Ordering integers under different permutations

Akshay Gupte

Abstract

The question of finding the largest integer contained between two given lists of integers is trivial when integer ordering is interpreted in its usual way. We propose a nontrivial variant wherein each ordering comparison is performed after integers have been mapped under some bijection, and analyze the computational complexity of our combinatorial problem under a family of bijections on $\mathbb{Z}$ introduced in this paper. Deciding feasibility is strongly NP-hard but enumeration finds the optimum in polynomial time when the number of ordering comparisons is bounded by some input constant. Our main contribution is to give a polynomial time factor-preserving reduction to our problem from any cardinality maximizing combinatorial optimization problem whose constraints can be represented in polynomial time by linear lexicographic relations. The clique problem on graphs and maximum boolean satisfiability are two examples of such cardinality problems. Combining our reduction with hardness results for the clique problem, we obtain that the optimum of our problem is hard to approximate within factor $\Omega(n^{1/2-\epsilon})$, for any $\epsilon > 0$, where n is the maximum size of integers in the list input to our problem.

Keywords. Permutations and Bijections, Binary encoding, Lexicography, Hardness of Approximation, Clique in Graphs

AMS 2010 subject classification. 90C27, 68Q17, 06A05

1 A Bijection on Integers

Given L_1 and L_2 as two lists of integers each containing at most k integers, one can easily compute the largest integer x^* such that $\mathtt{max}\{a : a \in L_2\} \leqslant x^* \leqslant \mathtt{min}\{a : a \in L_1\}$. The algorithm is trivial: sort each list to get the smallest and largest integers in L_1 and L_2, respectively, say $\tilde{a}$ and $\hat{a}$, and then if $\hat{a} \leqslant \tilde{a}$, we have $x^* = \tilde{a}$, otherwise the problem is infeasible. The complexity is $\mathcal{O}(k \log k)$ due to the sorting step. This problem is easy because all the ordering comparisons are carried out under the usual total order $\leqslant$, which is interpreting integers in their normal sense. To make the problem nontrivial, the integers must be mapped to some other integer before asking for the list to be ordered. We introduce the notion of creating alternate representations of integers by defining a family of bijections

on integers, and consider the ordering question under these bijective transformations. In order to formally define our combinatorial problem, some notation must be introduced first.

The set of nonnegative integers is $\mathbb{N}$. Every $x \in \mathbb{N}$ has a unique base-2 representation (binary encoding) that is a vector of 0's and 1's. The length of this binary encoding is equal to $\lfloor \log_2 x \rfloor + 1$. For convenience, we denote the size of an integer as

$$\mathsf{size}(x) \coloneqq \lfloor \log_2 x \rfloor.$$

Let us denote the binary encoding of an integer by

$$\mathsf{bits} \colon x \in \mathbb{N} \;\mapsto\; \mathsf{bits}(x) \in \{0,1\}^n, \qquad \text{for some } n \geqslant \mathsf{size}(x) + 1,$$

where we allow for the 0/1 vector to be longer than what it is required to be by padding it with zeros. The binary encoding is a bijection between $\mathbb{N}$ and the set of all 0/1 vectors. To be more precise, if for $n \in \mathbb{N}$ we let $I_n \subset \mathbb{N}$ be the subset

$$I_n \coloneqq \{0, \ldots, 2^n - 1\}, \;\; n \geqslant 1, \qquad I_0 \coloneqq \emptyset,$$

then $\mathsf{bits}\,|_{I_n}$ (the restriction of bits to I_n) is a bijection between I_n and $\{0,1\}^n$. The inverse of bits is the linear functional

$$\mathsf{bits}^{-1} \colon y \in \{0,1\}^n \;\mapsto\; \langle \mathbf{2}, y \rangle \in I_n,$$

where $\mathbf{2} = (1, 2, 4, 8, \ldots)$ and $\langle , \rangle$ is the dot product between vectors. Observe that any $x \in \mathbb{N}$ has $x = \langle \mathbf{2}, \mathsf{bits}(x) \rangle$ and $\mathsf{size}(x) = \mathsf{min}\{n : \; x \in I_n\} - 1$.

