2004; Azar et al. 2008; Chakrabarty and Goel 2010; Kalaitzis et al. 2015). To the best of our knowledge, the approximability of the concave-additive case has yet to be considered for indivisible items.

Another key motivation for our model is that an *additive* approximation for ICA translates to a standard multiplicative approximation for the problem of *Nash Welfare Maximization*. Here, the objective is to maximize the weighed product of valuations $\left(\prod_i (v_i(u_i))^{\eta_i}\right)^{1/\eta}$, where $\eta_i > 0$ is the weight of each agent and $\eta = \sum_i \eta_i$ is the sum of weights. Observe that as long as each valuation $v_i(\cdot)$ is concave additive, we can convert the objective to an instance of ICA[1] by taking the logarithm, giving us the objective $\frac{1}{\eta}\sum_i \eta_i \ln(v_i(u_i))$. Furthermore, an additive approximation of α for the log objective translates to a multiplicative approximation of e^{α} for the original product objective.

The main appeal of Nash welfare is that the objective itself is a natural balance between utilitarian and egalitarian welfare and has solutions with appealing fairness guarantees (see, e.g., (Conitzer, Freeman, and Shah 2017; Caragiannis et al. 2019)). It has been known for over eighty years that the optimal Nash-welfare assignment for divisible items can be obtained by solving the famous Eisenberg-Gale convex program (Eisenberg and Gale 1959). More recently, there has been an explosion of work examining the Nash welfare objective in the indivisible setting. As we will show, our general techniques for ICA contribute to this growing body of work by yielding new approximations when agents have *smooth* valuations. (We formally define this setting momentarily.)

Finally, we remark that in addition to modeling other settings in computational economics such as agents with spending constrained utilities (see (Vazirani 2010; Cole et al. 2017)), concave-additive functions occur naturally in many other allocation and matching problems in the broader scope of convex optimization and AI. Such examples include word alignment in natural language processing (Lin and Bilmes 2011), match scoring in database search (Bai, Bilmes, and Noble 2016), and ensuring diversity on team projects (Ahmed, Dickerson, and Fuge 2020).

Contributions

In this paper, we obtain both *multiplicative and additive approximations for the ICA problem*. Our approximations depend on novel curvature parameters that measure the degree of local change that can occur in the concave valuation of each agent. For a given agent valuation $v_i(\cdot)$ and maximum item utility $\max_j u_{i,j}$ for agent i, we denote the *local multiplicative and additive curvatures* of $v_i(\cdot)$ as μ_i and α_i, respectively, where higher/lower values of μ_i and α_i correspond to higher/lower degrees of local curvature in $v_i(\cdot)$ with respect to a local change of $\max_j u_{i,j}$.[2] As we will see, these parameters also provide our algorithms with a precise guide for striking the balance between utilitarian and egalitarian allocations when redistributing items among agents.

For the key applications of our general results, we also show our guarantees yield *approximations for Asymmetric Nash Welfare with smoothed agent valuations*, as well as instances where agents have *piecewise-linear valuations*. The details of these results and applications are stated formally below.

General Guarantees. For our main result, we give a polynomial-time algorithm for a general instance of ICA that achieves a multiplicative approximation ratio of $(1 + \epsilon)\max_i \mu_i$. Note for our general multiplicative results, we assume that each v_i maps from $\mathbf{R}_+$ to $\mathbf{R}_+$ and is differentiable over $\mathbf{R}_+$.

Theorem 1. *Consider an instance of* ICA *such that each* $v_i : \mathbf{R}_+ \to \mathbf{R}_+$ *has well-defined first derivative. Let* $u_i^{\max} = \sum_j u_{ij}$ *and* $\rho_{\max} = \max_i(v_i'(0)u_i^{\max}/v_i(u_i^{\max}))$*. Then there exists an algorithm that achieves an approximation of* $(1+\epsilon)\max_i \mu_i$ *that runs in time* $O(mn\ln(\rho_{\max}/\epsilon)/\epsilon)$*.*

Additionally, we show this bound is tight among algorithms that utilize the natural assignment convex program, i.e., the integrality gap of the ICA convex program is precisely our multiplicative curvature parameter (proof in full version of the paper).

Theorem 2. *Consider an instance* ICA *where each agent has valuation function* $v(\cdot)$ *and let* $u = \max_{i,j} u_{i,j}$*. Then the integrality gap of the* ICA *assignment convex program is the multiplicative curvature of the instance* μ *(determined by* $v(\cdot)$ *and* u*).*

These results can also be extended to obtain an algorithm with an additive guarantee for ICA, where the performance bound instead scales according to the additive curvatures of the valuations (formally stated later in Theorem 11). We emphasize that our algorithms are simple, efficient, and do not utilize a black-box convex program solver. Instead, we leverage the duality techniques introduced in (Chakrabarty and Goel 2010) and (Devanur and Jain 2012) to guide a tatonnement style local-search algorithm.

In fact, one can view our algorithm as a concave generalization of the LP primal-dual $4/3$-approximation algorithm for budget-additive functions in (Chakrabarty and

[1] An ICA instance with the added constraint that allocating $v_i(u_i) = 0$ for some agent i is disallowed or infinitely penalized.

[2] These parameters are formally defined and diagrammed in our preliminaries section.

