Lexicographic Optimization: Algorithms and Stability

Jacob Abernethy
Google Research
Georgia Institute of Technology

Robert E. Schapire
Microsoft Research

Umar Syed
Google Research

Abstract

A lexicographic maximum of a set $X \subseteq \mathbb{R}^n$ is a vector in X whose smallest component is as large as possible, and subject to that requirement, whose second smallest component is as large as possible, and so on for the third smallest component, etc. Lexicographic maximization has numerous practical and theoretical applications, including fair resource allocation, analyzing the implicit regularization of learning algorithms, and characterizing refinements of game-theoretic equilibria. We prove that a minimizer in X of the exponential loss function $L_c(\mathbf{x}) = \sum_i \exp(-cx_i)$ converges to a lexicographic maximum of X as $c \to \infty$, provided that X is *stable* in the sense that a well-known iterative method for finding a lexicographic maximum of X cannot be made to fail simply by reducing the required quality of each iterate by an arbitrarily tiny degree. Our result holds for both near and exact minimizers of the exponential loss, while earlier convergence results made much stronger assumptions about the set X and only held for the exact minimizer. We are aware of no previous results showing a connection between the iterative method for computing a lexicographic maximum and exponential loss minimization. We show that every convex polytope is stable, but that there exist compact, convex sets that are not stable. We also provide the first analysis of the convergence rate of an exponential loss minimizer (near or exact) and discover a curious dichotomy: While the two smallest components of the vector converge to the lexicographically maximum values very quickly (at roughly the rate $\frac{\log n}{c}$), all other components can converge arbitrarily slowly.

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

1 INTRODUCTION

A *lexicographic maximum* of a set $X \subseteq \mathbb{R}^n$ is a vector in X that is at least as large as any other vector in X when sorting their components in non-decreasing order and comparing them lexicographically. For example, if

$$X = \left\{ \mathbf{x}_1 = \begin{pmatrix} 5 \\ 2 \\ 4 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} 2 \\ 6 \\ 3 \end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix} 8 \\ 7 \\ 1 \end{pmatrix} \right\}$$

then the lexicographic maximum of X is $\mathbf{x}_1$, since both $\mathbf{x}_1$ and $\mathbf{x}_2$ have the largest smallest components, and $\mathbf{x}_1$ has a larger second smallest component than $\mathbf{x}_2$. Infinite and unbounded sets can also contain a lexicographic maximum.

One of the first applications of lexicographic maximization was fair bandwidth allocation in computer networks (Hayden, 1981; Mo and Walrand, 1998; Le Boudec, 2000). Lexicographic maximimization avoids a 'rich-get-richer' allocation of a scarce resource, since a lexicographic maximum is both Pareto optimal and has the property that no component can be increased without decreasing another *smaller* component. More recently, Diana et al. (2021) considered lexicographic maximization as an approach to fair regression, where the components of the lexicographic maximum represent the performance of a model on different demographic groups. Bei et al. (2022) studied the "cake sharing" problem in mechanism design and showed that assigning a lexicographically maximum allocation to the agents is a truthful mechanism.

Rosset et al. (2004) and Nacson et al. (2019) used lexicographic maximization to analyze the implicit regularization of learning algorithms that are based on minimizing an objective function. In particular, they showed for certain objective functions and model classes that the vector of model predictions converges to a lexicographic maximum.

In game theory, Dresher (1961) described an equilibrium concept in which each player's payoff vector is a lexicographic maximum of the set of their possible payoff vectors. Van Damme (1991) showed that in a zero-sum game this concept is equivalent to a *proper equilibrium*, a well-known refinement of a Nash equilibrium. Lexicographic maximization is also used to define the *nucleolus* of a cooperative game (Schmeidler, 1969).

Two methods for computing a lexicographic maximum have been described in the literature. The first method is an iterative algorithm that solves n optimization problems, with the ith iteration computing the ith smallest component of the lexicographic maximum, and is guaranteed to find the lexicographic maximum if one exists. This method is often called the *progressive filling algorithm* (Bertsekas and Gallager, 2021). The second method minimizes the *exponential loss*, which is defined

$$L_c(\mathbf{x}) = \sum_{i=1}^{n} \exp(-cx_i), \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $c \geq 0$. Previous work has shown that a minimizer in X of $L_c(\mathbf{x})$ may converge to a lexicographic maximum of X as $c \to \infty$, but the conditions placed on X to ensure convergence were quite restrictive. Mo and Walrand (1998), Le Boudec (2000) and Rosset et al. (2004) all assumed that X is a convex polytope, while Nacson et al. (2019) assumed that X is the image of a simplex under a continuous, positive-homogeneous mapping. These results were all based on asymptotic analyses, and did not establish any bounds on the rate of convergence to a lexicographic maximum. As far as we know, no previous work has drawn a connection between the two methods for computing a lexicographic maximum.

Our contributions: The iterative algorithm described above is guaranteed to find a lexicographic maximum of a set if the optimization problems in each iteration are solved exactly. But if the optimization problems are solved only approximately, then the output of the algorithm can be far from a lexicographic maximum, even if the approximation error is arbitrarily small (but still non-zero). We define a property called *lexicographic stability*, which holds for a set $X \subseteq \mathbb{R}^n$ whenever this pathological situation does not occur, and prove that it has an additional powerful implication: Any vector $\mathbf{x}_c \in X$ that is less than $\exp(-O(c))$ from the minimum value of $L_c(\mathbf{x})$ converges to a lexicographic maximum of X as $c \to \infty$. By proving convergence for all non-pathological sets, we significantly generalize existing convergence criteria for exponential loss minimization.

We show that all convex polytopes are stable, thereby subsuming most of the previous works mentioned above. On the other hand, we show that sets that are convex and compact but not polytopes need not be stable, in general. We also show that our convergence result does not hold generally when stability is not assumed by constructing a set X that is not lexicographically stable, and for which $\mathbf{x}_c$ is bounded away from a lexicographic maximum of X for all sufficiently large c, even if $\mathbf{x}_c$ is an exact minimizer of the exponential loss.

We also study the rate at which $\mathbf{x}_c$ approaches a lexicographic maximum, and find a stark discrepancy for different components of $\mathbf{x}_c$. The smallest and second smallest components of $\mathbf{x}_c$ are never more than $O\left(\frac{\log n}{c}\right)$ below their lexicographically maximum values, even if X is not lexicographically stable. However, all other components of $\mathbf{x}_c$ can remain far below their lexicographically maximum values for arbitrarily large c, even if $\mathbf{x}_c$ is an exact minimizer of the exponential loss and X is lexicographically stable with a seemingly benign structure (it can be a single line segment).

Finally, we prove that the multiplicative weights algorithm (Littlestone and Warmuth, 1994; Freund and Schapire, 1999) is guaranteed to converge to the lexicographic maximum of a convex polytope, essentially because it minimizes the exponential loss. While Syed (2010) proved that the multiplicative weights algorithm diverges from the lexicographic maximum when the learning rate is constant, we guarantee convergence by setting the learning rate to $O(1/t)$ in each iteration t.

Additional related work: Lexicographic maximization is often applied to multi-objective optimization problems where, for a given function $\mathbf{f} : \Theta \to \mathbb{R}^n$, the goal is to find $\theta^* \in \Theta$ such that $\mathbf{f}(\theta^*)$ is a lexicographic maximum of the set $X = \{\mathbf{f}(\theta) : \theta \in \Theta\}$ (Luss and Smith, 1986). As in our work, this approach sorts the components of each vector before performing a lexicographic comparison, which contrasts with other work in which an ordering of the dimensions is fixed in advance (Sherali and Soyster, 1983; Martínez-Legaz and Singer, 1987).

Diana et al. (2021) and Henzinger et al. (2022) critiqued the lexicographic maximum as a fairness solution concept because of its potential for instability when subject to small perturbations. They gave examples demonstrating that instability can occur, but did not relate the instability to the problem of computing a lexicographic maximum via loss minimization. The primary objective of our work is to provide a much fuller characterization of this instability and to explore its implications.

In the analytical framework used by Rosset et al. (2004) and Nacson et al. (2019), larger values of c for the exponential loss $L_c(\mathbf{x})$ correspond to weaker explicit regularization by a learning algorithm, so understanding the behavior of the minimizer of $L_c(\mathbf{x})$ as $c \to \infty$ helps to characterize the algorithm's implicit regularization.

Hartman et al. (2023) appear to contradict one of our key negative results by showing that if the iterative algorithm uses an approximate solver in each iteration then its output will always be close to a lexicographic maximum, provided that the approximation error is sufficiently small (see their Theorem 9). However, this apparent discrepancy with our results is in fact due only to us using a different and incompatible definition of "closeness." See Appendix B for a discussion.

2 PRELIMINARIES

For any non-negative integer n let $[n] := \{1, \ldots, n\}$, and note that $[0]$ is the empty set. Let $\mathbb{R}_{\geq 0} := \{x \in \mathbb{R} : x \geq 0\}$ be the non-negative reals. Let x_i be the ith component of the vector $\mathbf{x} \in \mathbb{R}^n$. Let $\|X\|_\infty := \sup_{\mathbf{x} \in X} \max_i |x_i|$ be the largest ℓ_∞ norm of any vector in $X \subseteq \mathbb{R}^n$.

For each $i \in [n]$ let $\sigma_i : \mathbb{R}^n \to \mathbb{R}$ be the *ith sorting function*, such that $\sigma_i(\mathbf{x})$ is the ith smallest component of $\mathbf{x} \in \mathbb{R}^n$. For example, if $\mathbf{x} = (2, 1, 2)^\top$ then $\sigma_1(\mathbf{x}) = 1$, $\sigma_2(\mathbf{x}) = 2$ and $\sigma_3(\mathbf{x}) = 2$. We define a total order $\geq_\sigma$ on vectors in $\mathbb{R}^n$ as follows: for any points $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$ we say that $\mathbf{x} \geq_\sigma \mathbf{x}'$ if and only if $\mathbf{x} = \mathbf{x}'$ or $\sigma_i(\mathbf{x}) > \sigma_i(\mathbf{x}')$ for the smallest $i \in [n]$ such that $\sigma_i(\mathbf{x}) \neq \sigma_i(\mathbf{x}')$.

Definition 1. *A* lexicographic maximum *of a set $X \subseteq \mathbb{R}^n$ is a vector $\mathbf{x}^* \in X$ for which $\mathbf{x}^* \geq_\sigma \mathbf{x}$ for every $\mathbf{x} \in X$. Let $\operatorname{lexmax} X$ be the set of all lexicographic maxima of X.*

While $\operatorname{lexmax} X$ can be empty, this can only occur if X is empty or not compact (see Theorem 1).

For notation, we always write $\mathbf{x}^*$ to denote a lexicographic maximum of X. Also, for all $c, \gamma \geq 0$ we write $\mathbf{x}_{c,\gamma}$ to denote an arbitrary vector in X that satisfies

$$L_c(\mathbf{x}_{c,\gamma}) \leq \inf_{x \in X} L_c(\mathbf{x}) + \gamma \exp(-c \|X\|_\infty).$$

In other words, $\mathbf{x}_{c,\gamma}$ is a near minimizer in X of the exponential loss if γ is small, and $\mathbf{x}_{c,0}$ is an exact minimizer. The notation $\mathbf{x}^*$ and $\mathbf{x}_{c,\gamma}$ suppresses the dependence on X, which will be clear from context. While $\mathbf{x}^*$ and $\mathbf{x}_{c,0}$ do not exist in every set X, we are only interested in cases where they do, and implicitly make this assumption throughout our analysis. An exception is when we construct a set X to be a counterexample. In these cases we explicitly prove that $\mathbf{x}^*$ and $\mathbf{x}_{c,0}$ exist. Also, a set may contain multiple vectors that satisfy the definitions of $\mathbf{x}^*$ or $\mathbf{x}_{c,\gamma}$, and our results apply no matter how they are chosen.

Our goal is to characterize when $\mathbf{x}_{c,\gamma}$ is "close" to $\mathbf{x}^*$, where we use the following definition of closeness.

Definition 2. *The* lexicographic distortion *between $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$ with respect to $I \subseteq [n]$ is*

$$d_I(\mathbf{x} \mid \mathbf{x}') \triangleq \max_{k \in I} [\max\{0, \sigma_k(\mathbf{x}) - \sigma_k(\mathbf{x}')\}].$$

If $I = [n]$ we abbreviate $d(\mathbf{x} \mid \mathbf{x}') \triangleq d_I(\mathbf{x} \mid \mathbf{x}')$.