The symmetric group of n elements is $\mathfrak{S}_n$. This is the set of all permutations of $\{1, \ldots, n\}$, or equivalently, the set of all bijective functions on $\{1, \ldots, n\}$. The action of $\mathfrak{S}_n$ on $\mathbb{R}^n$ is $\sigma \cdot y = (y_{\sigma(1)}, y_{\sigma(2)}, \ldots, y_{\sigma(n)})$.[1] Taking the composition of this action on 0/1 vectors with bits^{-1} allows us to define for any $\sigma \in \mathfrak{S}_n$ the function

$$\xi_\sigma \colon x \in I_n \;\mapsto\; \mathsf{bits}^{-1}(\sigma \cdot \mathsf{bits}(x)) \in I_n. \tag{1a}$$

By considering arbitrary permutations, ξ_σ generalizes the identity function $\xi_{id} \colon x \mapsto x$, thus allowing us to create alternate integer values of a given integer. Thus, $\xi_\sigma(x)$ is the *integer value of x under the permutation σ*. Each ξ_σ is a bijection since it is the composition of three bijections bits^{-1}, σ and bits. Although bits^{-1} and the action of $\mathfrak{S}_n$ on $\mathbb{R}_n$ (through permutation matrices) are linear transformations, bits is nonlinear (it is piecewise-defined, cf. (5)). For any $S \subseteq I_n$ and $\sigma \in \mathfrak{S}_n$, we have S and $\xi_\sigma(S)$ in bijection but they may not be equal to each other. Clearly, for $S = I_n$ we have $\xi_\sigma(I_n) = I_n$ for all σ. The inverse functional is the bijection

$$\xi_\sigma^{-1} \colon x \in I_n \;\mapsto\; \mathsf{bits}^{-1}(\sigma^{-1} \cdot \mathsf{bits}(x)) \in I_n. \tag{1b}$$

The following example illustrates our bijections.

[1] Note that this is not a group action, but that does not matter to us. For it to be a group action, we must define $\sigma \cdot y = (y_{\sigma^{-1}(1)}, y_{\sigma^{-1}(2)}, \ldots, y_{\sigma^{-1}(n)})$

Example 1. Let $n = 6$, so that $I_6 = \{0, 1, \ldots, 63\}$, and choose $\sigma, \sigma' \in \mathfrak{S}_6$ to be $\sigma = (5\ 3\ 1\ 6\ 2\ 4)$ and $\sigma' = (5\ 4\ 2\ 6\ 1\ 3)$. Take $x = 25 \in I_6$, which has $\mathsf{size}(x) = 4$. The binary encoding of 25 is $(1, 0, 0, 1, 1) \in \{0, 1\}^5$. Extending this to $\{0, 1\}^6$, after padding an extra zero, gives $\mathsf{bits}(25) = (1, 0, 0, 1, 1, 0)$. The values of 25 under the two permutations are

$$\xi_\sigma(25) = \langle \mathbf{2},\ (5\ 3\ 1\ 6\ 2\ 4) \cdot (1, 0, 0, 1, 1, 0) \rangle = \langle \mathbf{2},\ (1, 0, 1, 0, 0, 1) \rangle = 2^{1-1} + 2^{3-1} + 2^{6-1} = 37,$$
$$\xi_{\sigma'}(25) = \langle \mathbf{2},\ (5\ 4\ 2\ 6\ 1\ 3) \cdot (1, 0, 0, 1, 1, 0) \rangle = \langle \mathbf{2},\ (1, 1, 0, 0, 1, 0) \rangle = 2^{1-1} + 2^{2-1} + 2^{5-1} = 19.$$

The inverse values are $\xi_\sigma^{-1}(25) = 50$ and $\xi_{\sigma'}^{-1}(25) = 49$. Similarly, for $x' = 14 \in I_6$, we have $\mathsf{bits}(14) = (0, 1, 1, 1, 0, 0)$, and hence $\xi_\sigma(14) = 50$ and $\xi_{\sigma'}(14) = 38$. Note that $\xi_\sigma(14) > \xi_\sigma(25)$ and $14 < \xi_\sigma^{-1}(25)$, even though $14 < 25 < 50$. $\diamond$