Goel 2010). This algorithm begins with a utilitarian allocation by initializing dual bids greedily and assigning each item j to the agent i with maximum utility $u_{i,j}$. The algorithm then continuously lowers bids for items assigned to agents who are "over allocated," defecting items to the highest bidder throughout (thus creating a more egalitarian allocation). The crux of the algorithm is to determine at what point the bid lowering procedure should stop — an algorithm that stops too soon will leave too many agents either over or under allocated, whereas one that continues too long will assign too many items to agents with comparatively lower utilities.

In (Chakrabarty and Goel 2010), the authors derive the optimal stopping point from a set of algebraically obtained equations, which directly corresponds to the $4/3$ approximation ratio. The key to our analysis is also deriving the correct stopping point; however, a new approach is needed in our more general setting (e.g., we do not make assumptions about the algebraic closed-form of each $v_i(\cdot)$). For our main technical insight, we show this algebraic approach can be bypassed via more elegant geometric arguments that coincide directly with the definitions of our curvature parameters μ_i and α_i in the multiplicative and additive settings, respectively.

Application: Smooth Asymmetric Nash Welfare. As discussed earlier, there is a recent line of work that has extensively examined the Nash welfare objective for indivisible items. This interest was sparked by the seminal result of Cole and Gkatzelis (Cole and Gkatzelis 2015), who obtained the first constant approximation for *symmetric agents* with *additive valuations*, i.e., for all agents i we have $\eta_i = 1$ and $v_i(u_i) = u_i$. This bound has been subsequently improved and extended to more general valuation functions in the symmetric agent case. (See our related works section below for a detailed list.)

However, many important applications are in fact captured by the general *asymmetric* objective, i.e., agents have general weights $\eta_i > 0$. For instance, Asymmetric Nash Welfare has been leveraged in applications such as committee bargaining (e.g., each agent i represents a committee with size η_i) and ensuring diversity in the housing allocations, where weights express priority levels of various ethnic groups (Laruelle and Valenciano 2007; Benabbou et al. 2018). Other examples include allocating water in transboundary river basins (Degefu et al. 2016) and reaching compromises in international negotiations for climate policy (Yu et al. 2017).

Unfortunately, even when agents have additive valuations, the approximability of the asymmetric case still remains a key open problem in the area, where the best known approximation bound is currently $O(n)$ (Garg, Kulkarni, and Kulkarni 2020). We note that the original breakthrough for the symmetric objective in (Cole and Gkatzelis 2015) was directly aimed at circumventing the $\Omega(n)$ integrality gap of the assignment convex program for the Nash objective. Subsequent work has either generalized these techniques or utilized other approaches that exploit the symmetry of the agents' valuations. Currently it is unclear how to extend these ideas to the asymmetric setting.

Since our general techniques for ICA compete against the optimal fractional objective of the assignment convex program, the $\Omega(n)$ integrality gap also remains a barrier to directly applying our approach to the standard asymmetric objective. However, another proposed alternative for naturally handling the large integrality gap is to examine agents with *smooth valuations* (Fain, Munagala, and Shah 2018; Fluschnik et al. 2019; Barman et al. 2022). In the smooth valuation setting, we give each agent a (potentially fractional) copy of her favorite item at the outset of the allocation, which relaxes the degree to which the objective penalizes under allocations. Furthermore, smoothing the valuations still captures part of the technical challenge of the standard non-smooth asymmetric objective.[3]

Thus, as the main application of our general techniques for ICA, we show that *our additive bounds imply multiplicative approximations for Asymmetric Nash Welfare with smooth additive valuations*, formally defined as follows. First observe we can scale the Nash welfare objective of each agent i by $(\max_j u_{i,j})^{-\eta_i}$ without changing the approximation factor of the algorithm. Therefore, wlog we can assume that $\max_j u_{i,j} = 1$. We then define the smooth version of the Nash objective to be $\left(\prod_i (u_i + \omega)^{\eta_i}\right)^{1/\eta}$, where $\omega \in (0, 1]$ denotes the *smoothing parameter* for the instance. We then obtain the following result.

Theorem 3. *Consider an instance of Asymmetric Nash Welfare Maximization with smooth additive valuations. Then there exists an algorithm that runs in time $O(nm^2/(\epsilon\omega))$ that achieves an approximation of $O(e^{\epsilon}/(\omega \ln(1 + 1/\omega)))$ for any smoothing parameter $\omega \in (0, 1]$.*

Observe that for any constant smoothing parameter ω, we obtain a constant approximation in the smooth setting. For example when $\omega = 1$, the approximation ratio of the algorithm is ≈ 1.061 as $\epsilon \to 0$. Again, one should interpret ω as giving each agent a ω-fraction copy of her favorite item at the start of the allocation.

Furthermore, the resulting algorithm has an interesting combinatorial interpretation, which we call the *Weighted Bang-Per-Buck* (WBB) algorithm. Many approximations for the symmetric case (e.g., (Anari et al. 2018; Barman, Krishnamurthy, and Vaish 2018)) also use tatonnement-style algorithms that adjust prices p_j for each item j, maintaining throughout that each item is always assigned to a maximum "bang-per-buck" agent, i.e., agents i such that the ratio $u_{i,j}/p_j$ is maximized. In our WBB algorithm, we instead adjust a uniform bid b_i that agent i makes for *all* items, but then each item is assigned based on maximum *weighted* bang-per-buck ratios, i.e., item j is assigned to the agent that maximizes $(\eta_i u_{i,j})/b_i$. To the best of our knowledge, weighted bang-per-buck ratios have yet to be considered in the context of algorithm design for Asymmetric Nash Welfare. We hope this concept and interpretation prove useful for making progress in the challenging non-smooth case.