Lexicographic distortion is useful for quantifying the closeness of $\mathbf{x}_{c,\gamma}$ to $\mathbf{x}^*$ because $d(\mathbf{x}^* \mid \mathbf{x}_{c,\gamma}) = 0$ if and only if $\mathbf{x}_{c,\gamma} \in \operatorname{lexmax} X$. It is important to note, however, that $d_I(\cdot \mid \cdot)$ is *not* a symmetric function, and most typically $d_I(\mathbf{x} \mid \mathbf{x}') \neq d_I(\mathbf{x}' \mid \mathbf{x})$

Problem statement: We want to describe conditions on $X \subseteq \mathbb{R}^n, I \subseteq [n], c \geq 0$ and $\gamma \geq 0$ which ensure that $d_I(\mathbf{x}^* \mid \mathbf{x}_{c,\gamma})$ is almost equal to zero. We are particularly interested in cases where $I = [n]$, since this implies that $\mathbf{x}_{c,\gamma}$ is close to a lexicographic maximum. Also, we want to identify counterexamples where $d_I(\mathbf{x}^* \mid \mathbf{x}_{c,\gamma})$ is far from zero. Since $d(\mathbf{x}^* \mid \mathbf{x}_{c,\gamma}) \geq d_I(\mathbf{x}^* \mid \mathbf{x}_{c,\gamma})$ for all $I \subseteq [n]$, this implies that $\mathbf{x}_{c,\gamma}$ is far from a lexicographic maximum.

3 BASIC PROPERTIES

We prove several basic properties of lexicographic maxima that will be useful in our subsequent analysis. We first show that conditions which suffice to ensure that a subset of $\mathbb{R}$ contains a maximum also ensure that a subset of $\mathbb{R}^n$ contains a lexicographic maximum.

Theorem 1. *If $X \subseteq \mathbb{R}^n$ is non-empty and compact then $\operatorname{lexmax} X$ is non-empty.*

Proof. Define $X_0, \ldots, X_n$ and $\Sigma_1, \ldots, \Sigma_n$ recursively as follows: Let $X_0 = X$, $\Sigma_i = \{\sigma_i(\mathbf{x}) : \mathbf{x} \in X_{i-1}\}$ and $X_i = \{\mathbf{x} \in X_{i-1} : \sigma_i(\mathbf{x}) \geq \sup \Sigma_i\}$. We will prove by induction that each X_i is non-empty and compact, which holds for X_0 by assumption. Since $X_n = \operatorname{lexmax} X$ this will complete the proof.

By Theorem 11 in Appendix A.1, each sorting function σ_i is continuous. If X_{i-1} is non-empty and compact then Σ_i is non-empty and compact, since it is the image of a compact set under a continuous function. Therefore $\sup \Sigma_i \in \Sigma_i$. If X_{i-1} is non-empty and compact and $\sup \Sigma_i \in \Sigma_i$ then X_i is non-empty and closed, since it is the pre-image of a compact set under a continuous function. Also, X_i is bounded, since $X_i \subseteq X_{i-1}$, and therefore X_i is compact. □

Furthermore, if X is also convex, we are assured that the lexicographic maximum of X is unique.

Theorem 2. *If $X \subseteq \mathbb{R}^n$ is non-empty, compact and convex then $|\operatorname{lexmax} X| = 1$.*

Proof. Suppose X is nonempty, compact and convex. By Theorem 1, $|\operatorname{lexmax} X| \geq 1$, so it only remains to show that $|\operatorname{lexmax} X| \leq 1$. Suppose, towards a contradiction, there exist distinct points $\mathbf{x}, \mathbf{y} \in X$ that are both lexicographic maxima. We assume without loss of generality that the coordinates of $\mathbf{x}$ are sorted in nondecreasing order and, further, that on any segment of "ties" on the sorted $\mathbf{x}$, the corresponding segment in $\mathbf{y}$ is nondecreasing. Thus, for $i, j \in [n]$, if $i \leq j$ then $x_i \leq x_j$, and in addition, if $x_i = x_j$ then $y_i \leq y_j$. Since both $\mathbf{x}$ and $\mathbf{y}$ are lexicographic maxima, it follows that $\sigma_i(\mathbf{y}) = \sigma_i(\mathbf{x}) = x_i$ for $i \in [n]$.

Let k be the smallest index on which $\mathbf{x}$ and $\mathbf{y}$ differ (so that $x_i = y_i$ for $i < k$ and $x_k \neq y_k$). Let $I = [k-1]$. Then $\sigma_i(\mathbf{y}) = \sigma_i(\mathbf{x}) = x_i = y_i$ for $i \in I$. Therefore, $\sigma_k(\mathbf{y})$ is the smallest of the remaining components of $\mathbf{y}$, implying

$$y_k \geq \min\{y_k, \ldots, y_n\} = \sigma_k(\mathbf{y}) = \sigma_k(\mathbf{x}) = x_k, \quad (2)$$

and so that $y_k > x_k$.

Let $\mathbf{z} = (\mathbf{x} + \mathbf{y})/2$, which is in X since X is convex. We consider the components of $\mathbf{z}$ relative to x_k. Let $i \in [n]$.

If $i < k$ then $z_i = x_i \leq x_k$ (since $x_i = y_i$).

If $i = k$ then $z_k = (y_k + x_k)/2 > x_k$ since $y_k > x_k$.

Finally, suppose $i > k$, implying $x_i \geq x_k$. If $x_i > x_k$ then $z_i = (y_i + x_i)/2 > x_k$ since $y_i \geq x_k$ (by Eq. (2)). Otherwise, $x_i = x_k$, implying, by how the components are sorted, that $y_i \geq y_k > x_k$; thus, again, $z_i = (y_i + x_i)/2 > x_k$.

To summarize, $z_i = y_i = x_i \leq x_k$ if $i \in I$, and $z_i > x_k$ if $i \notin I$. It follows that $\sigma_i(\mathbf{z}) = \sigma_i(\mathbf{x}) = x_i$ for $i = 1, \ldots, k-1$, and that $\sigma_k(\mathbf{z}) = \min\{z_k, \ldots, z_n\} > x_k = \sigma_k(\mathbf{x})$. However, this contradicts that $\mathbf{x}$ is a lexicographic maximum. □

4 COMPUTING A LEXICOGRAPHIC MAXIMUM

Algorithm 1 below is an iterative procedure for computing a lexicographic maximum of a set $X \subseteq \mathbb{R}^n$. In each iteration k, Algorithm 1 finds a vector in X with the (approximately) largest kth smallest component, subject to the constraint that its $k-1$ smallest components are at least as large as the $k-1$ smallest components of the vector from the previous iteration. The quality of the approximation is governed by a tolerance parameter ε. Since the optimization problem in each iteration can have multiple solutions, the output of Algorithm 1 should be thought of as being selected arbitrarily from a set of possible outputs. We write $\mathcal{A}(X, \varepsilon)$ for the set of all possible outputs when Algorithm 1 is run on input (X, ε).

Algorithm 1 and close variants have been described many times in the literature. In general, the optimization problem in each iteration can be difficult to solve, especially due to the presence of sorting functions in the constraints. Consequently, much previous work has focused on tractable special cases where the optimization problem can be reformulated as an equivalent linear or convex program (Luss, 1999; Miltersen and Sørensen, 2006). Other authors have used Algorithm 1 with tolerance parameter $\varepsilon = 0$ as the definition of a lexicographic maximum (Van Damme, 1991; Nacson et al., 2019; Diana et al., 2021). Theorem 3 explains the relationship between Definition 1 and Algorithm 1 when $\varepsilon = 0$.

Theorem 3. *Let $X \subseteq \mathbb{R}^n$. A vector $\hat{\mathbf{x}} \in \mathbb{R}^n$ is a possible output of Algorithm 1 on input $(X, 0)$ if and only if $\hat{\mathbf{x}} \in \operatorname{lexmax} X$. That is, $\mathcal{A}(X, 0) = \operatorname{lexmax} X$.*

Algorithm 1: Compute a lexicographic maximum.

Input: Set $X \subseteq \mathbb{R}^n$, tolerance $\varepsilon \geq 0$.
for $k = 1, \ldots, n$ **do**
  Find an ε-optimal solution $\mathbf{x}^{(k)}$ to the optimization problem:

$$\max_{\mathbf{x} \in X} \sigma_k(\mathbf{x})$$
$$\text{subject to } \sigma_i(\mathbf{x}) \geq \sigma_i(\mathbf{x}^{(k-1)}) \text{ for all } i \in [k-1]$$

end
Return: $\mathbf{x}^{(n)}$.

Proof. Let $\mathbf{x}^* \in \operatorname{lexmax} X$. We prove by induction that in each iteration k of Algorithm 1 we have $\sigma_i(\mathbf{x}^{(k)}) = \sigma_i(\mathbf{x}^*)$ for all $i \in [k]$. Setting $k = n$ proves the theorem. For the base case $k = 1$, observe that the algorithm assigns $\mathbf{x}^{(1)} \in \arg\max_{\mathbf{x} \in X} \sigma_1(\mathbf{x})$. Therefore $\sigma_1(\mathbf{x}^{(1)}) = \sigma_1(\mathbf{x}^*)$, by the definition of $\mathbf{x}^*$. In each iteration $k > 1$ the algorithm assigns

$$\mathbf{x}^{(k)} \in \arg\max_{\mathbf{x} \in X} \sigma_k(\mathbf{x})$$
$$\text{subject to } \sigma_i(\mathbf{x}) \geq \sigma_i(\mathbf{x}^*) \text{ for all } i \in [k-1],$$

where the constraints are implied by the inductive hypothesis. Therefore $\sigma_i(\mathbf{x}^{(k)}) = \sigma_i(\mathbf{x}^*)$ for all $i \in [k]$, again by the definition of $\mathbf{x}^*$. □

While Algorithm 1 can only find a lexicographic maximum if one exists in X, we recall from Theorem 1 that this holds whenever X is non-empty and compact.

Diana et al. (2021) and Henzinger et al. (2022) proposed $\mathcal{A}(X, \varepsilon)$ (or minor variants thereof) as the definition of the ε-approximate lexicographic maxima of X. However, they observed that $\mathcal{A}(X, \varepsilon)$ may nonetheless contain vectors that are far from any lexicographic maximum, even if ε is very small (but still non-zero). In the next section we formally characterize this phenomenon, and in the rest of the paper we explore its implications.

5 LEXICOGRAPHIC STABILITY

Theorem 3 states that Algorithm 1 outputs a lexicographic maximum (assuming one exists) if the optimization problem in each iteration of the algorithm is solved exactly. In practice, the optimization problems will be solved by a numerical method up to some tolerance $\varepsilon > 0$, with smaller values of ε typically requiring longer running times. Ideally, we would like the quality of the output of Algorithm 1 to vary smoothly with ε, and if this happens for a set X then we say that X is *lexicographically stable*.

Definition 3. *A set $X \subseteq \mathbb{R}^n$ is* lexicographically stable *if for all $\delta > 0$ there exists $\varepsilon > 0$ such that for all $\hat{\mathbf{x}} \in \mathcal{A}(X, \varepsilon)$ and $\mathbf{x}^* \in \operatorname{lexmax} X$ we have $d(\mathbf{x}^* \mid \hat{\mathbf{x}}) < \delta$.*

Definition 3 says that if a set X is lexicographically stable, then in order to find an arbitrarily good approximation of its lexicographic maximum, it suffices to run Algorithm 1 with a sufficiently small tolerance parameter $\varepsilon > 0$. On the other hand, if X is not lexicographically stable, then no matter how small we make $\varepsilon > 0$ the output of Algorithm 1 can be far from a lexicographic maximum.

A sufficient (but not necessary) condition for a set $X \subseteq \mathbb{R}^n$ to be lexicographically stable is that X is a convex polytope (i.e., the convex hull of finitely many points in $\mathbb{R}^n$).

Theorem 4. *If $X \subseteq \mathbb{R}^n$ is a convex polytope then X is lexicographically stable.*

Proof sketch. We outline the main steps of the proof. Detailed justification for these steps is given in Appendix A.2. We suppose X is a convex polytope and that $\mathbf{x}^*$ is its unique lexicographic maximum.

On each round k, Algorithm 1 approximately solves an optimization problem whose value is given by some function $H_k(\mathbf{x}^{(k-1)})$ so that $\mathbf{x}^{(k)}$ must satisfy $\sigma_k(\mathbf{x}^{(k)}) \geq H_k(\mathbf{x}^{(k-1)}) - \varepsilon$. In fact, we show that a point $\mathbf{x}$ is a possible output on (X, ε) if and only if it satisfies $\sigma_k(\mathbf{x}) \geq H_k(\mathbf{x}) - \varepsilon$ for all $k \in [n]$. This function H_k can be lower-bounded by another function G_I, which, if X is a convex polytope, is concave and lower semicontinuous.