2 The Problem

Our combinatorial problem pertains to finding the largest integer contained between two given lists of integers, where the ordering comparison is performed after mapping the unknown integer under the bijections ξ_σ for a given collection of σ's. That is, we want the largest integer whose values under different permutations satisfy either the $\leqslant$ or $\geqslant$ relationship with a given list of integers. We measure the largest integer not in magnitude but in terms of $\mathsf{size}(\cdot)$. The formal statement is as follows.

Problem (INTORD)**.** Given two finite collections of tuples $S_1, S_2 \subset \mathbb{N} \times \mathfrak{S}_n$, where the length n of the permutations is such that $n \geqslant \max\{\mathsf{size}(a) + 1 :\ (a, \sigma) \in S_1 \cup S_2\}$, find an $x \in I_n$ of largest size satisfying

$$\xi_\sigma(x) \leqslant a,\ \forall (a, \sigma) \in S_1, \qquad \xi_\sigma(x) \geqslant a,\ \forall (a, \sigma) \in S_2. \tag{2}$$

Inequalities (2) are the ordering constraints under different permutations. The assumption that n is larger than the size of every input integer is required for our function $\xi_\sigma \colon I_n \to I_n$. The input to INTORD is an instance $\mathcal{I} = (n, S_1, S_2)$, and its optimal value is denoted by OPT, which is obtained by solving the following combinatorial optimization problem:

$$\mathsf{OPT} \coloneqq \max_x \left\{ \mathsf{size}(x) :\ x \in I_n,\ x \text{ satsifies (2)} \right\}. \tag{3}$$

Since $\mathsf{size}(x) = \min\{n - 1 :\ x \in I_n\}$, it follows that $\mathsf{OPT} \in \{0, 1, \ldots, n-1\}$. An instance is trivial if $\mathsf{OPT} = 0$, which is equivalent to saying that the only values of x satisfying (2) are either $x = 0$ or $x = 1$ ($x = 0$ satisfies (2) if and only if $S_2 = \emptyset$ or $a = 0$ for all $(a, \sigma) \in S_2$).

In general, $\xi_\sigma(x)$ can be smaller or larger than x and neither ξ_σ nor ξ_σ^{-1} are order-preserving, i.e., $x_1 < x_2$ does not imply $\xi_\sigma(x_1) < \xi_\sigma(x_2)$ and $\xi_\sigma(x_1) > x_2$ does not imply $x_1 > \xi_\sigma^{-1}(x_2)$. All of these relationships depend on the permutation under which they are considered. The following example provides a simple instance of our problem that can be solved by enumeration and inspection.

Example 2. Let $n = 6$, so that $I_6 = \{0, 1, \ldots, 63\}$, and choose $\sigma, \sigma' \in \mathfrak{S}_6$ to be $\sigma = (5\ 3\ 1\ 6\ 2\ 4)$ and $\sigma' = (5\ 4\ 2\ 6\ 1\ 3)$. Consider the instance $(n, S_1, S_2) = (6, (50, \sigma), (14, \sigma'))$ of INTORD. We saw in Example 1 that $\xi_\sigma(14) = 50$ and $\xi_{\sigma'}(14) = 38$. This means $x = 14$ satisfies the ordering constraints (2) ($50 = 50$ and $38 > 14$) and so $x = 14$ is a