[3] One can verify that the algorithm in (Garg, Kulkarni, and Kulkarni 2020) still has an approximation of $\Omega(n)$ when each agent gets an additional copy of her most-valued item at the outset.

Application: Piecewise-linear Valuations. As an additional application, we apply our techniques to instances with *piecewise-linear valuations*. Such functions have been considered in a variety of settings in computational economics, since a continuous concave function can be closely approximated by one that is piecewise-linear. Such examples include computing market equilibria (Vazirani and Yannakakis 2011; Garg et al. 2015), Nash Welfare Maximization (Anari et al. 2018; Chaudhury et al. 2018), and mixed manna allocations (Chaudhury et al. 2021). To the best of our knowledge, the approximability of welfare maximization for such functions has yet to be studied.

In the context of ICA, piecewise-linear valuations are formally defined as follows: each agent valuation function $v_i(\cdot)$ is defined over a sequence of conjoined line segments. Let $x_{i,k}$ denote the transition point on the x-axis between the k-th and $(k+1)$-th segments (where $x_{i,0} = 0$). For any such function, we give an algorithm that achieves an approximation ratio of at most $(1+\epsilon)4/3$, as long as the maximum utility gained for a single item is at most the length (along the x-axis) of any segment of the piecewise-linear function. (Proof in full version.)

Theorem 4. *Consider an ICA instance where $v_i(u_i)$ is a linear piecewise function such that $\min_k (x_{i,k+1} - x_{i,k}) \geq \max_j u_{ij}$. Then there exists an algorithm whose approximation ratio is $(1+\epsilon)\max_i \mu_i \leq (1+\epsilon)4/3$.*

For the special case of budget-additive functions with utility caps c_i, the condition required by Theorem 4 is equivalent to the standard assumption that $\max_j u_{i,j} \leq c_i$. Thus, our bound essentially (i.e., barring the $4/3-\delta$ approximation for a small constant δ via the configuration LP in (Kalaitzis et al. 2015)) matches the state-of-the-art approximation for budget-additive valuations but in the more general setting of piecewise-linear functions.

Related Work

There is a long line of work that examined the approximability of welfare maximization for the special case of budget-additive functions. A series of results (Garg, Kumar, and Pandit 2001; Andelman and Mansour 2004; Azar et al. 2008; Chakrabarty and Goel 2010; Kalaitzis et al. 2015) improved the best approximation to $4/3-\delta$ for a small constant δ, while the hardness lower bound is currently $16/15$ (Chakrabarty and Goel 2010). Budget-additive valuations have also been heavily studied in the online setting, typically called the *Adwords* problem. Please see the excellent survey by Mehta (2013) for an overview of this area. We note the algorithm in (Devanur and Jain 2012) for concave-additive valuations with fractional items is viewed as the state of the art in the online setting. Their approximation ratio is also expressed as a parameter that depends on the curvature of the given concave functions.

Since the seminal result in (Cole and Gkatzelis 2015), the symmetric Nash welfare objective has been extensively studied for a variety of valuation classes (Barman, Krishnamurthy, and Vaish 2018; Cole et al. 2017; Garg, Hoefer, and Mehlhorn 2018; Anari et al. 2018; Garg, Husić, and Végh 2021a; Li and Vondrák 2021). For the asymmetric objective, (Garg, Kulkarni, and Kulkarni 2020) gave $O(n)$ approximations for additive and budget-additive valuations. More recently, (Garg, Husić, and Végh 2021b) showed approximations parameterized by the max-to-min ratio of agent weights, and (Garg et al. 2021) gave a PTAS for the cases of identical and two-value agents.

Nash welfare has also been extensively studied for its appealing fairness properties; for example, Caragiannis et al. (2019) call the objective “unreasonably fair.” For additional results on fairness, see (Barman, Krishnamurthy, and Vaish 2018; Plaut and Roughgarden 2020; McGlaughlin and Garg 2020). Smooth valuations have also been considered primarily in the context of fairness (Fain, Munagala, and Shah 2018; Fluschnik et al. 2019; Barman et al. 2022).

Preliminaries

Convex Program Formulation. Recall that every agent i has a non-decreasing concave valuation function $v_i(\cdot)$. Our algorithm utilizes the natural assignment convex program for the problem, which we will refer to as ICA-CP:

$$(\text{ICA-CP}) : \max \sum_i v_i(u_i)$$
$$\forall i : u_i = \sum_j u_{i,j} x_{ij}$$
$$\forall j : \sum_i x_{i,j} \leq 1$$
$$\forall i,j : x_{i,j} \geq 0$$

The algorithm is primal-dual in nature, and uses the dual program which was defined for the online variant of the problem in (Devanur and Jain 2012):

$$(\text{ICA-D}) : \min \sum_i y_i(t_i) + \sum_j p_j$$
$$\forall i,j : p_j \geq u_{i,j} v_i'(t_i)$$
$$\forall i,j : t_i, p_j \geq 0$$

where $y_i(t_i) = v_i(t_i) - t_i v_i'(t_i)$ is defined as the y-intercept of the tangent to v_i at t_i. Thus we have the following lemma.

Lemma 5 (shown in (Devanur and Jain 2012))**.** *The above convex programs form a primal-dual pair. That is, any feasible solution to* ICA-D *has objective at least that of any feasible solution to* ICA-CP.