If X is not lexicographically stable, then there exists sequences $\mathbf{x}_t$ and $\varepsilon_t > 0$ with $\varepsilon_t \to 0$ and $\mathbf{x}_t \in \mathcal{A}(X, \varepsilon_t)$ such that $\mathbf{x}_t \to \hat{\mathbf{x}}$ for some $\hat{\mathbf{x}} \in X$ but $\hat{\mathbf{x}} \neq \mathbf{x}^*$. We can re-index points in X so that the components of $\hat{\mathbf{x}}$ are sorted, and furthermore, that when there are ties, they are additionally sorted according to the components of $\mathbf{x}^*$. Let k be the first component for which $\hat{x}_k \neq x^*_k$. We show this implies $G_I(\hat{\mathbf{x}}) > \hat{x}_k$. Combining facts, we then have

$$\begin{aligned}\hat{x}_k < G_I(\hat{\mathbf{x}}) &\leq \liminf_{t\to\infty} G_I(\mathbf{x}_t) \\ &\leq \liminf_{t\to\infty} H_k(\mathbf{x}_t) \leq \liminf_{t\to\infty} [\sigma_k(\mathbf{x}_t) + \varepsilon_t] = \hat{x}_k,\end{aligned}$$

a contradiction. □

Combining Theorem 4 with Theorem 5 below recovers results due to Mo and Walrand (1998), Le Boudec (2000) and Rosset et al. (2004) which show that an exponential loss minimizer in X converges to a lexicographic maximum of X when X is a convex polytope.

Fairly simple non-convex sets that contain a lexicographic maximum but are not lexicographically stable exist. We provide an example in Theorem 6, one that also shows how, in such cases, minimizing exponential loss might not lead to a lexicographic maximum.

Theorem 4 shows that a nonempty set X is lexicographically stable if it is a convex polytope, a condition that implies that X is also convex and compact. The theorem still holds, by the same proof, with the weaker requirement that X is convex, compact and *locally simplicial*, a property defined in Rockafellar (1970, page 84). However, the theorem is false, in general, if we only require that X be convex and compact.

As an example, in $\mathbb{R}^3$, let $Z = [-1, 0] \times [0, 1] \times [0, 1]$, and let

$$X = \left\{\mathbf{x} \in Z : x_1(x_3 - 1) \geq x_2^2\right\}. \tag{3}$$

This set is compact and convex, and has a unique lexicographic maximum, namely, $\mathbf{x}^* = (0, 0, 1)^\top$. For $\varepsilon \in (0, 1)$, let $\mathbf{x}_\varepsilon = \left(-\varepsilon^2, \varepsilon/2, 3/4\right)^\top$. It can be shown that $\mathbf{x}_\varepsilon \in \mathcal{A}(X, \varepsilon)$. It follows that X is not lexicographically stable since $\sigma_3(\mathbf{x}_\varepsilon) = 3/4$ for all $\varepsilon > 0$ while $\sigma_3(\mathbf{x}^*) = 1$. (Details are given in Appendix A.3.)

6 CONVERGENCE ANALYSIS OF EXPONENTIAL LOSS MINIMIZATION

In this section we study conditions under which a near or exact minimizer $\mathbf{x}_{c,\gamma} \in X$ of the exponential loss $L_c(\mathbf{x})$ converges to a lexicographic maximum $\mathbf{x}^* \in \operatorname{lexmax} X$ as $c \to \infty$. To see why convergence should be expected, note that when c is large, the dominant term of $L_c(\mathbf{x})$ corresponds to the smallest component of $\mathbf{x}$, since the function $x \mapsto \exp(-cx)$ decreases very quickly. Therefore minimizing $L_c(\mathbf{x})$ will tend to make this term as large as possible. Further, among vectors $\mathbf{x}$ that maximize their smallest component, the dominant term in $L_c(\mathbf{x})$ corresponds to the second smallest component of $\mathbf{x}$, if we ignore terms that are equal for all vectors. In general, when c is large, the magnitudes of the terms in $L_c(\mathbf{x})$ decrease sharply when they are sorted in increasing order of the components of $\mathbf{x}$, and this situation tends to favor a minimizer of $L_c(\mathbf{x})$ that is also a lexicographic maximum. Although such reasoning is intuitive, proving convergence to a lexicographic maximum can be challenging; indeed, convergence need not hold for every set X, as will be seen shortly.

6.1 Asymptotic results

Theorem 5 states our main convergence result: If X is lexicographically stable then $\mathbf{x}_{c,\gamma}$ converges to a lexicographic maximum $\mathbf{x}^*$ as $c \to \infty$, provided that $\gamma \in [0, 1)$. In other words, in the contrapositive, if a near minimizer in X of $L_c(\mathbf{x})$ fails to converge to a lexicographic maximum as $c \to \infty$, then Algorithm 1 can fail to find a good approximation of a lexicographic maximum for any tolerance $\varepsilon > 0$.

Theorem 5. *Let $X \subseteq \mathbb{R}^n$ and $\gamma \in [0, 1)$. If X is lexicographically stable then for all $\mathbf{x}^* \in \operatorname{lexmax} X$*

$$\lim_{c\to\infty} d(\mathbf{x}^* \mid \mathbf{x}_{c,\gamma}) = 0.$$

Theorem 5's requirement that $\gamma \in [0, 1)$ ensures that $\mathbf{x}_{c,\gamma}$ is less than $\exp(-c \|X\|_\infty)$ from the minimum value of $L_c(\mathbf{x})$. To see why this condition aids convergence to a lexicographic maximum, consider that the smallest possible value of any term in $L_c(\mathbf{x})$ is $\exp(-c \|X\|_\infty)$. So if the minimization error were larger than this value then $\mathbf{x}_{c,\gamma}$ may not make every term of $L_c(\mathbf{x}_{c,\gamma})$ small, which in turn could cause $\mathbf{x}_{c,\gamma}$ to be far from a lexicographic maximum.

Before proving Theorem 5, we introduce some additional notation and a key lemma. For any $X \subseteq \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^n$ and $k \in \{0\} \cup [n]$ let

$$X_k(\mathbf{x}) = \{\mathbf{x}' \in X : \sigma_i(\mathbf{x}') \geq \sigma_i(\mathbf{x}) \text{ for all } i \in [k]\} \quad (4)$$

be the set of all vectors in X whose k smallest components are at least as large as the k smallest components of $\mathbf{x}$. Note that $X_0(\mathbf{x}) = X$. Also, if $\mathbf{x}^{(k-1)}$ is the vector selected in iteration $k-1$ of Algorithm 1, then $X_{k-1}(\mathbf{x}^{(k-1)})$ is the set of feasible solutions to the optimization problem in iteration k of the algorithm. Lemma 1 below, which is key to our convergence results and proved in Appendix A.4, states that if $\mathbf{x}_{c,\gamma}$ is selected in any iteration of Algorithm 1 then it is a good solution for the next iteration when c is large.

Lemma 1. *For all $X \subseteq \mathbb{R}^n, \gamma \in [0, 1), c > 0$ and $k \in [n]$*

$$\sigma_k(\mathbf{x}_{c,\gamma}) \geq \sup_{\mathbf{x} \in X_{k-1}(\mathbf{x}_{c,\gamma})} \sigma_k(\mathbf{x}) - \frac{1}{c} \log \left(\frac{n-k+1}{1-\gamma} \right).$$

We are now ready to prove Theorem 5.

Proof of Theorem 5. Say that a vector $\mathbf{x} \in \mathbb{R}^n$ is (X, ε)-*valid* if it is a solution to all of the optimization problems in Algorithm 1 when run on input (X, ε). In other words, if Algorithm 1 is run on input (X, ε), then the algorithm can let $\mathbf{x}^{(1)} = \mathbf{x}, \mathbf{x}^{(2)} = \mathbf{x}, \ldots, \mathbf{x}^{(n)} = \mathbf{x}$. For all $c > 0$ let $\varepsilon(c) = \frac{1}{c} \log \frac{n}{1-\gamma}$. By Lemma 1 we have

$$\sigma_k(\mathbf{x}_{c,\gamma}) \geq \sup_{\mathbf{x} \in X_{k-1}(\mathbf{x}_{c,\gamma})} \sigma_k(\mathbf{x}) - \varepsilon(c)$$

for all $k \in [n]$, which immediately implies that $\mathbf{x}_{c,\gamma}$ is $(X, \varepsilon(c))$-valid. Let $\{\delta_t\}$ be a positive sequence with $\lim_{t\to\infty} \delta_t = 0$. Since $\varepsilon(c)$ is a continuous function with range $(0, \infty)$, by Definition 3 for each δ_t there exists $c_t > 0$ such that for all $\hat{\mathbf{x}} \in \mathcal{A}(X, \varepsilon(c_t))$ and $k \in [n]$ we have

$$\delta_t \geq \sigma_k(\mathbf{x}^*) - \sigma_k(\hat{\mathbf{x}}).$$

Also, since $\mathcal{A}(X, \varepsilon) \subseteq \mathcal{A}(X, \varepsilon')$ if $\varepsilon < \varepsilon'$, and $\varepsilon(c)$ is a decreasing function, we can arrange $\{c_t\}$ to be an increasing sequence with $\lim_{t\to\infty} c_t = \infty$.

Since $\mathbf{x}_{c,\gamma}$ is $(X, \varepsilon(c))$-valid we have for all $k \in [n]$

$$\delta_t \geq \sigma_k(\mathbf{x}^*) - \sigma_k(\mathbf{x}_{c_t,\gamma}).$$

By taking the limit superior of both sides we have for all $k \in [n]$

$$0 \geq \limsup_{t\to\infty} \left[\sigma_k(\mathbf{x}^*) - \sigma_k(\mathbf{x}_{c_t,\gamma})\right],$$

and therefore

$$\lim_{t\to\infty} \max_{k\in[n]} [\max\{0, \sigma_k(\mathbf{x}^*) - \sigma_k(\mathbf{x}_{c_t,\gamma})\}] = 0,$$

which proves the theorem. $\square$

The lexicographic stability requirement in Theorem 5 cannot be relaxed without risking non-convergence. Theorem 6 below constructs a lexicographically unstable set for which the exact exponential loss minimizer is bounded away from the lexicographic maximum. The set is a piecewise linear path consisting of two adjoining line segments that is bounded, closed and connected, but not convex.

Theorem 6. *For all $n \geq 8$ there exists a set $X \subseteq \mathbb{R}^n$ consisting of two line segments with a shared endpoint and satisfying $\|X\|_\infty = 1$ such that for all $\mathbf{x}^* \in \operatorname{lexmax} X$ and $c \geq 2$*

$$d(\mathbf{x}^* \mid \mathbf{x}_{c,0}) \geq d_{\{n\}}(\mathbf{x}^* \mid \mathbf{x}_{c,0}) \geq \frac{1}{2}.$$

Proof sketch. The full construction and proof are given in Appendix A.5. Briefly, the lexicographic maximum $\mathbf{x}^*$ in X is a vector consisting of 0 in the first $n-1$ components and 1 in the nth component. For all $\varepsilon > 0$, X also includes a vector $\mathbf{x}^{(\varepsilon)}$ whose first component is $-\frac{\varepsilon}{2}$, next $n-2$ components are $\frac{\varepsilon}{4}$, and nth component is $\frac{1}{2}$. We prove that for all $n \geq 8$ and $c \geq 2$ there exists $\varepsilon > 0$ such that $L_c(\mathbf{x}^{(\varepsilon)}) < L_c(\mathbf{x}^*)$, essentially because when n is sufficiently large the lower loss on the middle $n-2$ components compensates for the higher loss on the first and last components. Observing that $\sigma_n(\mathbf{x}^{(\varepsilon)}) = \sigma_n(\mathbf{x}^*) - \frac{1}{2}$ completes the proof. $\square$

From Theorem 5 it immediately follows that the set constructed in Theorem 6 is not lexicographically stable. We can also give more direct intuition for why the set is unstable. The set X in Theorem 6 consists of a "good" and a "bad" line segment, and the unique lexicographic maximum is a point on the "good" line segment. If the optimization problem in the first iteration of Algorithm 1 is solved exactly, then the smallest component of the solution will be equal to 0. In this case, every iteration of the algorithm will output a solution on the "good" line segment, since only points on that segment have a smallest component that is at least 0. However, if the optimization problem in the first iteration is solved with a tolerance $\varepsilon > 0$, then the smallest component of the solution can be as small as $-\varepsilon$. In this case, every iteration of the algorithm will output a solution on the "bad" line segment, since the 2nd, 3rd, ..., $(n-1)$th smallest components of the points on that segment are larger than the corresponding components of the points on the "good" line segment. However, the largest component of each point on the "good" line segment is equal to 1, while the largest component of each point on the "bad" line segment is equal to $\frac{1}{2}$. As a result, the algorithm outputs a vector whose largest component has a value that is far from its lexicographic maximum.