feasible solution. Similarly, one can verify that $x = 25$ is also feasible. Since $\mathsf{size}(25) = 4 > 3 = \mathsf{size}(14)$, we have $\mathsf{OPT} \geqslant 4$. So now the question is whether there exists some $x \in I_6 \setminus I_5 = \{32, 33, \ldots, 63\}$ satisfying $\xi_\sigma(x) \leqslant 50$ and $\xi_{\sigma'}(x) \geqslant 14$. Such an x will have $\mathtt{bits}(x) = (*, *, *, *, *, 1)$. Since $\sigma(4) = \sigma'(4) = 6$ and the 6^{th} bit in $\mathtt{bits}(x)$ is 1, we have both $\xi_\sigma(x)$ and $\xi_{\sigma'}(x)$ being at least $2^{4-1} = 8$. Hence $x = 32$ is not feasible since it violates the lower bound of 14 ($\xi_{\sigma'}(32) = 8 < 14$). The lower bound of 14 has $\mathtt{bits}(14) = (0, 1, 1, 1, 0, 0)$ and the 2nd, 3rd, 4th bits being 1 and $\sigma'(2) = 4, \sigma'(3) = 2, \sigma'(4) = 6$ implies that any $x \in I_6 \setminus I_5$ with $\mathtt{bits}(x) = (*, 1, *, 1, *, 1)$ satisfies the lower bound of 14. However, such an x violates the upper bound of 50 since $\sigma^{-1}(2) = 5, \sigma^{-1}(4) = 6, \sigma^{-1}(6) = 4$ and so $\xi_\sigma(x) \geqslant 2^{5-1} + 2^{6-1} + 2^{4-1} = 16 + 32 + 8 = 56 > 50$. So, we proceed to solve the problem by enumeration. Considering the second integer $x = 33$ in the list $I_6 \setminus I_5$ (we already saw the first integer 32 is infeasible), we have that $\mathtt{bits}(33) = (1, 0, 0, 0, 0, 1)$ and so $\xi_\sigma(33) = 4 + 8 = 12 < 50$ and $\xi_{\sigma'}(33) = 8 + 16 = 24 > 14$, making $x = 33$ feasible. Hence, $\mathsf{OPT} = 5$ for this instance. $\diamond$

The above example was easy to solve since it had only one $\leqslant$ and one $\geqslant$ constraint. As the number of inequalities increases, enumeration and inspection are not so straightforward and it becomes nontrivial to order integers under different permutations simultaneously and find the largest integer satisfying these orderings.

2.1 Our Results

We are interested in the complexity of computing λ^* exactly and approximately. The first step towards this is to understand the complexity status of the feasibility question. Not all instances are feasible since the $\leqslant$ and $\geqslant$ constraints may contradict each other. Note that even though OPT belongs to the linearly sized interval $\{0, 1, \ldots, n-1\}$, for each λ in this interval, checking by enumeration to see if there exists a $x \in I_n \setminus I_\lambda$ satisfying (2) is not a polynomial time procedure. We establish in Proposition 7 that deciding feasibility of INTORD, and hence computing OPT, is *strongly NP-hard* in general. Proposition 8 observes that OPT can be computed in polynomial time when the number of ordering constraints is bounded by a constant. The enumeration argument used for this is different than the enumeration and inspection carried out in Example 2.

We are interested in the hardness of approximating OPT within a multiplicative factor for feasible instances. We consider two approximation ratios for any $x \in I_n$ satisfying (2):

$$\widetilde{\mu}_{\mathsf{INTORD}}(x) \coloneqq \frac{\mathsf{OPT} - \ell_{\min}}{\mathsf{size}(x) - \ell_{\min}}, \qquad \mu_{\mathsf{INTORD}}(x) \coloneqq \frac{\mathsf{OPT}}{\mathsf{size}(x)}. \tag{4a}$$

The first ratio $\widetilde{\mu}$ is a relative measure of shifted objective values with $\ell_{\min}$ being a lower bound on the minimization version of our problem, i.e.,

$$\ell_{\min} \leqslant \min\{\mathsf{size}(x) : x \in I_n, x \text{ satisfies (2)}\}, \tag{4b}$$

whereas the second ratio μ is the one that is commonly used in literature while studying hardness of approximation. Our main result establishes how inapproximability of $\widetilde{\mu}$ is related to a classical combinatorial problem on graphs.