Local Curvature Parameters. Both the definition and guarantees provided by our algorithm depend on parameters that measure the local curvatures of each agent valuation function, both in multiplicative (we will use μ) and additive senses (α). To define these parameters, let

$$\sigma_i(z,w) := \frac{v_i(z+w) - v_i(z)}{w}, \qquad (1)$$

be the slope of the lower-bounding secant line that intersects v_i at points $(z, v_i(z))$ and $(z+w, v_i(z+w))$. Define the *local multiplicative curvature* of a function v_i at point z with x-width $w > 0$ to be:

$$\mu_i(z,w) := \max_{z^* \in (0,w)} \left[\frac{v_i(z+z^*)}{v_i(z) + z^* \sigma_i(z,w)} \right]. \qquad (2)$$

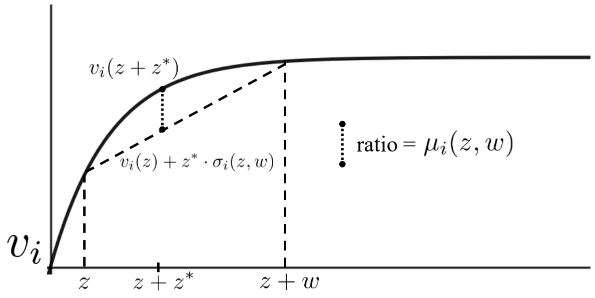

Figure 1: Illustration of the definition of the multiplicative local curvature at point z with width w for function v_i (denoted $\mu_i(z,w)$).

Informally, $\mu_i(z,w)$ measures the largest multiplicative gap between a the function evaluated at $z+z^*$ and the corresponding point on the lower bounding secant line; see Figure 1. The overall local multiplicative curvature for agent i is then defined to be $\mu_i := \max_{z,u_{i,j}} \mu_i(z,u_{i,j})$.

Similarly, we define the *local additive curvature* for an agent at point z with x-width w as

$$\alpha_i(z,w) := \max_{z^* \in (0,w)} \left[v_i(z+z^*) - (v_i(z) + z^* \sigma_i(z,w)) \right],$$

and let $\alpha_i := \max_{z,u_{i,j}} \alpha_i(z,u_{i,j})$.

General Algorithm for ICA

In this section, we define our $(1+\epsilon)\max_i \mu_i$-approximation algorithm for ICA, assuming that the algorithm has knowledge of the values of the μ_i for any given set of valuation functions. (In the full version, we discuss how this assumption can be removed.)

Algorithm Definition. As was done for the algorithm in (Chakrabarty and Goel 2010) for budget-additive functions, it will be useful to partition the cost of the dual solution according to algorithm's current assignments. Let t_i be the current dual variable maintained by the algorithm for agent i. Define

$$D(u_i) := y_i(t_i) + u_i v_i'(t_i) \tag{3}$$

to be the utility for agent i but instead evaluated according to the tangent line in the dual objective taken at point t_i. The algorithm will maintain that at any point, each item j will be assigned to the bidder i that maximizes $u_{i,j}v_i'(t_i)$ (and will reassign an item if this doesn't hold). Call such an assignment a *proper assignment.* In an allocation where all item's are properly assigned, we can obtain the following characterization of the dual objective (proof in full version).

Lemma 6. *Fix a point in the algorithm with primal and dual variables u_i and t_i for each agent i. If all items are properly allocated, then (i) setting $p_j = \max_i v_i'(t_i)u_{i,j}$ forms a feasible solution to the dual program* ICA-*D, and (ii) the objective of the dual can be expressed as $\sum_i D(u_i)$.*

We can now define our algorithm, given formally in Algorithm 1. The algorithm maintains a setting of the dual variable t_i for every agent i and p_j for every item j. Each t_i variable is initialized to be 0, and then items are properly assigned accordingly. The algorithm proceeds by continuously increasing t_i values (thus decreasing $v_i'(t_i)$), allowing items to defect if they are no longer properly assigned. The goal of the algorithm is to eventually is obtain an allocation such that $D(u_i)/v_i(u_i) \leq \mu_i$ for all agents. Under this condition, as long items remain (close to) properly assigned upon the algorithm's termination, Lemmas 5 and 6 imply that the approximation ratio of the algorithm is at most $\max_i \mu_i$.

Algorithm 1 Multiplicative Algorithm for ICA

```
 1: Initialize all $t_i = 0$
 2: Allocate each item $j$ to agent $\arg\max_i (u_{i,j}v_i'(0))$
 3: while $\exists$ an agent $i$ such that $D(u_i)/v_i(u_i) > \mu_i$ do
 4:     while $D(u_i)/v_i(u_i) > \mu_i$ do
 5:         if there is an item $j$ such that $j$ is assigned to $i$ and
            $i \neq \arg\max_k (u_{k,j}v_k'(t_k))$ then
 6:             Reassign $j$ to agent $\arg\max_k (u_{k,j}v_k'(t_k))$
 7:         else
 8:             Increase tangent point $t_i$ until $v_i'(t_i)$ decreases
                by a factor of $1/(1+\epsilon)$
 9:         end if
10:     end while
11: end while
12: Output resulting allocation $u_i$ for all agents
```

Algorithm Analysis. The algorithm seeks to find an allocation such that, for every agent i, the inequality $D(u_i)/v_i(u_i) \leq \mu_i$ is satisfied. To this end, we use the following terminology.

Definition 7. *There are two types of agents such that $D(u_i)/v_i(u_i) > \mu_i$: either $u_i < t_i$ or $u_i > t_i$. Call such agents under allocated and over allocated, respectively.*

The main technical hurdles for our more general setting of concave-additive functions are showing that no agent becomes under allocated, and establishing the claimed run time bound. We first state and prove a useful a property of monotone-concave functions (proof in full version).