Theorem 6 provides a much stronger example of instability than the construction of Diana et al. (2021), who showed that for all $\varepsilon > 0$ there exists a set X such that $\mathcal{A}(X, \varepsilon)$, the set of possible outputs on input (X, ε), contains an element that is far from the lexicographic maximum $\mathbf{x}^*$ of X. By contrast, Theorem 6 reverses the order of the quantifiers, and shows that there exists a set X such that for all $\varepsilon > 0$ the set $\mathcal{A}(X, \varepsilon)$ contains an element that is far from $\mathbf{x}^*$.

6.2 Convergence rates

Theorems 7 and 8 below give bounds on the rate at which a near or exact minimizer $\mathbf{x}_{c,\gamma} \in X$ of the exponential loss function $L_c(\mathbf{x})$ converges to a lexicographic maximum $\mathbf{x}^* \in \operatorname{lexmax} X$ as $c \to \infty$. Theorem 7 states that the smallest and second smallest components of $\mathbf{x}_{c,\gamma}$ are never more than $O(1/c)$ below their lexicographically maximum values, provided that $\gamma \in [0, 1)$, so that an arbitrarily good approximation is possible by making c large. Note that the theorem makes no assumptions about X, not even that it is lexicographically stable. While the rate of convergence for the smallest component of $\mathbf{x}_{c,\gamma}$ has been studied previously (Rosset et al., 2004), we believe that we are the first to prove unconditional convergence, even asymptotically, for the second smallest component.

Theorem 7. *For all $n \geq 2$, $X \subseteq \mathbb{R}^n, \mathbf{x}^* \in \operatorname{lexmax} X, c > 0$ and $\gamma \in [0, 1)$*

$$d_{\{1,2\}}(\mathbf{x}^* \mid \mathbf{x}_{c,\gamma}) \leq \frac{1}{c} \log\left(\frac{n - k + 1}{1 - \gamma}\right).$$

Proof. By Lemma 1 we only need to show

$$\sup_{\mathbf{x} \in X_{k-1}(\mathbf{x}_{c,\gamma})} \sigma_k(\mathbf{x}) \geq \sigma_k(\mathbf{x}^*)$$

for $k \in \{1, 2\}$. By the definition of $\mathbf{x}^*$ we have $\sup_{\mathbf{x} \in X} \sigma_1(\mathbf{x}) = \sigma_1(\mathbf{x}^*)$, and since $X_0(\mathbf{x}_{c,\gamma}) = X$ this implies $\sup_{\mathbf{x} \in X_0(\mathbf{x}_{c,\gamma})} \sigma_1(\mathbf{x}) = \sigma_1(\mathbf{x}^*)$. We also have

$$\begin{aligned}
\sigma_2(\mathbf{x}^*) &= \sup_{\mathbf{x} \in X : \sigma_1(\mathbf{x}) \geq \sigma_1(\mathbf{x}^*)} \sigma_2(\mathbf{x}) && \because \text{Definition of } \mathbf{x}^* \\
&\leq \sup_{\mathbf{x} \in X : \sigma_1(\mathbf{x}) \geq \sigma_1(\mathbf{x}_{c,\gamma})} \sigma_2(\mathbf{x}) && \because \sigma_1(\mathbf{x}^*) \geq \sigma_1(\mathbf{x}_{c,\gamma}) \\
&= \sup_{\mathbf{x} \in X_1(\mathbf{x}_{c,\gamma})} \sigma_2(\mathbf{x}) && \because \text{Eq. (4)}
\end{aligned}$$

□

In contrast to Theorem 7, the situation is very different for the kth smallest component of $\mathbf{x}_{c,\gamma}$ for all $k \geq 3$. Theorem 8 states that this component can remain far below its lexicographically maximum value for arbitrarily large values of c, even if X is a bounded line segment (and thus is lexicographically stable) and $\gamma = 0$ (i.e., the minimization is exact).

Theorem 8. *For all $n \geq k \geq 3$ and $a \geq 1$ there exists a line segment $X \subseteq \mathbb{R}^n$ satisfying $\|X\|_\infty = 1$ such that for all $\mathbf{x}^* \in \operatorname{lexmax} X$ and $c > 0$*

$$d(\mathbf{x}^* \mid \mathbf{x}_{c,0}) \geq d_{\{k\}}(\mathbf{x}^* \mid \mathbf{x}_{c,0}) \geq \frac{1}{3} \min\left\{1, \frac{a}{c}\right\}.$$

Proof sketch. We consider only the case $n = 3$ here, as the general case $n \geq 3$ proceeds very similarly. The complete proof is provided in Appendix A.6.

If $n = 3$ then we define X to be the line segment joining the following two points:

$$\mathbf{x}^* = (\varepsilon, \varepsilon, 1)^\top \text{ and } \mathbf{x}' = \left(0, \frac{2}{3}, \frac{2}{3}\right)^\top$$

where $\varepsilon > 0$. Clearly $\mathbf{x}^*$ is the lexicographic maximum of X. As discussed earlier, if c is large then the dominant term in $L_c(\mathbf{x})$ corresponds to the smallest component of $\mathbf{x}$. However if c is small then several of the largest terms in $L_c(\mathbf{x})$ can have similar magnitude. We show that if $c \leq a = \Omega(\log \frac{1}{\varepsilon})$ then at least the two largest terms in $L_c(\mathbf{x})$, which correspond to the two smallest components of $\mathbf{x}$, have roughly the same magnitude. In this case the minimizer of $L_c(\mathbf{x})$ will be much closer to $\mathbf{x}'$ than to $\mathbf{x}^*$, because the second smallest component of $\mathbf{x}'$ is roughly $\frac{2}{3}$ larger than the second smallest component of $\mathbf{x}^*$, while the smallest component of $\mathbf{x}^*$ is only ε larger than the smallest component of $\mathbf{x}'$. □

6.3 A related algorithm using multiplicative weights

Next, we discuss a related approach for finding a lexicographic maximum using no-regret strategies to solve an associated zero-sum game, as was considered in great detail by Syed (2010). This is another natural approach for finding lexicographic maxima since, at least when X is a convex polytope, we can view the lexicographic maximization computational task through the lens of solving a zero-sum game, a problem where no-regret algorithms have found a great deal of use.

In the game theory perspective, we are trying to solve the following minimax problem:

$$\min_{\mathbf{p} \in \Delta_m} \max_{\mathbf{q} \in \Delta_n} \mathbf{p}^\top \mathbf{M} \mathbf{q}$$

where $\mathbf{M} \in \mathbb{R}^{m \times n}$ and Δ_m, Δ_n are the probability simplices on m, n items, respectively. An *equilibrium pair* of this minimax problem is a pair of distributions $\hat{\mathbf{p}} \in \Delta_m$ and $\hat{\mathbf{q}} \in \Delta_n$ satisfying

$$\mathbf{p}^\top \mathbf{M} \hat{\mathbf{q}} \geq \hat{\mathbf{p}}^\top \mathbf{M} \hat{\mathbf{q}} \geq \hat{\mathbf{p}}^\top \mathbf{M} \mathbf{q}$$

for all $\mathbf{p} \in \Delta_m$ and $\mathbf{q} \in \Delta_n$. Von Neumann (1928) showed that such a "minimax-optimal" pair, commonly known as a Nash equilibrium in a zero-sum game, always exists for every $\mathbf{M}$. There has been considerable work on how to compute such a pair, including through the use of no-regret online learning algorithms for sequentially updating $\mathbf{p}$ and $\mathbf{q}$. For example, Multiplicative Weights (Algorithm 2) is known to compute an ϵ-approximate equilibrium of the game given by $\mathbf{M}$, with $\epsilon = O\left(\sqrt{\log(m)/T} + \sqrt{\log(n)/T}\right)$ (Freund and Schapire, 1999).

Algorithm 2: Multiplicative weights method for computing a Nash equilibrium

Input: $\mathbf{M} \in \mathbb{R}^{m\times n}$, num. iter. T
Input: $\eta_1, \eta_2, \ldots > 0$ learning parameters
$\mathbf{p}_1 \leftarrow \left(\frac{1}{n}, \ldots, \frac{1}{n}\right)^\top$
for $t = 1, \ldots, T$ **do**
  $\mathbf{q}_t \leftarrow$ any element in $\arg\max_{\mathbf{q}\in\Delta_n} \mathbf{p}_t^\top \mathbf{M}\mathbf{q}$
  $\mathbf{p}_{t+1} \leftarrow \frac{1}{Z_{t+1}} \exp\left(-\eta_t \mathbf{M} \sum_{s=1}^{t} \mathbf{q}_s\right)$
    where $\exp$ is applied coordinate wise,
    and Z_{t+1} is the normalizer.
end
Return: $\bar{\mathbf{p}}_T := \frac{1}{T}\sum_{t=1}^{T} \mathbf{p}_t, \bar{\mathbf{q}}_T := \frac{1}{T}\sum_{t=1}^{T} \mathbf{q}_t$

In this paper, we study how to compute $\operatorname{lexmax} X$ for some $X \subseteq \mathbb{R}^m$. Let us suppose X is a convex polytope, that is, the convex hull of a finite set of points $S \subseteq \mathbb{R}^m$. Let $\mathbf{M}$ be a matrix whose columns are the points in S, so that $X = \{\mathbf{M}\mathbf{q} : \mathbf{q} \in \Delta_n\}$. We can then define a lexmax equilibrium strategy for the column player as any $\mathbf{q}^*$ for which $\mathbf{M}\mathbf{q}^* \in \operatorname{lexmax} X$. A "lexmin" equilibrium strategy for the row player can be defined similarly. Thus, a lexmin and lexmax equilibrium pair $\mathbf{p}^*, \mathbf{q}^*$ is a "special" Nash equilibrium which satisfies the additional constraints of being lexicographically optimal. In this work, we focus only on computing a lexmax solution $\mathbf{q}^*$ for the column player.

The use of no-regret algorithms (such as Multiplicative Weights) has been very helpful for finding Nash equilibria in zero-sum games, among many other applications, and the exponentiation used in the update in Algorithm 2 has an attractive similarity to the minimization of the exponential loss (Eq. (1)) considered primarily in this work. A natural question is whether Algorithm 2 is suitable for finding not just any equilibrium strategy, but a lexmax equlibrium strategy $\mathbf{p}^*$ as defined above. Unfortunately, prior work suggests this is not the case when the learning parameter η_t is fixed to a constant:

Theorem 9 (Informal summary of (Syed, 2010, Theorem 3.7))**.** *There is a family of game matrices $\mathbf{M} \in \mathbb{R}^{3\times 4}$ such that if Algorithm 2 is run with a constant learning parameter $\eta_t = \eta$, the output $\bar{\mathbf{q}}_T$ will not converge to a lexmax equilibrium strategy for the row player as $T \to \infty$.*

On the positive side, Syed (2010) also gives a result for when Algorithm 2 computes a lexocographically optimal solution, but only in a very specific case where the solution has distinct values.

We now aim to rehabilitate Algorithm 2, the work of Syed (2010) notwithstanding. As we will see, the choice of learning parameters η_t is indeed quite important. First, let us define the function $H_c : \Delta_n \to \mathbb{R}$ as

$$H_c(\mathbf{q}) := \frac{1}{c}\log\left(\sum_{i=1}^{m} \exp(-c\mathbf{e}_i^\top M\mathbf{q})\right), \tag{5}$$

where $\mathbf{e}_i$ is the ith basis vector. This function is strongly related to $L_c(\cdot)$ as in Eq. (1), except that we have $\frac{1}{c}\log(\cdot)$ operating on the outside. Notice, however, that the log transformation is monotonically increasing, so any minimizer of the exponential loss also minimizes $H_c(\cdot)$. Second, we observe that H_c is *c-smooth* — that is, it satisfies $\|\nabla H_c(\mathbf{q}) - \nabla H_c(\mathbf{q}')\| \le c\|\mathbf{q} - \mathbf{q}'\|$ for any $\mathbf{q}, \mathbf{q}' \in \Delta_n$.

To give our main result in this section, we emphasize that the following leans on a primal-dual perspective on optimization that uses aforementioned tools on game playing. See Wang et al. (2023) for a complete description.