Theorem 1. *Let* CLIQUE *be the problem of finding maximum cardinality clique in a graph on m vertices. If it is NP-hard to approximate* CLIQUE *within factor $f(m)$ for some monotone function $f(\cdot)$, then for instances of* INTORD *whose feasibility can be checked in polynomial time, it is NP-hard to approximate $\widetilde{\mu}_{\mathsf{INTORD}}$ within factor $f(\sqrt{n+1}-1) = \Omega(f(\sqrt{n}))$.*

There are many inapproximability guarantees (non-existence of polynomial time algorithms) for CLIQUE under different assumptions; see for example [KP06, Table 1]. We note the implication of Theorem 1 under two widely-held assumptions that are known to provide some of the strongest results for CLIQUE.

Corollary 2. *Assuming*[2] *$NP \not\subset ZPP$, for any constant $\epsilon > 0$, there does not exist a randomized polynomial time algorithm that approximates $\widetilde{\mu}_{\mathsf{INTORD}}$ within factor $\Omega(n^{\frac{1}{2}-\epsilon})$ for instances whose feasibility can be checked in polynomial time.*

Proof. Follows from Theorem 1 and Håstad [Hås99, Theorem 5.2] giving us CLIQUE to be inapproximable within $m^{1-\epsilon}$ under the assumption $NP \not\subset ZPP$. □

Corollary 3. *Assuming*[3] *$NP \not\subset BPTIME\left(\exp\left(\log n\right)^{\mathcal{O}(1)}\right)$, for any constant $\epsilon > 0$, there does not exist a probabilistic polynomial time algorithm that approximates $\widetilde{\mu}_{\mathsf{INTORD}}$ within factor $\Omega(\sqrt{n}/2^{(\log n)^{\frac{3}{4}+\epsilon}})$ for instances whose feasibility can be checked in polynomial time.*

Proof. Khot and Ponnuswami [KP06, Theorem 1] gives us CLIQUE to be inapproximable within $m/2^{(\log m)^{\frac{3}{4}+\epsilon}}$, assuming $NP \not\subset BPTIME\left(2^{(\log m)^{\mathcal{O}(1)}}\right)$. It is straightforward to verify that $m/2^{(\log m)^{\frac{3}{4}+\epsilon}}$ is monotone in m. The claim then follows from Theorem 1. □

We remark that our proof for Theorem 1 establishes a reduction technique that works for a general class of cardinality optimization problems of which CLIQUE is a member, another one being the maximum boolean satisfiability problem MAXSAT. However, we state our main result in terms of CLIQUE and not MAXSAT since the former, which is not in *APX*, has worse hardness results than the latter which is in *APX*.

Another consequence of Theorem 1 is that it implies inapproximability of μ_{INTORD}. We argue this at the end. However, this factor is a very weak guarantee since it is $1 + \Omega(\frac{1}{n}) = 1 + o(1)$ (in fact, it is easy to argue that this factor is less than $5/4$ for $n \geqslant 2$).

2.2 Outline of the Paper

We begin by giving some background and preliminaries on our combinatorial problem in §3. We also present two technical lemmas showing how the approximation ratios $\widetilde{\mu}$ and μ are related to each other, which is useful for us to argue how Theorem 1 implies a hardness guarantee on μ_{INTORD}, and may also be of independent interest. The hardness of deciding feasibility of INTORD is shown in §4.1 and a polynomially solvable case is observed in §4.2. Theorem 1 is proved in §5 in three steps that are as follows. First we define an equivalent optimization problem in §5.1. Then, the main part of the proof comes in §5.2

[2]Problems in *NP* cannot be solved by randomized algorithms which always return the correct answer and whose expected running time is polynomial in input size

[3]Problems in *NP* cannot be solved by randomized algorithms whose probability of giving a wrong answer is strictly less than $1/2$ and which have quasi-polynomial running time for every input

where we establish a PTAS reduction from *any* cardinality optimization problem having certain special structure on its constraints to this new problem. This allows using known inapproximability results for a wide family of optimization problems. The third and final step is in in §5.3 where we identify two well-known cardinality optimization problems — max satisfiability and independent sets in graphs, whose constraints have the special structure we need for our reduction. Between these two problems, the independent set problem, which is equivalent to CLIQUE, has worse hardness results, thus leading to the inapproximability claimed in Theorem 1. We end by proving a factor on μ_{INTORD} in Corollary 20.