Lemma 8. *Let $f : \mathbf{R}_+ \to \mathbf{R}_+$ be a monotone concave function. Suppose $\ell_1(\cdot)$ and $\ell_2(\cdot)$ are the equations of two lines tangent to f at points $(t_1, f(t_1))$ and $(t_2, f(t_2))$, respectively. If there exists $x \geq \max(t_1,t_2)$ such that $\ell_2(x) \leq \ell_1(x)$, then $\ell_1(\tilde{x}) \leq \ell_2(\tilde{x})$ for all $\tilde{x} \leq t_1$.*

We now establish that throughout the algorithm's execution, at no point does an agent become under allocated.

Lemma 9. *Throughout the algorithm, an agent never becomes under allocated. In particular, if $u_i < t_i$ then $D(u_i)/v_i(u_i) \leq \mu_i$.*

Proof. At the start of the algorithm $t_i = 0$ for all agents, and so no agent can be under allocated at the outset of the algorithm. Therefore, the only point at which an agent i with total utility u_i could potentially become under allocated is when some item j is reassigned to another agent on Line 6 of the algorithm such that after the reassignment $u_i - u_{i,j} < t_i$. Fix such a point in the algorithm.

Let $s(x)$ denote the equation for the secant line that passes through $v_i(\cdot)$ at points $(u_i - u_{i,j}, v_i(u_i - u_{i,j}))$ and $(u_i, v_i(u_i))$. More formally, let $\sigma_i' := \sigma_i(u_i - u_{i,j}, u_{i,j})$, where the definition of $\sigma_i(\cdot)$ is given by Equation (1). Then the equation for $s(x)$ is

$$s(x) = \sigma_i' x + v_i(u_i - u_{i,j}) - \sigma_i'(u_i - u_{i,j}). \tag{4}$$

Let $\tilde{\mu}_i := \tilde{\mu}_i(u_i - u_{i,j})$ denote the local multiplicative curvature of $v_i(\cdot)$ at $u_i - u_{i,j}$, and let z^* be value (given in Equation (2)) that determines $\tilde{\mu}_i$. Based on the definition of $\tilde{\mu}_i$, observe that if we scale $s(x)$ by a factor of $\tilde{\mu}_i$, we obtain an equation for a line that is tangent to $v_i(\cdot)$ at the point $(u_i - u_{i,j} + z^*, v_i(u_i - u_{i,j} + z^*))$. Denote the equation for this line as $\tilde{s}(x) = \tilde{\mu}_i s(x)$.

Since by definition $s(u_i) = v_i(u_i)$, it follows that

$$\tilde{s}(u_i) = \tilde{\mu}_i s(u_i) = \tilde{\mu}_i v_i(u_i) \leq \mu_i v_i(u_i) < D(u_i), \tag{5}$$

where the first inequality holds by definition of μ_i and the last inequality holds because i entered the loop on Line 3.

Given Inequality (5), it now follows from Lemma 8 that $D(u_i - u_{i,j}) \leq \tilde{s}(u_i - u_{i,j})$ by setting $\ell_1(\cdot) = D(\cdot)$, $\ell_2(\cdot) = \tilde{s}(\cdot)$, and $x = u_i$. This inequality implies:

$$\frac{D(u_i - u_{i,j})}{v_i(u_i - u_{i,j})} = \frac{D(u_i - u_{i,j})}{s(u_i - u_{i,j})} \leq \frac{\tilde{s}(u_i - u_{i,j})}{s(u_i - u_{i,j})} = \tilde{\mu}_i \leq \mu_i, \tag{6}$$

where first equality follows from the definition of $s(x)$. Thus agent i did not become under allocated after the reassignment of item j, as desired. □

The next lemma establishes the run time of the algorithm (proof in full version). For simplicity, we will assume that ϵ is selected such that for all agents $\mu_i \geq 1+\epsilon$. This is possible as long as $\mu > 1$. (If $\mu_i = 1$, then $v_i(u_i) = D(u_i)$, and thus the agent is never under or over allocated.)

Lemma 10. *Let $u_i^{\max} = \sum_j u_{i,j}$ denote the maximum possible utility for a fixed agent i, and let $\rho_{\max} = \max_i (v_i'(0) u_i^{\max} / v_i(u_i^{\max}))$. If for all agents $\mu_i \geq 1 + \epsilon$, then the algorithm terminates in time $O(mnT \ln(\rho_{\max}/\epsilon)/\epsilon)$, where T is the time needed to perform the update of t_i on Line 8.*

By combining these lemmas, we can prove Theorem 1; the proof is in the full version.

Adaptation to Additive Guarantee An appealing feature of our geometric-based arguments is that they easily extend to obtain additive guarantees as well. This result is stated formally as follows . The details of this adaptation are given in the full version of the paper.

Theorem 11. *There exists an algorithm for ICA that achieves an additive bound of $\sum_i \alpha_i + \epsilon$ and runs in time $O(nm^2 T v_i'(0)/\epsilon)$, where T is the time needed to perform the update of t_i on Line 8 of Algorithm 1. Furthermore, the integrality gap of the assignment convex program is $\sum_i \alpha = n\alpha$ for instances with additive curvature α.*

Extension to Smooth Asymmetric Nash Welfare Maximization

We now apply our techniques to Nash Welfare Maximization for asymmetric agents with smooth additive additive valuations. In this problem, each agent i has a weight $\eta_i > 0$, and the goal is to find an allocation that maximizes $(\prod_i (u_i + \omega)^{\eta_i})^{1/\eta}$ where $\eta = \sum_i \eta_i$ is the sum of the agent weights and $\omega \in (0, 1]$ denotes the smoothing parameter of the instance. Recall that we can scale the objective of each agent i by $(\max_j u_{i,j})^{-\eta_i}$ without changing the approximation ratio of the algorithm, and thus wlog we assume that $\max_j u_{i,j} = 1$ for every agent i. To simplify notation, we also assume weights are normalized by dividing them by η, so $\eta = 1$ (i.e., we bring the $1/\eta$ exponent into each term in the product objective).