Theorem 10. *If Algorithm 2 is run with parameter $\eta_t = \frac{c}{t}$ then*

$$H_c(\bar{\mathbf{q}}_T) - \min_{\mathbf{q}\in\Delta_n} H_c(\mathbf{q}) = O\left(\frac{c\log T}{T}\right).$$

Proof. This proof proceeds in three parts. First, we recall the well-known *Frank-Wolfe* procedure for minimizing smooth convex functions on constrained sets, applied here to $H_c(\mathbf{q})$. Second, we show that, with the appropriate choice of update parameters, Frank-Wolfe applied to $H_c(\mathbf{q})$ is identical Algorithm 2. Finally, we appeal to standard convergence guarantees for Frank-Wolfe to obtain the desired convergence rate.

The Frank-Wolfe algorithm, applied to $H_c(\cdot)$, is as follows. Let $\bar{\mathbf{q}}_0 \in \Delta_n$ be an arbitrary initial point, and let $\gamma_1, \gamma_2, \ldots > 0$ be a step-size schedule. On each iteration $t = 1, \ldots, T$, we perform the following:

$$\nabla_t \leftarrow \nabla H_c(\bar{\mathbf{q}}_{t-1}) \tag{6}$$

$$\mathbf{q}_t \leftarrow \arg\min_{\mathbf{q}\in\Delta_n} \langle \mathbf{q}, \nabla_t\rangle \tag{7}$$

$$\bar{\mathbf{q}}_t \leftarrow (1-\gamma_t)\bar{\mathbf{q}}_{t-1} + \gamma_t \mathbf{q}_t. \tag{8}$$

Ultimately, the algorithm returns $\bar{\mathbf{q}}_T$.

We now show that this implementation of Frank-Wolfe is identical to Algorithm 2, as long as we have $\gamma_t = \frac{1}{t}$. To see this, we need to observe that the gradients ∇_t can be written as

$$\nabla_t = \nabla H_c(\bar{\mathbf{q}}_{t-1}) \propto -\exp(-c\mathbf{M}\bar{\mathbf{q}}_{t-1})$$

$$= -\exp\left(-\frac{c}{t-1}\mathbf{M}\sum_{s=1}^{t-1}\mathbf{q}_s\right).$$

In other words, on every round we have that $\mathbf{p}_t = -\nabla_t$. Furthermore, the vectors $\mathbf{q}_t$ are chosen in exactly the same way, as the difference in sign is accounted for by the fact that $\mathbf{q}_t$ is chosen as an $\arg\max$ in Algorithm 2 as opposed to an $\arg\min$ in Frank-Wolfe.

Let us finally recall a result that can be found in Abernethy and Wang (2017), that the Frank-Wolfe algorithm run with parameters $\gamma_t = \frac{1}{t}$, for any function which is α-smooth on its domain, converges at a rate of $O(\alpha \log(T)/T)$. We can now appeal to a well-known result on the convergence of Frank-Wolfe.[1] $\square$

This observation emphasizes that the popular multiplicative update method (Algorithm 2), that appears to fail to solve the desired problem according to the work of Syed (2010), actually succeeds in finding a lexicographic maximum insofar as the exponential loss minimization scheme succeeds. The critical "patch" that fixes the algorithm is the modified learning rate of $\frac{c}{t}$ in place of fixed η.

7 CONCLUSION

We proved a close connection between the two primary methods for computing a lexicographic maximum of a set, and used this connection to show that the method based on exponential loss minimization converges to a correct solution for sets that are lexicographically stable. We believe our results represent the most general convergence criteria for exponential loss minimization that are known. We also undertook the first analysis of the convergence rate of exponential loss minimization, and found that even when convergence is guaranteed, the components of the minimizing vector can converge at vastly different rates. Finally, we showed that the well-known Multiplicative Weights algorithm can find a lexicographic maximum of a lexicographically stable set if the learning rate is suitably chosen.

Acknowledgements

We thank Miroslav Dudík , Saharon Rosset and Matus Telgarsky and for helpful discussions, and the anonymous reviewers for their comments and suggestions.

References

Abernethy, J. D. and Wang, J.-K. (2017). On frank-wolfe and equilibrium computation. *Advances in Neural Information Processing Systems*, 30.

Bei, X., Lu, X., and Suksompong, W. (2022). Truthful cake sharing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4809–4817.

Bertsekas, D. and Gallager, R. (2021). *Data networks*. Athena Scientific.

Diana, E., Gill, W., Globus-Harris, I., Kearns, M., Roth, A., and Sharifi-Malvajerdi, S. (2021). Lexicographically fair learning: Algorithms and generalization. In *2nd Symposium on Foundations of Responsible Computing*, page 1.

Dresher, M. (1961). The mathematics of games of strategy: Theory and applications prentice-hall. *Englewood Cliffs, NJ*.

Freund, Y. and Schapire, R. E. (1999). Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103.

Hartman, E., Hassidim, A., Aumann, Y., and Segal-Halevi, E. (2023). Leximin approximation: From single-objective to multi-objective.

Hayden, H. P. (1981). *Voice flow control in integrated packet networks*. PhD thesis, Massachusetts Institute of Technology.

Henzinger, M., Peale, C., Reingold, O., and Shen, J. H. (2022). Leximax approximations and representative cohort selection. *arXiv preprint arXiv:2205.01157*.

Le Boudec, J.-Y. (2000). Rate adaptation, congestion control and fairness: A tutorial. *Ecole Polytechnique Federale de Lausanne*.

Littlestone, N. and Warmuth, M. K. (1994). The weighted majority algorithm. *Information and computation*, 108(2):212–261.

Luss, H. (1999). On equitable resource allocation problems: A lexicographic minimax approach. *Operations Research*, 47(3):361–378.

Luss, H. and Smith, D. R. (1986). Resource allocation among competing activities: A lexicographic minimax approach. *Operations Research Letters*, 5(5):227–231.

Martínez-Legaz, J.-E. and Singer, I. (1987). Lexicographical separation in rn. *Linear Algebra and Its Applications*, 90:147–163.

Miltersen, P. B. and Sørensen, T. B. (2006). Computing proper equilibria of zero-sum games. In *International Conference on Computers and Games*, pages 200–211. Springer.

Mo, J. and Walrand, J. (1998). Fair end-to-end window-based congestion control. In Lai, W. S. and Cooper, R. B., editors, *Performance and Control of Network Systems II*, volume 3530, pages 55 – 63. International Society for Optics and Photonics, SPIE.

Nacson, M. S., Gunasekar, S., Lee, J., Srebro, N., and Soudry, D. (2019). Lexicographic and depth-sensitive margins in homogeneous and non-homogeneous deep

[1] We emphasize that the more classical version of Frank-Wolfe uses slightly different update parameters, $\gamma_t = \frac{2}{t+2}$, and this indeed can be used to obtain convergence that removes the $\log T$ dependence. But this is outside of the scope of this work.

models. In *International Conference on Machine Learning*, pages 4683–4692. PMLR.

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press.

Rosset, S., Zhu, J., and Hastie, T. (2004). Boosting as a regularized path to a maximum margin classifier. *J. Mach. Learn. Res.*, 5:941–973.

Schmeidler, D. (1969). The nucleolus of a characteristic function game. *SIAM Journal on applied mathematics*, 17(6):1163–1170.

Sherali, H. D. and Soyster, A. L. (1983). Preemptive and nonpreemptive multi-objective programming: Relationship and counterexamples. *Journal of Optimization Theory and Applications*, 39:173–186.

Syed, U. A. (2010). *Reinforcement learning without rewards*. Ph.D. Thesis.

Van Damme, E. (1991). *Stability and perfection of Nash equilibria*, volume 339. Springer.

Von Neumann, J. (1928). Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320.

Wang, J.-K., Abernethy, J., and Levy, K. Y. (2023). No-regret dynamics in the fenchel game: A unified framework for algorithmic convex optimization. *Mathematical Programming*, pages 1–66.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A SUPPLEMENTARY MATERIAL

A.1 Lipschitz continuity of sorting functions

Theorem 11. *For all $k \in [n]$ and $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$ we have $|\sigma_k(\mathbf{x}) - \sigma_k(\mathbf{x}')| \le 3 \|\mathbf{x} - \mathbf{x}'\|_\infty$. Therefore, σ_k is continuous.*

Proof. Let $\varepsilon = \|\mathbf{x} - \mathbf{x}'\|_\infty$ and assume without loss of generality that $\sigma_k(\mathbf{x}) \le \sigma_k(\mathbf{x}')$. It suffices to show that $\sigma_k(\mathbf{x}') \le \sigma_k(\mathbf{x}) + 3\varepsilon$. Choose $i, j \in [n]$ such that $\sigma_k(\mathbf{x}) = x_i$ and $\sigma_k(\mathbf{x}') = x'_j$. If $x_i \ge x_j - 2\varepsilon$ then

$$\sigma_k(\mathbf{x}') = x'_j \le x_j + \varepsilon \le x_i + 3\varepsilon = \sigma_k(\mathbf{x}) + 3\varepsilon.$$

It remains to show that $x_i < x_j - 2\varepsilon$ cannot be true. Suppose it is. Since $\sigma_k(\mathbf{x}) = x_i$, we have

$$k \le |\{\ell \in [n] : x_\ell \le x_i\}| \le |\{\ell \in [n] : x_\ell < x_j - 2\varepsilon\}|.$$

Since $\max_{\ell \in [n]} |x_\ell - x'_\ell| \le \varepsilon$, this implies

$$k \le |\{\ell \in [n] : x'_\ell < x'_j\}|$$

which implies $\sigma_k(\mathbf{x}') < x'_j$, a contradiction. □

A.2 Proof of Theorem 4

In this section, we prove Theorem 4. We first sketch an outline.

As preliminary steps, we begin by introducing a function H_k that, in a sense, summarizes the optimization problem being solved (approximately) on each round of Algorithm 1. Since this function is somewhat difficult to work with, we next introduce another function with much more favorable properties, denoted G_I, where $I \subseteq [n]$. We show that, if $|I| = k - 1$ then $H_k \ge G_I$. Moreover, if X is a convex polytope, then G_I is concave and lower semicontinuous.

With these definitions and preliminaries, we then proceed below with the proof of Theorem 4. By way of contradiction, we suppose that X is a convex polytope but nevertheless is not lexicographically stable. This implies the existence of a sequence $\varepsilon_t > 0$ with $\varepsilon_t \to 0$, and another sequence $\mathbf{x}_t \in \mathcal{A}(X, \varepsilon_t)$ converging to a point $\hat{\mathbf{x}} \in X$ that is different than X's lexicographic maximum $\mathbf{x}^*$. With some re-indexing, we then identify an index $k \in [n]$ such that $H_k(\mathbf{x}_t)$ is asymptotically upper-bounded by $\hat{x}_k$, while, on the other hand $G_I(\hat{\mathbf{x}})$ strictly exceeds $\hat{x}_k$, for some $I \subseteq [n]$ with $|I| = k - 1$. Combining, these facts lead to the contradiction:

$$\hat{x}_k < G_I(\hat{\mathbf{x}}) \le \liminf_{t \to \infty} G_I(\mathbf{x}_t) \le \liminf_{t \to \infty} H_k(\mathbf{x}_t) \le \hat{x}_k.$$

We now provide details. Let $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$. For $k \in [n]$, we define the function $H_k : \mathbb{R}^n \to \overline{\mathbb{R}}$ by

$$H_k(\mathbf{x}) = \sup\{\sigma_k(\mathbf{z}) : \mathbf{z} \in X, \sigma_i(\mathbf{z}) \ge \sigma_i(\mathbf{x}) \text{ for } i = 1, \dots, k-1\}. \tag{9}$$

Then Algorithm 1 operates exactly by choosing, on each round k, $\mathbf{x}^{(k)}$ so that

$$\sigma_k(\mathbf{x}^{(k)}) \ge H_k(\mathbf{x}^{(k-1)}) - \varepsilon \tag{10}$$

(and also so that $\sigma_i(\mathbf{x}^{(k)}) \ge \sigma_i(\mathbf{x}^{(k-1)})$ for $i \in [k-1]$).

In fact, this condition can be simplified: As we show next, a point $\mathbf{x}$ is a possible output of Algorithm 1 if and only if it satisfies Eq. (10) with both $\mathbf{x}^{(k-1)}$ and $\mathbf{x}^{(k)}$ replaced by $\mathbf{x}$. Said differently, $\mathbf{x}$ is valid in the sense of the proof of Theorem 5 if and only if it is a possible output.