3 Preliminaries

We denote the vectors $\mathbf{0} = (0, \dots, 0)$, $\mathbf{1} = (1, \dots, 1)$, and

$$\mathbf{2} \;=\; (1, 2, 4, 8, \dots).$$

The lengths of these vectors are conformable to the context in which they are used. A vector $\pi = (\pi_1, \pi_2, \dots)$ is said to be a strictly superincreasing sequence if $\pi_i > \sum_{1 \leqslant k < i} \pi_k$ for all $i \geqslant 2$. The vector $\mathbf{2}$ has $2^{i-1} = 1 + \sum_{1 \leqslant k < i} 2^{k-1}$; this superincreasing property leads to $\mathtt{bits}$ being a bijection between $\mathbb{N}$ and all $0/1$ vectors. The entries of the binary encoding $\mathtt{bits}(x)$ can be found by a greedy backward recursion and are contingent on the value of a partial sum as follows:

$$\mathtt{bits}_i(x) \;=\; \begin{cases} 1, & x \geqslant \displaystyle\sum_{k=i}^{\mathtt{size}(x)} 2^{k-1}\, \mathtt{bits}_k(x) \\[2ex] 0, & \text{otherwise} \end{cases}, \qquad i = \mathtt{size}(x), \dots, 1, \tag{5}$$

and $\mathtt{bits}_i(x) = 0$ for $i = \mathtt{size}(x) + 1, \dots, n$. For our bijection ξ_σ defined in (1a), we have

$$\xi_\sigma(x) \;=\; \sum_{i=1}^{\lfloor \log_2 x \rfloor} 2^{\sigma^{-1}(i)-1}\, \mathtt{bits}_i(x) \;=\; \sum_{\substack{i=1: \\ \mathtt{bits}_i(x)=1}}^{\lfloor \log_2 x \rfloor} 2^{\sigma^{-1}(i)-1}.$$

This follows from the definition $\xi_\sigma := \langle \mathbf{2},\, \sigma \cdot \mathtt{bits}(x) \rangle = \langle \sigma^{-1} \cdot \mathbf{2},\, \mathtt{bits}(x) \rangle$. Since $\mathtt{bits}(0)$ and $\mathtt{bits}(1)$ are vectors with all zeros and all ones, respectively, we have that $\xi_\sigma(x) = 0$ if and only if $x = 0$, and $\xi_\sigma(x) = 2^n - 1$ if and only if $x = 2^n - 1$.

Observe the following alternate formulation for INTORD.

Problem (INTORD)**.** Given finite $S_1, S_2 \subset I_n \times \mathfrak{S}_n$ for some $n \in \mathbb{N}$, find the largest $\lambda \in \mathbb{N}$ for which there exists some $x \in I_n \setminus I_\lambda$ satisfying (2).

Note that $a \in I_n$ for every input $(a, \sigma) \in S_1 \cup S_2$ enforces $n \geqslant \max\{\mathtt{size}(a)+1 : (a, \sigma) \in S_1 \cup S_2\}$, which is an assumption in the original problem statement. Since $x \in I_n \setminus I_\lambda$ means $x \in \{2^\lambda, \dots, 2^n - 1\}$, we have that $\mathtt{size}(x) = \lambda$ if and only if $x \in I_n \setminus I_\lambda$. Therefore maximizing $\mathtt{size}(x)$ is equivalent to maximizing λ under the constraint $x \in I_n \setminus I_\lambda$. Thus, solving the above problem gives us OPT.

Observation 1. *The optimal value of any feasible instance of* INTORD *satisfies*

$$\mathsf{OPT} \;=\; \lambda^* \;\coloneqq\; \max_{\lambda,x} \left\{\lambda :\; \lambda \in \mathbb{N},\, x \in I_n \setminus I_\lambda,\, x \text{ satsifies (2)}\right\}.$$

For the rest of the paper, we will use the above mathematical formulation with (λ, x) as variables.