Algorithm Definition. Our algorithm has a combinatorial interpretation which we call the *Weighted Bang-Per-Buck* (WBB) algorithm. To define the algorithm, we first explicitly define the additive curvature parameter α_i in the case where $v_i(u_i) = \eta_i \ln(u_i + \omega)$. Let $\sigma_i(z, 1)$ denote the slope of the lower-bounding secant line that intersects the points $(z, \eta_i \ln(z + \omega))$ and $(z + 1, \eta_i \ln(z + \omega + 1))$, given as:

$$\begin{aligned}\sigma_i(z, 1) &:= \eta_i \ln(z + \omega + 1) - \eta_i \ln(z + \omega) \\ &= \eta_i \ln\left(1 + (z + \omega)^{-1}\right). \end{aligned} \tag{7}$$

We then define the local additive curvature bound α_i at z for agent i:

$$\begin{aligned}\alpha_i(z) := \max_{z^* \in (0,1)} [&\eta_i \ln(z + z^* + \omega) - \\ &(\eta_i \ln(z + \omega) + z^* \sigma_i(z, 1))]. \end{aligned} \tag{8}$$

Thus from the definition of α_i (given in the preliminaries), we have $\alpha_i = \max_i \alpha_i(z)$ since $\max_j u_{i,j} = 1$.

The WBB Algorithm is given in Algorithm 2. Throughout its execution, we adjust a uniform bid b_i each agent i makes for on every item in the instance. The algorithm starts with bids that are underestimates of the optimal dual bids, and thus proceeds by increasing the uniform bid of each agent one at a time, ensuring throughout that every item is assigned to a maximum weighted bang-per-buck ratio agent, i.e., an agent that maximizes $(\eta_i u_{i,j})/b_i$. The algorithm stops increasing the bid of an agent according an exponential potential function proportional to agent's average *unweighted* MBB ratio.

Analysis. We first argue that the WBB algorithm is equivalent to executing the ICA algorithm for an additive guarantee (given by Theorem 11). We then derive a closed-form for the local additive curvature α_i in terms of the smoothing parameter ω.

Lemma 12. *The WBB algorithm (Algorithm 2) is equivalent to executing the ICA algorithm for an additive guarantee, where in the ICA instance $v_i(u_i) = \eta_i \ln(u_i + \omega)$.*

Proof. By Lemma 5, the dual program for an ICA instance with $v_i(u_i) = \eta_i \ln(u_i + \omega)$ is given by the following concave program:

Algorithm 2 Weighted Bang-per-buck Algorithm (WBB)

```
1:  Initialize fixed bid $b_i \leftarrow \omega$ for each agent $i$.
2:  Allocate each item $j$ to $\arg\max_i \left(\frac{\eta_i u_{i,j}}{b_i}\right)$
3:  while there exists an agent $i$ such that
        $(u_i + \omega)/b_i < \exp((u_i + \omega)/b_i - 1 - \alpha_i)$ do
4:      while $(u_i + \omega)/b_i < \exp((u_i + \omega)/b_i - 1 - \alpha_i)$ do
5:          if there is an item $j$ assigned to agent $i$ such that $i$
            is not $j$'s maximum WBB agent then
6:              Reassign $j$ to agent $\arg\max_k \left(\frac{\eta_k u_{k,j}}{b_k}\right)$
7:          else
8:              Increase agent $i$'s bid to be $b_i \leftarrow \frac{\eta_i m b_i}{\eta_i m - \epsilon b_i}$
9:          end if
10:     end while
11: end while
12: Output resulting allocation $u_i$ for all agents
```

$$\text{(ASN-D)}: \min \sum_i \eta_i \left(\ln(t_i + \omega) - \frac{t_i}{t_i + \omega}\right) + \sum_j \beta_j$$
$$\forall i, j : \beta_j \geq \frac{\eta_i u_{i,j}}{t_i + \omega}, \quad \forall i, j : t_i, \beta_j \geq 0$$

Note that for this application, we denote the dual variable p_j as β_j, since it is interpreted as the weighted MBB ratio, not the price. In particular, in the WBB algorithm, we substitute the $t_i + \omega$ terms in the ASN-D program to be the uniform bid b_i made by agent i for all items. Thus the function $D(u_i)$ becomes:

$$D(u_i) = \eta_i \left[\frac{u_i + \omega}{b_i} + \ln(b_i) - 1\right].$$

Substituting this into the while-loop condition of our general additive algorithm[4] and simplifying, we obtain the while-loop condition in Algorithm 2. Furthermore, since the WBB algorithm maintains an assignment where each item is assigned to the agent with maximum weighted bang-per-buck ratio, the variables $t_i = b_i - \omega$ and $\beta_j = \arg\max_i \left(\frac{\eta_i u_{i,j}}{b_i}\right)$ form a feasible dual solution. Finally, one can verify the update to bid b_i decreases $v_i'(t_i)$ by ϵ/m. □

The next lemma is proved in the full version of the paper.