Proposition 1. *Let $\mathbf{x} \in X$, and let $\varepsilon \ge 0$. Then $\mathbf{x} \in \mathcal{A}(X, \varepsilon)$ if and only if*

$$\sigma_k(\mathbf{x}) \ge H_k(\mathbf{x}) - \varepsilon \tag{11}$$

for all $k \in [n]$.

Proof. If Eq. (11) holds for all $k \in [n]$, then, in running Algorithm 1, we can choose $\mathbf{x}^{(k)} = \mathbf{x}$ on every iteration, proving that $\mathbf{x} \in \mathcal{A}(X, \varepsilon)$.

Conversely, suppose $\mathbf{x} \in \mathcal{A}(X, \varepsilon)$. Let $\mathbf{x}^{(k)}$, for $k \in [n]$, be the sequence of iterates computed by Algorithm 1 resulting in the final output $\mathbf{x}^{(n)} = \mathbf{x}$. Then, by the manner in which these are computed, $\sigma_i(\mathbf{x}) = \sigma_i(\mathbf{x}^{(n)}) \geq \cdots \geq \sigma_i(\mathbf{x}^{(i)})$ for $i \in [n]$. This implies, for $k \in [n]$, that $H_k(\mathbf{x}) \leq H_k(\mathbf{x}^{(k-1)})$ since if $\sigma_i(\mathbf{z}) \geq \sigma_i(\mathbf{x})$ then $\sigma_i(\mathbf{z}) \geq \sigma_i(\mathbf{x}^{(k-1)})$ for $i < k$. Thus,

$$\sigma_k(\mathbf{x}) \geq \sigma_k(\mathbf{x}^{(k)}) \geq H_k(\mathbf{x}^{(k-1)}) - \varepsilon \geq H_k(\mathbf{x}) - \varepsilon,$$

proving Eq. (11). □

Next, let $I \subsetneq [n]$ be a set of indices (other than $[n]$), and let us define the functions $f_I : \mathbb{R}^n \to \mathbb{R}$ and $G_I : \mathbb{R}^n \to \overline{\mathbb{R}}$ by

$$f_I(\mathbf{x}) = \min_{i \in [n] \setminus I} x_i$$

and

$$G_I(\mathbf{x}) = \sup \{ f_I(\mathbf{z}) : \mathbf{z} \in X, z_i = x_i \text{ for } i \in I \} \tag{12}$$

for $\mathbf{x} \in \mathbb{R}^n$. Thus, $f_I(\mathbf{x})$ is equal to the minimum value of the components not in I, and $G_I(\mathbf{x})$ is the maximum value of $f_I(\mathbf{z})$ over all points $\mathbf{z} \in X$ that agree with $\mathbf{x}$ on all components in I. The function G_I bears some resemblence to H_k, but, as we will see, is easier to work with since, with the set I fixed, we avoid the sorting functions σ_i. The next proposition establishes that connection:

Proposition 2. *Let $k \in [n]$ and let $I \subseteq [n]$ with $|I| = k - 1$. Then $H_k(\mathbf{x}) \geq G_I(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$.*

Proof. Let $\mathbf{x} \in \mathbb{R}^n$, and let $\mathbf{z} \in X$ be such that $z_i = x_i$ for $i \in I$. We show, in cases, that $f_I(\mathbf{z}) \leq H_k(\mathbf{x})$. From G_I's definition (Eq. (12)), this will prove the proposition.

Suppose, in the first case, that $\sigma_i(\mathbf{z}) < \sigma_i(\mathbf{x})$ for some $i \in I$. There must exist a set of indices $J \subseteq [n]$ such that $|J| = i$ and

$$\sigma_i(\mathbf{z}) = \max\{z_j : j \in J\}. \tag{13}$$

We claim $J \not\subseteq I$. Otherwise, if $J \subseteq I$, then we must have

$$\sigma_i(\mathbf{z}) = \max\{x_j : j \in J\} \geq \sigma_i(\mathbf{x}).$$

The equality is by Eq. (13) and since $z_j = x_j$ for $j \in I$. The inequality is because $\sigma_i(\mathbf{x})$ is the i-th smallest component of $\mathbf{x}$ and $|J| = i$. This contradicts that $\sigma_i(\mathbf{z}) < \sigma_i(\mathbf{x})$.

Thus, there must exist $j \in J \setminus I$, implying

$$f_I(\mathbf{z}) \leq z_j \leq \sigma_i(\mathbf{z}) < \sigma_i(\mathbf{x}) \leq \sigma_k(\mathbf{x}) \leq H_k(\mathbf{x}).$$

The first inequality is because $j \notin I$; the second is by Eq. (13) and since $j \in J$; the third is by assumption; the fourth is because $i < k$; and the last is by H_k's definition (Eq. (9)).

In the alternative case, $\sigma_i(\mathbf{z}) \geq \sigma_i(\mathbf{x})$ for all $i \in I$, implying that $\mathbf{z}$ satisfies the conditions appearing in H_k's definition. Then, similar to the preceding arguments, there must exist a set $J \subseteq [n]$ with $|J| = k$ and such that $\sigma_k(\mathbf{z}) = \max\{z_j : j \in J\}$. Since $|I| = k - 1$, this implies that there must exist $j \in J \setminus I$. Thus, $f_I(\mathbf{z}) \leq z_j \leq \sigma_k(\mathbf{z}) \leq H_k(\mathbf{x})$ where the last inequality follows from H_k's definition. □

A function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is *lower semicontinuous relative to a set $S \subseteq \mathbb{R}^n$* if for all $\mathbf{x} \in S$ and for every sequence $\mathbf{x}_t$ in S with $\mathbf{x}_t \to \mathbf{x}$, we have

$$\liminf_{t \to \infty} f(\mathbf{x}_t) \geq f(\mathbf{x}).$$

Similarly, the function is *upper semicontinuous relative to S* if instead $\limsup_{t \to \infty} f(\mathbf{x}_t) \leq f(\mathbf{x})$ whenever $\mathbf{x}_t \to \mathbf{x}$ (with $\mathbf{x}_t \in S$).

We next prove useful properties of G_I when X is a convex polytope:

Lemma 2. *Let $I \subsetneq [n]$, and assume X is convex and nonempty. Then G_I is concave. If, in addition, X is a convex polytope, then G_I is lower semicontinuous relative to X.*

Proof. Let $k = |I|$ and suppose $I = \{i_1, \ldots, i_k\}$. Let $P : \mathbb{R}^n \to \mathbb{R}^k$ be the linear mapping that projects a point $\mathbf{x}$ onto just the coordinates in I (so that $[P(\mathbf{x})]_j = x_{i_j}$ for $\mathbf{x} \in \mathbb{R}^n$ and $j \in [k]$). Let δ_X be the indicator function for X so that $\delta_X(\mathbf{x})$ is 0 if $\mathbf{x} \in X$ and is $+\infty$ otherwise. Let $g : \mathbb{R}^k \to \overline{\mathbb{R}}$ be defined by

$$g(\mathbf{y}) = \inf\{-f_I(\mathbf{z}) + \delta_X(\mathbf{z}) : \mathbf{z} \in \mathbb{R}^n, P(\mathbf{z}) = \mathbf{y}\}$$

for $\mathbf{y} \in \mathbb{R}^k$. Note that $-f_I$ is convex, being the pointwise maximum of linear (and so convex) functions, and δ_X is also convex since X is. Therefore $-f_I + \delta_X$ is convex. It follows that g is convex, being the so-called image of $-f_I + \delta_X$ under P. Note further that $-G_I(\mathbf{x}) = g(P(\mathbf{x}))$ for $\mathbf{x} \in \mathbb{R}^n$, so $-G_I$ is convex as well, proving G_I is concave. (In showing g and $-G_I$ are convex, we applied general facts given in Rockafellar (1970, Theorem 5.7).)

If $\mathbf{x} \in X$, then, from G_I's definition, $-G_I(\mathbf{x}) \le -f_I(\mathbf{x}) < +\infty$; thus, X is included in $-G_I$'s effective domain (set of points where it is not $+\infty$). If X is a convex polytope, then it is also *locally simplicial*, and, being included in $-G_I$'s effective domain, it then follows that $-G_I$ is upper semicontinuous relative to X (Rockafellar, 1970, Theorems 10.2 and 20.5). Therefore, G_I is lower semicontinuous relative to X. $\square$

Proof of Theorem 4. Let $X \subseteq \mathbb{R}^n$ be a convex polytope. Let $\mathbf{x}^*$ be the unique lexicographic maximum (which exists by Theorem 2). Suppose by way of contradiction that X is not lexicographically stable. Then there exists $\delta > 0$ such that for all $\varepsilon > 0$ there exists $\hat{\mathbf{x}} \in \mathcal{A}(X, \varepsilon)$ with $d(\mathbf{x}^* \mid \hat{\mathbf{x}}) \ge \delta$, implying that $\sigma_k(\hat{\mathbf{x}}) \le \sigma_k(\mathbf{x}^*) - \delta$ for some $k \in [n]$. Since there are only finitely many values in $[n]$, this in turn implies that there exists $k_0 \in [n]$, $\delta > 0$, a sequence $\varepsilon_t > 0$ with $\varepsilon_t \to 0$, and a sequence $\mathbf{x}_t \in \mathcal{A}(X, \varepsilon_t)$ such that $\sigma_{k_0}(\mathbf{x}_t) \le \sigma_{k_0}(\mathbf{x}^*) - \delta$ for infinitely many t. By discarding all other elements from the sequence, we assume henceforth that this holds for all t.

Since X is a convex polytope, it is also compact. Therefore, the sequence $\mathbf{x}_t$ must have a subsequence converging to some point $\hat{\mathbf{x}} \in X$. By discarding all other elements, we can assume the entire sequence converges so that $\mathbf{x}_t \to \hat{\mathbf{x}}$. Further, by continuity (Theorem 11), $\sigma_{k_0}(\hat{\mathbf{x}}) \le \sigma_{k_0}(\mathbf{x}^*) - \delta$. In particular, this shows that $\hat{\mathbf{x}} \neq \mathbf{x}^*$.

By possibly permuting the components of points in X, we assume without loss of generality that the components are sorted according to the components of $\hat{\mathbf{x}}$, and, when there are ties, according to the components of $\mathbf{x}^*$. That is, we assume the indices have been permuted in such a way that for all $i, j \in [n]$, if $i \le j$ then $\hat{x}_i \le \hat{x}_j$, and in addition, if $\hat{x}_i = \hat{x}_j$ then $x_i^* \le x_j^*$. In particular, this implies $\hat{x}_1 \le \cdots \le \hat{x}_n$, and $\sigma_i(\hat{\mathbf{x}}) = \hat{x}_i$ for $i \in [n]$.

Let k be the smallest index on which $\hat{\mathbf{x}}$ and $\mathbf{x}^*$ differ (so that $\hat{x}_i = x_i^*$ for $i < k$ and $\hat{x}_k \neq x_k^*$). Let $I = [k-1]$.

Claim 1. $\sigma_i(\mathbf{x}^*) = \sigma_i(\hat{\mathbf{x}})$ for $i \in I$.

Proof of claim. We prove, by induction on $m = 0, 1, \ldots, k-1$, that $\sigma_i(\mathbf{x}^*) = \sigma_i(\hat{\mathbf{x}})$ for $i \le m$. This holds vacuously in the base case that $m = 0$. Suppose $m \in [k-1]$ and that the claim holds for $m-1$. Then $\sigma_i(\mathbf{x}^*) = \sigma_i(\hat{\mathbf{x}})$ for $i \le m-1$. Therefore, $\sigma_m(\mathbf{x}^*) \ge \sigma_m(\hat{\mathbf{x}})$ since $\mathbf{x}^*$ is lexicographically optimal. On the other hand, since $\sigma_m(\mathbf{x}^*)$ is the m-th smallest component of $\mathbf{x}^*$, $\sigma_m(\mathbf{x}^*) \le \max\{x_1^*, \ldots, x_m^*\} = \max\{\hat{x}_1, \ldots, \hat{x}_m\} = \hat{x}_m = \sigma_m(\hat{\mathbf{x}})$. This completes the induction. $\Diamond$

By Claim 1, $\sigma_i(\mathbf{x}^*) = \sigma_i(\hat{\mathbf{x}}) = \hat{x}_i = x_i^*$ for $i \in I$. Therefore, $\sigma_k(\mathbf{x}^*)$ is the smallest of the remaining components of $\mathbf{x}^*$. Thus,

$$x_k^* \ge f_I(\mathbf{x}^*) = \sigma_k(\mathbf{x}^*) \ge \sigma_k(\hat{\mathbf{x}}) = \hat{x}_k \tag{14}$$

where the second inequality follows from Claim 1 since $\mathbf{x}^*$ is lexicographically maximal. Since we assumed $x_k^* \neq \hat{x}_k$, we must have $x_k^* > \hat{x}_k$.