Our complexity analysis is aided by the lexicographic (lex) order on the integer lattice. For $y, z \in \mathbb{Z}^n$, y lex less than or equal to z is denoted by $y \preccurlyeq z$ and is defined as

$$y \preccurlyeq z \;\equiv\; y = z, \text{ or } \exists\, i\colon y_i \leqslant z_i - 1,\; y_k = z_k, k = i+1, \dots, n.$$

If $y \preccurlyeq z$ and $y \neq z$, then we write $y \prec z$. The relation $y \succcurlyeq z$ is defined analogously, and $y \preccurlyeq z$ is equivalent to $z \succcurlyeq y$. For $y, z \in \{0,1\}^n$, the superincreasing sequence $\mathbf{2} = (1, 2, 4, 8, \dots)$ allows us to write the lex relation $y \preccurlyeq z$ as

$$y \preccurlyeq z \iff \langle \mathbf{2}, y\rangle \leqslant \langle \mathbf{2}, z\rangle. \tag{6}$$

More efficient and stronger linear representations of the lex relation can be found in [Gup16; GP18]. Allowing for permutation of coordinates, we have the lex order under $\sigma \in \mathfrak{S}_n$, written as

$$y \preccurlyeq_\sigma z \;\equiv\; \sigma \cdot y \preccurlyeq \sigma \cdot z.$$

Note that for any $y \in \{0,1\}^n$, we have $y \preccurlyeq (0, 1, \dots, 1)$ if and only if $\sigma \cdot y \preccurlyeq (0, 1, \dots, 1)$ for any $\sigma \in \mathfrak{S}_n$. This is because both lex constraints are equivalent to $\sum_{i=1}^n y_i \leqslant n-1$.

For $J \subseteq \{1, \dots, n\}$, we adopt the shorthand notation $(\{y_j\}_{j\in J}) \preccurlyeq (\{z_j\}_{j\in J})$ to denote

$$(\{y_j\}_{j\in J}) \preccurlyeq (\{z_j\}_{j\in J}) \;\equiv\; (\{y_j\}_{j\notin J}, \{y_j\}_{j\in J}) \preccurlyeq (1, \dots, 1, \{z_j\}_{j\in J}). \tag{7a}$$

That is, $(\{y_j\}_{j\in J}) \preccurlyeq (\{z_j\}_{j\in J})$ corresponds to any permuted lex order for permutations whose action on $\{0,1\}^n$ is to order the elements of J as the most significant ones used for comparison and the elements not in J being inconsequential to the ordering relation. The ordering of elements within J is allowed to be arbitrary. Since the permutation σ in $\preccurlyeq_\sigma$ is clear from the equivalence in (7a), we suppress it for ease of reading. Similarly for $\succcurlyeq_\sigma$,

$$(\{y_j\}_{j\in J}) \succcurlyeq (\{z_j\}_{j\in J}) \;\equiv\; (\{y_j\}_{j\notin J}, \{y_j\}_{j\in J}) \succcurlyeq (0, \dots, 0, \{z_j\}_{j\in J}). \tag{7b}$$

We use the lex ordering for 0/1 vectors that arise from the binary encoding of integers. A well-known fact about integers is that for any $a, b \in \mathbb{N}$, we have

$$a < b \iff \mathtt{bits}(a) \prec \mathtt{bits}(b), \qquad a = b \iff \mathtt{bits}(a) = \mathtt{bits}(b). \tag{8}$$

This leads to an alternative equivalent representation for our problem, which we will state in the next section.

For instance $\mathcal{I}$ of a maximization problem Q, the optimal value is $\mathtt{opt}_\mathsf{Q}(\mathcal{I})$ and the feasibility problem is $\mathsf{FEAS}_\mathsf{Q}(\mathcal{I})$, and we drop $\mathcal{I}$ in the notation for arbitrary instances. Note that for $\mathsf{Q} = \mathsf{INTORD}$, for convenience, we denote $\mathtt{opt}_\mathsf{INTORD}(\mathcal{I})$ simply as OPT (cf. (3)). We only consider those instances of Q with $\mathtt{opt}_\mathsf{Q}(\mathcal{I}) > 0$. As seen in (4) for the problem