Lemma 13. *When valuation function of agent i is* $v_i(u_i) = \eta_i \ln(u_i + \omega)$*, then* $\alpha_i = O\left(\eta_i \ln\left(\frac{1}{\omega \ln(1+1/\omega)}\right)\right)$.

We can now prove Theorem 3.

Proof of Theorem 3. By Theorem 11, the algorithm achieves the desired run-time bound, since $v_i'(0) = \eta_i/\omega \leq 1/\omega$ (recall we normalized agent weights to be η_i/η) and the update on Line 8 in WBB takes $O(1)$ time. By Lemma 12 and Theorem 11, the algorithm achieves an additive approximation of $\sum_i \alpha_i$ (where α_i is given by Lemma 13) on the log objective. Thus, we are left with bounding the multiplicative approximation of the algorithm for the product objective.

Let OPT and OPT_{log} denote the objective value of the optimal solution for the product and log objective, respectively. Consider the dual variables (β_j, t_i) corresponding to the allocation returned by the algorithm. By the proof of Theorem 11, $\beta_j + \epsilon/m$ is a feasible solution to the dual program ASN-D (see footnote in Lemma 12). Thus by Lemma 5, ASN-D(t, β) is bounded below by:

$$\sum_i \eta_i \left(\ln(t_i + \omega) - \frac{t_i}{t_i + \omega}\right) + \sum_j \left(\beta_j + \frac{\epsilon}{m}\right)$$
$$= \sum_i D(u_i) + \epsilon \geq \text{OPT}_{\text{log}}. \quad (9)$$

When the algorithm terminates we have $\eta_i \ln(u_i + \omega) \geq D(u_i) - \alpha_i$ for every agent i. Along with Inequality (9), this implies the total objective of the algorithm is bounded by:

$$\sum_i \eta_i \ln(u_i + \omega) \geq \sum_i (D(u_i) - \alpha_i) \geq \text{OPT}_{\text{log}} - \sum_i \alpha_i - \epsilon.$$

From this inequality, and the fact that $\text{OPT}_{\text{log}} = \ln(\text{OPT})$, it follows the algorithm's objective on the product objective $\prod_i (u_i + \omega)^{\eta_i} = \exp\left(\sum_i \ln(u_i + \omega)\right)$ is lower bounded by:

$$\exp\left(\text{OPT}_{\text{log}} - \sum_i \alpha_i - \epsilon\right) = \exp\left(-\sum_i \alpha_i - \epsilon\right) \text{OPT}.$$

From Lemma 13, we have that $\sum_i \alpha_i = O\left(\ln\left(\frac{1}{\omega \ln(1+1/\omega)}\right)\right)$, and therefore $\exp\left(\sum_i \alpha_i + \epsilon\right) = O(e^{\epsilon}/(\omega \ln(1 + 1/\omega)))$. Substituting and rearranging the bound estabalished above, we obtain the theorem. □

Conclusion

In this paper we obtain tight approximations for allocating indivisible items to agents with concave additive valuations. We conclude by proposing potential directions for future work. The focus of this paper has been on the approximability of ICA. As discussed in the introduction, several prior works have established connections between approximability and mechanisms with fairness guarantees. Therefore, it would be interesting to see if the concepts introduced in this paper (local curvature, weighted bang-per-buck ratios, etc.) can be leveraged to define formal notions of fairness.

Another interesting direction would be to examine vector monotone concave functions (i.e., valuations $v_i(\cdot)$ are monotone concave functions of the vector $\{u_{i,j} x_{i,j}\}_{i,j}$), and perhaps see if a more general notion of local curvature can be used to characterize the approximability of the problem in this setting. We also believe our techniques could point to a more general framework for deriving algorithms from the curvature of their objective functions (as highlighted by our WBB algorithm).

[4] The adaptation of Algorithm 1 that instead obtains an additive guarantee (given in Theorem 11) uses $D(u_i) - v_i(u_i) > \alpha_i$ for its while-loop condition. Similarly in the analysis, one adjusts dual variables by *adding* ϵ/m to each β_j to obtain a feasible solution.

Acknowledgments

The authors would like to thank Rupert Freeman and Debmalya Panigrahi for their helpful feedback on prior drafts of the paper. This work was completed while KS was at Duke University. He was supported in part by NSF grants CCF-1750140 (CAREER), CCF-1955703, and ARO grant W911NF2110230.

References

Ahmed, F.; Dickerson, J.; and Fuge, M. 2020. Forming diverse teams from sequentially arriving people. *Journal of Mechanical Design*, 142(11): 111401.

Anari, N.; Mai, T.; Gharan, S. O.; and Vazirani, V. V. 2018. Nash social welfare for indivisible items under separable, piecewise-linear concave utilities. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2274–2290. SIAM.

Andelman, N.; and Mansour, Y. 2004. Auctions with budget constraints. In *Scandinavian Workshop on Algorithm Theory*, 26–38. Springer.

Azar, Y.; Birnbaum, B.; Karlin, A. R.; Mathieu, C.; and Nguyen, C. T. 2008. Improved approximation algorithms for budgeted allocations. In *International Colloquium on Automata, Languages, and Programming*, 186–197. Springer.

Bai, W.; Bilmes, J.; and Noble, W. S. 2016. Bipartite matching generalizations for peptide identification in tandem mass spectrometry. In *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 327–336.

Barman, S.; Krishna, A.; Narahari, Y.; and Sadhukhan, S. 2022. Nash Welfare Guarantees for Fair and Efficient Coverage. *arXiv preprint arXiv:2207.01970*.