Claim 2. $f_I(\mathbf{x}^*) > \hat{x}_k$.

Proof of claim. The proof is very similar to the proof of Theorem 2. Suppose by way of contradiction that the claim is false. Then, in light of Eq. (14), we must have $\sigma_k(\mathbf{x}^*) = f_I(\mathbf{x}^*) = \hat{x}_k$. Let $\mathbf{y} = (\mathbf{x}^* + \hat{\mathbf{x}})/2$, which is in X since X is convex. We compare the components of $\mathbf{y}$ to $\hat{x}_k$. Let $i \in [n]$.

If $i < k$ then $y_i = \hat{x}_i \le \hat{x}_k$ (since $\hat{x}_i = x_i^*$).

If $i = k$ then $y_k = (x_k^* + \hat{x}_k)/2 > \hat{x}_k$ since $x_k^* > \hat{x}_k$.

Finally, suppose $i > k$, implying $\hat{x}_i \ge \hat{x}_k$. If $\hat{x}_i > \hat{x}_k$ then $y_i = (x_i^* + \hat{x}_i)/2 > \hat{x}_k$ since $x_i^* \ge \hat{x}_k$ (by Eq. (14)). Otherwise, $\hat{x}_i = \hat{x}_k$, implying, by how the components are sorted, that $x_i^* \ge x_k^* > \hat{x}_k$; thus, again, $y_i = (x_i^* + \hat{x}_i)/2 > \hat{x}_k$.

To summarize, $y_i = x_i^* \le \hat{x}_k$ if $i \in I$, and $y_i > \hat{x}_k$ if $i \notin I$. It follows that $\sigma_i(\mathbf{y}) = \sigma_i(\mathbf{x}^*) = x_i^*$ for $i = 1, \ldots, k-1$, and that $\sigma_k(\mathbf{y}) = f_I(\mathbf{y}) > \hat{x}_k = \sigma_k(\mathbf{x}^*)$. However, this contradicts that $\mathbf{x}^*$ is lexicographically maximal. $\Diamond$

Combining, we now have

$$\hat{x}_k < f_I(\mathbf{x}^*) \leq G_I(\hat{\mathbf{x}}) \leq \liminf_{t\to\infty} G_I(\mathbf{x}_t) \leq \liminf_{t\to\infty} H_k(\mathbf{x}_t) \leq \liminf_{t\to\infty} [\sigma_k(\mathbf{x}_t) + \varepsilon_t] = \hat{x}_k.$$

The first inequality is by Claim 2. The second is from G_I's definition (Eq. (12)). The third is because G_I is lower semicontinuous relative to X (Lemma 2). The fourth is by Proposition 2. The fifth is by Proposition 1, since $\mathbf{x}_t \in \mathcal{A}(X, \varepsilon_t)$. The equality is because $\varepsilon_t \to 0$ and $\mathbf{x}_t \to \hat{\mathbf{x}}$, implying $\sigma_k(\mathbf{x}_t) \to \sigma_k(\hat{\mathbf{x}}) = \hat{x}_k$ (using Theorem 11).

Having reached a contradiction, we conclude that X is lexicographically stable. □

A.3 A compact, convex set that is not lexicographically stable

Let X be as in Eq. (3), which is convex and compact. We show that X is not lexicographically stable.

By Theorem 2, X has a unique lexicographic maximum $\mathbf{x}^*$. We first argue that $\mathbf{x}^* = (0, 0, 1)^\top$. Note first that for all $\mathbf{x} \in X$, $x_1 \leq 0 \leq \min\{x_2, x_3\}$, implying $\sigma_1(\mathbf{x}) = x_1$. In particular, since $\sigma_1(\mathbf{x}^*)$ maximizes $\sigma_1(\mathbf{x})$ over $\mathbf{x} \in X$, this means we must have $x_1^* = 0$. By X's definition, this implies $x_2^* = 0$. We can then choose $x_3^* = 1$, since this is that component's largest possible value.

Next, we argue that, for $\varepsilon \in (0, 1)$, $\mathbf{x}_\varepsilon = \left(-\varepsilon^2, \varepsilon/2, 3/4\right)^\top \in \mathcal{A}(X, \varepsilon)$, and more specifically that $\mathbf{x}_\varepsilon$ can be chosen for $\mathbf{x}^{(k)}$ on each round k of Algorithm 1.

On round 1, as noted already, if $\mathbf{x} \in X$ then $\sigma_1(\mathbf{x}) = x_1 \leq 0$. Therefore, $\sigma_1(\mathbf{x}_\varepsilon) = -\varepsilon^2$ is within ε of maximizing $\sigma_1(\mathbf{x})$ over $\mathbf{x} \in X$, so we can choose $\mathbf{x}^{(1)} = \mathbf{x}_\varepsilon$.

For round 2, suppose $\mathbf{x} \in X_1(\mathbf{x}_\varepsilon)$, where $X_k(\mathbf{x})$ is as defined in Eq. (4). That is, $x_1 = \sigma_1(\mathbf{x}) \geq -\varepsilon^2$. Then $x_2^2 \leq (1 - x_3)(-x_1) \leq \varepsilon^2$, so $\sigma_2(\mathbf{x}) \leq x_2 \leq \varepsilon$. Therefore, $\sigma_2(\mathbf{x}_\varepsilon) = \varepsilon/2$ is within ε of maximizing $\sigma_2(\mathbf{x})$ over $\mathbf{x} \in X_1(\varepsilon)$, so we can choose $\mathbf{x}^{(2)} = \mathbf{x}_\varepsilon$.

Finally, for round 3, suppose $\mathbf{x} \in X_2(\mathbf{x}_\varepsilon)$, implying $x_1 = \sigma_1(\mathbf{x}) \geq -\varepsilon^2$ and $x_2 \geq \sigma_2(\mathbf{x}) \geq \varepsilon/2$. These imply

$$\varepsilon^2(1 - x_3) \geq (-x_1)(1 - x_3) \geq x_2^2 \geq (\varepsilon/2)^2,$$

and consequently that $x_3 \leq 3/4$. Thus, $\sigma_3(\mathbf{x}_\varepsilon) = 3/4$ maximizes $\sigma_3(\mathbf{x})$ over $\mathbf{x} \in X_2(\mathbf{x}_\varepsilon)$.

We conclude that $\mathbf{x}_\varepsilon \in \mathcal{A}(X, \varepsilon)$. Since $\sigma_3(\mathbf{x}_\varepsilon) = 3/4$ for all $\varepsilon \in (0, 1)$ but $\sigma_3(\mathbf{x}^*) = 1$ (implying $d(\mathbf{x}^* \mid \mathbf{x}_\varepsilon) \geq 1/4$), X is not lexicographically stable.

A.4 Proof of Lemma 1

Proof. Fix $k \in [n]$ and choose any $\mathbf{x} \in X_{k-1}(\mathbf{x}_{c,\gamma})$. By the definition of $\mathbf{x}_{c,\gamma}$, the definition of L_c, and rearranging terms

$$\begin{aligned} &- \gamma \exp(-c \left\| X \right\|_\infty) \leq L_c(\mathbf{x}) - L_c(\mathbf{x}_{c,\gamma}) \\ =& \sum_{i=1}^{n} \exp(-c\sigma_i(\mathbf{x})) - \exp(-c\sigma_i(\mathbf{x}_{c,\gamma})). \end{aligned} \tag{15}$$

Since $\mathbf{x} \in X_{k-1}(\mathbf{x}_{c,\gamma})$ and the function $x \mapsto \exp(-x)$ is decreasing

$$\begin{aligned} &\sum_{i=1}^{n} \exp(-c\sigma_i(\mathbf{x})) - \exp(-c\sigma_i(\mathbf{x}_{c,\gamma})) \\ \leq& \sum_{i=k}^{n} \exp(-c\sigma_i(\mathbf{x})) - \exp(-c\sigma_i(\mathbf{x}_{c,\gamma})). \end{aligned} \tag{16}$$

Since the function $x \mapsto \exp(-x)$ is decreasing and positive

$$\begin{aligned} &\sum_{i=k}^{n} \exp(-c\sigma_i(\mathbf{x})) - \exp(-c\sigma_i(\mathbf{x}_{c,\gamma})) \\ \leq& (n - k + 1) \exp(-c\sigma_k(\mathbf{x})) - \exp(-c\sigma_k(\mathbf{x}_{c,\gamma})). \end{aligned} \tag{17}$$

Combining (15), (16), and (17) and dividing through by $\exp(-c\sigma_k(\mathbf{x}_{c,\gamma}))$ yields

$$\frac{-\gamma \exp(-c\,\|X\|_\infty)}{\exp(-c\sigma_k(\mathbf{x}_{c,\gamma}))} \leq (n-k+1)\exp(-c(\sigma_k(\mathbf{x}) - \sigma_k(\mathbf{x}_{c,\gamma}))) - 1.$$

Since $\sigma_k(\mathbf{x}_{c,\gamma}) \leq \|X\|_\infty$ we have $\exp(-c\,\|X\|_\infty) \leq \exp(-c\sigma_k(\mathbf{x}_{c,\gamma}))$ and therefore

$$-\gamma \leq (n-k+1)\exp(-c(\sigma_k(\mathbf{x}) - \sigma_k(\mathbf{x}_{c,\gamma}))) - 1.$$

By rearranging we have

$$\sigma_k(\mathbf{x}_{c,\gamma}) \geq \sigma_k(\mathbf{x}) - \frac{1}{c}\log\left(\frac{n-k+1}{1-\gamma}\right).$$

Because $\mathbf{x} \in X_{k-1}(\mathbf{x}_{c,\gamma})$ was chosen arbitrarily this implies

$$\sigma_k(\mathbf{x}_{c,\gamma}) \geq \sup_{\mathbf{x}\in X_{k-1}(\mathbf{x}_{c,\gamma})} \sigma_k(\mathbf{x}) - \frac{1}{c}\log\left(\frac{n-k+1}{1-\gamma}\right).$$

□

A.5 Proof of Theorem 6

Define $\mathbf{x}^*, \mathbf{x}', \mathbf{x}'' \in \mathbb{R}^n$ as

$$\begin{aligned}\mathbf{x}^* &= (0,\dots,0,1)^\top,\\ \mathbf{x}' &= \left(0,\dots,0,\frac{1}{2}\right)^\top,\\ \mathbf{x}'' &= \left(-\frac{1}{2},\frac{1}{4},\dots,\frac{1}{4},\frac{1}{2}\right)^\top\end{aligned}$$

and let $X = \text{conv}(\{\mathbf{x}^*, \mathbf{x}'\}) \cup \text{conv}(\{\mathbf{x}', \mathbf{x}''\})$, where conv$(S)$ denotes the convex hull of $S \subseteq \mathbb{R}^n$. Clearly $\|X\|_\infty = 1$. For all $\lambda \in [0,1]$ let

$$\begin{aligned}\mathbf{x}(\lambda) &= \lambda\mathbf{x}^* + (1-\lambda)\mathbf{x}',\\ \mathbf{x}'(\lambda) &= \lambda\mathbf{x}' + (1-\lambda)\mathbf{x}''\end{aligned}$$

Since for all $\lambda \in [0,1)$

$$\begin{aligned}&\sigma_k(\mathbf{x}^*) = \sigma_k(\mathbf{x}(\lambda)) \text{ for } k \in [n-1] \text{ and } \sigma_n(\mathbf{x}^*) > \sigma_n(\mathbf{x}(\lambda)),\\ &\sigma_1(\mathbf{x}^*) > \sigma_1(\mathbf{x}'(\lambda)).\end{aligned}$$

we have $\operatorname{lexmax} X = \{\mathbf{x}^*\}$. We also know that $\mathbf{x}_{c,0}$ must exist, as it is defined to be the minimum of a continuous function on a compact set. Since for all $\lambda \in [0,1]$

$$\begin{aligned}L_c(\mathbf{x}^*) &\leq L_c(\mathbf{x}(\lambda))\\ \sigma_n(\mathbf{x}'(\lambda)) &= \sigma_n(\mathbf{x}^*) - \frac{1}{2}\end{aligned}$$

it remains to show that for all $c \geq 2$ there exists $\lambda_c \in [0,1]$ such that $L_c(\mathbf{x}'(\lambda_c)) < L_c(\mathbf{x}^*)$. Let $\lambda_c = 1 - \frac{2}{c}$. We have

$$\begin{aligned}L_c(\mathbf{x}'(\lambda_c)) - L_c(\mathbf{x}^*) &= \sum_{k=1}^{n} \exp(-c\sigma_k(\mathbf{x}'(\lambda_c))) - \exp(-c\sigma_k(\mathbf{x}^*))\\ &= \exp(1) - 1\\ &\quad + (n-2)\exp\left(-\frac{1}{2}\right) - (n-2)\end{aligned}$$

$$
\begin{aligned}
&+\exp\left(-\frac{c}{2}\right)-\exp(-c)\\
\le\ &\exp(1)-1\\
&+(n-2)\exp\left(-\frac{1}{2}\right)-(n-2)\\
&+\exp\left(-1\right) && \because c\ge 2 \text{ and } \exp(-c)\ge 0\\
<\ &0 && \because n\ge 8
\end{aligned}
$$