Barman, S.; Krishnamurthy, S. K.; and Vaish, R. 2018. Finding fair and efficient allocations. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, 557–574.

Benabbou, N.; Chakraborty, M.; Ho, X.-V.; Sliwinski, J.; and Zick, Y. 2018. Diversity constraints in public housing allocation. In *17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2018)*.

Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A. D.; Shah, N.; and Wang, J. 2019. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation (TEAC)*, 7(3): 1–32.

Chakrabarty, D.; and Goel, G. 2010. On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and GAP. *SIAM Journal on Computing*, 39(6): 2189–2211.

Chaudhury, B.; Cheung, Y. K.; Garg, J.; Garg, N.; Hoefer, M.; and Mehlhorn, K. 2018. On Fair Division of Indivisible Items. *arXiv preprint arXiv:1805.06232*.

Chaudhury, B. R.; Garg, J.; McGlaughlin, P.; and Mehta, R. 2021. Competitive allocation of a mixed manna. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1405–1424. SIAM.

Chen, X.; Dai, D.; Du, Y.; and Teng, S.-H. 2009. Settling the complexity of Arrow-Debreu equilibria in markets with additively separable utilities. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, 273–282. IEEE.

Cole, R.; Devanur, N.; Gkatzelis, V.; Jain, K.; Mai, T.; Vazirani, V. V.; and Yazdanbod, S. 2017. Convex program duality, Fisher markets, and Nash social welfare. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, 459–460.

Cole, R.; and Gkatzelis, V. 2015. Approximating the Nash social welfare with indivisible items. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, 371–380.

Conitzer, V.; Freeman, R.; and Shah, N. 2017. Fair public decision making. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, 629–646.

Degefu, D. M.; He, W.; Yuan, L.; and Zhao, J. H. 2016. Water allocation in transboundary river basins under water scarcity: a cooperative bargaining approach. *Water resources management*, 30(12): 4451–4466.

Devanur, N. R.; and Jain, K. 2012. Online matching with concave returns. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, 137–144.

Eisenberg, E.; and Gale, D. 1959. Consensus of subjective probabilities: The pari-mutuel method. *The Annals of Mathematical Statistics*, 30(1): 165–168.

Fain, B.; Munagala, K.; and Shah, N. 2018. Fair allocation of indivisible public goods. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, 575–592.

Fluschnik, T.; Skowron, P.; Triphaus, M.; and Wilker, K. 2019. Fair Knapsack. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 1941–1948.

Garg, J.; Hoefer, M.; and Mehlhorn, K. 2018. Approximating the Nash social welfare with budget-additive valuations. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2326–2340. SIAM.

Garg, J.; Husić, E.; Murhekar, A.; and Végh, L. 2021. Tractable fragments of the maximum Nash welfare problem. *arXiv preprint arXiv:2112.10199*.

Garg, J.; Husić, E.; and Végh, L. A. 2021a. Approximating Nash social welfare under Rado valuations. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 1412–1425.

Garg, J.; Husić, E.; and Végh, L. A. 2021b. Approximating Nash social welfare under Rado valuations. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 1412–1425.

Garg, J.; Kulkarni, P.; and Kulkarni, R. 2020. Approximating Nash social welfare under submodular valuations through (un) matchings. In *Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms*, 2673–2687. SIAM.

Garg, J.; Mehta, R.; Sohoni, M.; and Vazirani, V. V. 2015. A complementary pivot algorithm for market equilibrium under separable, piecewise-linear concave utilities. *SIAM Journal on Computing*, 44(6): 1820–1847.

Garg, R.; Kumar, V.; and Pandit, V. 2001. Approximation algorithms for budget-constrained auctions. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, 102–113. Springer.

Kalaitzis, C.; Madry, A.; Newman, A.; Poláček, L.; and Svensson, O. 2015. On the configuration LP for maximum budgeted allocation. *Mathematical Programming*, 154(1): 427–462.

Laruelle, A.; and Valenciano, F. 2007. Bargaining in committees as an extension of Nash's bargaining theory. *Journal of Economic Theory*, 132(1): 291–305.

Li, W.; and Vondrák, J. 2021. A constant-factor approximation algorithm for Nash Social Welfare with submodular valuations. *CoRR*, abs/2103.10536.

Lin, H.; and Bilmes, J. 2011. Word alignment via submodular maximization over matroids. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 170–175.

McGlaughlin, P.; and Garg, J. 2020. Improving Nash social welfare approximations. *Journal of Artificial Intelligence Research*, 68: 225–245.

Mehta, A. 2013. Online Matching and Ad Allocation. *Foundations and Trends® in Theoretical Computer Science*, 8(4): 265–368.

Plaut, B.; and Roughgarden, T. 2020. Almost envy-freeness with general valuations. *SIAM Journal on Discrete Mathematics*, 34(2): 1039–1068.

Vazirani, V. V. 2010. Spending constraint utilities with applications to the adwords market. *Mathematics of Operations Research*, 35(2): 458–478.

Vazirani, V. V.; and Yannakakis, M. 2011. Market equilibrium under separable, piecewise-linear, concave utilities. *Journal of the ACM (JACM)*, 58(3): 1–25.

Yu, S.; van Ierland, E.; Weikard, H.-P.; and Zhu, X. 2017. Nash bargaining solutions for international climate agreements under different sets of bargaining weights. *International Environmental Agreements: Politics, Law and Economics*, 17(5): 709–729.