A.6 Proof of Theorem 8

Let $\beta=\frac{2}{3}$. Choose $\varepsilon\in(0,\frac{1}{8}]$ such that $a=\frac{1}{\beta}\log\frac{2\beta-1-\varepsilon}{\varepsilon}$. This is feasible because the function $f(\varepsilon)=\frac{1}{\beta}\log\frac{2\beta-1-\varepsilon}{\varepsilon}$ is decreasing and continuous on the interval $(0,\frac{1}{8}]$, $\lim_{\varepsilon\to 0}f(\varepsilon)=\infty$, $f(\frac{1}{8})<1$ and $a\ge 1$. Note that $\varepsilon<2\beta-1<\beta$. Let

$$
\mathbf{x}^*=\Big(\varepsilon,\dots,\varepsilon,\overset{k\text{th term}}{\overset{\downarrow}{1}},\dots,1\Big)^\top\in\mathbb{R}^n,
$$

$$
\mathbf{x}'=\Big(0,\varepsilon,\dots,\varepsilon,\beta,\underset{\underset{k\text{th term}}{\uparrow}}{\beta},\beta,1,\dots,1\Big)^\top\in\mathbb{R}^n
$$

and $X=\operatorname{conv}(\{\mathbf{x}^*,\mathbf{x}'\})$, where $\operatorname{conv}(S)$ denotes the convex hull of $S\subseteq\mathbb{R}^n$. Clearly $\|X\|_\infty=1$. Let $\mathbf{x}(\lambda)=\lambda\mathbf{x}^*+(1-\lambda)\mathbf{x}'$ for $\lambda\in[0,1]$. Observe that for $\lambda\in[0,1)$

$$
\sigma_1(\mathbf{x}^*)=\sigma_1(\mathbf{x}(1))=\varepsilon>\lambda\varepsilon=\sigma_1(\mathbf{x}(\lambda))
$$

and thus $\operatorname{lexmax}X=\{\mathbf{x}^*\}$. We also know that $\mathbf{x}_{c,0}$ must exist and is unique, as it is defined to be the minimum of a strictly convex function on a line segment.

We have

$$
\begin{aligned}
\sigma_1(\mathbf{x}(\lambda))&=\lambda\epsilon,\\
\sigma_{k-1}(\mathbf{x}(\lambda))&=\lambda\epsilon+(1-\lambda)\beta,\\
\sigma_k(\mathbf{x}(\lambda))&=\lambda+(1-\lambda)\beta.
\end{aligned}
$$

Therefore

$$
\begin{aligned}
L_c(\mathbf{x}(\lambda))&=\sum_{i=1}^{n}\exp(-c\sigma_i(\mathbf{x}(\lambda)))\\
&=\exp(-c\lambda\epsilon)+(k-3)\exp(-c\varepsilon)+\exp\left(-c\lambda\epsilon-c(1-\lambda)\beta\right)\\
&\quad+\exp\left(-c\lambda-c(1-\lambda)\beta\right)+(n-k)\exp(-c).
\end{aligned}
$$

Let $\ell(\lambda)=L_c(\mathbf{x}(\lambda))$ for $\lambda\in[0,1]$. We have

$$
\begin{aligned}
\ell'(\lambda)=&-\exp(-c\lambda\epsilon)c\epsilon-\exp\left(-c\lambda\epsilon-c(1-\lambda)\beta\right)(c\epsilon-c\beta)\\
&-\exp\left(-c\lambda-c(1-\lambda)\beta\right)(c-c\beta)
\end{aligned}
$$

and

$$
\begin{aligned}
\ell''(\lambda)=&\exp(-c\lambda\epsilon)(c\epsilon)^2+\exp\left(-c\lambda\epsilon-c(1-\lambda)\beta\right)(c\epsilon-c\beta)^2\\
&+\exp\left(-c\lambda-c(1-\lambda)\beta\right)(c-c\beta)^2.
\end{aligned}
$$

We now divide into two cases, $c\le a$ and $c>a$. First suppose $c\le a$. Since $\ell''(\lambda)\ge 0$ for all $\lambda\in[0,1]$, we know that $\ell'(\lambda)$ is minimized at $\lambda=0$. We have

$$
\begin{aligned}
\ell'(0)\ge 0&\Leftrightarrow -c\varepsilon-\exp\left(-c\beta\right)(c\epsilon-c\beta)-\exp(-c\beta)(c-c\beta)\ge 0\\
&\Leftrightarrow \exp\left(-c\beta\right)(2c\beta-c-c\epsilon)\ge c\epsilon
\end{aligned}
$$

.

$$
\begin{aligned}
&\Leftrightarrow \exp\left(-c\beta\right) \geq \frac{\epsilon}{2\beta - 1 - \epsilon} \\
&\Leftrightarrow c \leq \frac{1}{\beta} \log \frac{2\beta - 1 - \varepsilon}{\varepsilon} \\
&\Leftrightarrow c \leq a.
\end{aligned}
$$

Thus if $c \leq a$ then $\ell'(\lambda) \geq 0$ for all $\lambda \in [0, 1]$, which implies that $\ell(\lambda)$ is minimized at $\lambda = 0$. In other words, $\mathbf{x}_{c,0} = \mathbf{x}'$, and therefore

$$
\sigma_k(\mathbf{x}_{c,0}) - \sigma_k(\mathbf{x}^*) = \sigma_k(\mathbf{x}') - \sigma_k(\mathbf{x}^*) = \beta - 1. \tag{18}
$$

Now suppose $c > a$, which by the above derivation implies that $\ell'(0) < 0$. We also have $\ell'(1) > 0$, since

$$
\begin{aligned}
\ell'(1) > 0 &\Leftrightarrow -\exp(-c\varepsilon)c\varepsilon - \exp(-c\varepsilon)(c\varepsilon - c\beta) - \exp(-c)(c - c\beta) > 0 \\
&\Leftrightarrow -c\varepsilon - (c\varepsilon - c\beta) - \exp(-c(1-\varepsilon))(c - c\beta) > 0 \qquad (19) \\
&\Leftrightarrow \frac{\beta - 2\varepsilon}{1 - \beta} > \exp(-c(1-\varepsilon)) \\
&\Leftrightarrow c > \frac{1}{1-\varepsilon} \log \frac{1-\beta}{\beta - 2\varepsilon}
\end{aligned}
$$

and

$$
\frac{1}{1-\varepsilon} \log \frac{1-\beta}{\beta - 2\varepsilon} \leq \frac{8}{7} \log \frac{\frac{1}{3}}{\frac{2}{3} - \frac{1}{4}} < 0 < a < c \tag{20}
$$

where in Eq. (19) we divided through by $\exp(-c\varepsilon)$ and in Eq. (20) we used $\beta = \frac{2}{3}$ and $\varepsilon \leq \frac{1}{8}$. Therefore $\ell'(\lambda) = 0$ for some $\lambda \in (0, 1)$, which must also be the minimizer of $\ell(\lambda)$. We have

$$
\begin{aligned}
\ell'(\lambda) = 0 \Leftrightarrow &-\exp(-c\lambda\epsilon)c\epsilon - \exp\left(-c\lambda\epsilon - c(1-\lambda)\beta\right)(c\epsilon - c\beta) \\
&- \exp\left(-c\lambda - c(1-\lambda)\beta\right)(c - c\beta) = 0 \\
\Leftrightarrow &-c\epsilon - \exp\left(-c(1-\lambda)\beta\right)(c\epsilon - c\beta) - \exp\left(-c(1-\lambda)\beta\right)(c - c\beta)\, z = 0 \qquad (21) \\
\Leftrightarrow &\exp(-c(1-\lambda)\beta)\big(c\beta - c\varepsilon - (c - c\beta)z\big) = c\varepsilon \\
\Leftrightarrow &\exp(-c(1-\lambda)\beta) = \frac{\varepsilon}{\beta - \varepsilon - (1-\beta)z} \\
\Leftrightarrow &\lambda = 1 - \frac{1}{c\beta} \log \frac{\beta - \epsilon - (1-\beta)z}{\epsilon}. \qquad (22)
\end{aligned}
$$

where in Eq. (21) we divided through by $\exp(-c\lambda\varepsilon)$ and let $z = \exp(-c\lambda(1-\varepsilon))$. Thus $\mathbf{x}_{c,0} = \mathbf{x}(\lambda_c)$, where λ_c is the value of λ that satisfies Eq. (22). We have

$$
\begin{aligned}
\sigma_k(\mathbf{x}_{c,0}) - \sigma_k(\mathbf{x}^*) &= \sigma_k(\mathbf{x}(\lambda_c)) - \sigma_k(\mathbf{x}^*) \\
&= \lambda_c + \beta(1 - \lambda_c) - 1 \\
&= 1 - \frac{1}{c\beta} \log \frac{\beta - \epsilon - (1-\beta)z}{\epsilon} + \frac{1}{c} \log \frac{\beta - \epsilon - (1-\beta)z}{\epsilon} - 1 \\
&= -\left(\frac{1}{\beta} - 1\right) \frac{1}{c} \log \frac{\beta - \epsilon - (1-\beta)z}{\epsilon} \\
&\leq -\left(\frac{1}{\beta} - 1\right) \frac{1}{c} \log \frac{2\beta - 1 - \epsilon}{\epsilon} \qquad (23) \\
&= (\beta - 1)\, \frac{1}{c\beta} \log \frac{2\beta - 1 - \epsilon}{\epsilon} \\
&= (\beta - 1)\, \frac{a}{c} \qquad (24)
\end{aligned}
$$

where in Eq. (23) we used $z \leq 1$. Combining Eq. (18) and (24) we have

$$
\sigma_k(\mathbf{x}_{c,0}) \leq \sigma_k(\mathbf{x}^*) + \max\left\{\beta - 1, (\beta - 1)\frac{a}{c}\right\} = \sigma_k(\mathbf{x}^*) - \frac{1}{3} \min\left\{1, \frac{a}{c}\right\}
$$

where we used $\beta = \frac{2}{3}$.

B Comparison to Hartman et al. (2023)

In this paper we consider $\mathbf{x} \in X$ to be ε-close to a lexicographic maximum of X if $\sigma_i(\mathbf{x}) \geq \sigma_i(\mathbf{x}^*) - \varepsilon$ for all $i \in [n]$ and $\mathbf{x}^* \in \operatorname{lexmax} X$ (see Definition 2). However, Hartman et al. take a different approach. For any $\mathbf{x}, \mathbf{y} \in X$ they say that $\mathbf{y} \succ_\varepsilon \mathbf{x}$ if there exists $k \in [n]$ such that

$$\begin{aligned} \sigma_i(\mathbf{y}) &\geq \sigma_i(\mathbf{x}) \ \text{ for all } i < k \\ \sigma_k(\mathbf{y}) &> \sigma_k(\mathbf{x}) + \varepsilon \end{aligned}$$

and go on to define $\mathbf{x} \in X$ to be ε-close to a lexicographic maximum of X if there is no $\mathbf{y} \in X$ such that $\mathbf{y} \succ_\varepsilon \mathbf{x}$ (see their Section 3.2, and in that section let $\alpha = 1$). These definitions are incompatible. Consider

$$X = \left\{ \mathbf{x}_1 = \begin{pmatrix} 10 \\ 1 \\ 1 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} 10 - \varepsilon \\ 1 - \varepsilon \\ 1 - \varepsilon \end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix} 5 \\ 5 \\ 1 - \varepsilon \end{pmatrix} \right\}$$

Note that $\mathbf{x}_1$ is the only lexicographic maximum of X. So only $\mathbf{x}_1$ and $\mathbf{x}_2$ are ε-close to a lexicographic maximum by our definition, and only $\mathbf{x}_1$ and $\mathbf{x}_3$ are ε-close to a lexicographic maximum by their definition.