INTORD, for any feasible solution χ of instance $\mathcal{I}$ of Q we consider two approximation ratios — $\widetilde{\mu}_{\mathsf{Q}}(\chi)$ and $\mu_{\mathsf{Q}}(\chi)$. In particular, if $\mathtt{obj}_{\mathcal{I}}\,\chi$ is the objective function value for χ, then

$$\widetilde{\mu}_{\mathsf{Q}}(\chi) = \frac{\mathtt{opt}_{\mathsf{Q}}(\mathcal{I}) - \ell^{\mathsf{Q}}_{\min}}{\mathtt{obj}_{\mathcal{I}}\,\chi - \ell^{\mathsf{Q}}_{\min}}, \qquad \mu_{\mathsf{Q}}(\chi) = \frac{\mathtt{opt}_{\mathsf{Q}}(\mathcal{I})}{\mathtt{obj}_{\mathcal{I}}\,\chi}, \tag{9}$$

where the shift $\ell^{\mathsf{Q}}_{\min}$ used in computing $\widetilde{\mu}_{\mathsf{Q}}(\chi)$ is some lower bound on the minimization version of Q. Assume that $\ell^{\mathsf{Q}}_{\min} > 0$ and is given as part of the input to problem Q.

Factor-preserving reductions Standard texts, such as Williamson and Shmoys [WS11], provide the background on hardness reductions between computational problems. For two maximization problems Q_1 and Q_2, we write $\mathsf{FEAS}_{\mathsf{Q}_1} \leqslant_{\mathrm{K}} \mathsf{FEAS}_{\mathsf{Q}_2}$ to say that the feasibility question of the first problem is Karp-reducible to the feasibility question of the second problem. Equivalence in feasibility is written as $\mathsf{FEAS}_{\mathsf{Q}_1} \equiv \mathsf{FEAS}_{\mathsf{Q}_2}$, which means $\mathsf{FEAS}_{\mathsf{Q}_1} \leqslant_{\mathrm{K}} \mathsf{FEAS}_{\mathsf{Q}_2}$ and $\mathsf{FEAS}_{\mathsf{Q}_2} \leqslant_{\mathrm{K}} \mathsf{FEAS}_{\mathsf{Q}_1}$. We write $\mathsf{Q}_1 \propto_{\widetilde{\mu}} \mathsf{Q}_2$ to denote a PTAS reduction from Q_1 to Q_2 that strictly preserves the approximation ratio $\widetilde{\mu}$. To be precise, $\mathsf{Q}_1 \propto_{\widetilde{\mu}} \mathsf{Q}_2$ means that any instance $\mathcal{I}_1$ of Q_1 can be mapped in polynomial time to an instance $\mathcal{I}_2$ of Q_2 such that any feasible solution χ to $\mathcal{I}_2$ can be mapped in polynomial time to a feasible solution χ' of $\mathcal{I}_1$ with $\widetilde{\mu}_{\mathsf{Q}_1}(\chi') \leqslant \widetilde{\mu}_{\mathsf{Q}_2}(\chi)$. Thus, inapproximability of $\widetilde{\mu}_{\mathsf{Q}_1}$ within factor $c > 1$ implies that $\widetilde{\mu}_{\mathsf{Q}_2}$ also cannot be approximated within factor c. Here, the lower bound values $\ell^{\mathsf{Q}_1}_{\min}$ and $\ell^{\mathsf{Q}_2}_{\min}$ to be used for computing $\widetilde{\mu}_{\mathsf{Q}_1}$ and $\widetilde{\mu}_{\mathsf{Q}_2}$ are considered as part of the input in $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively. Similarly, $\mathsf{Q}_1 \propto_{\mu} \mathsf{Q}_2$ denotes a PTAS reduction that preserves the ratio μ.

Relating the approximation ratios The ratios $\widetilde{\mu}_{\mathsf{Q}}$ and μ_{Q} are related as follows.

Lemma 4. *Let χ be a feasible solution to $\mathcal{I}$ with $\mathtt{obj}_{\mathcal{I}}\,\chi > \ell^{\mathsf{Q}}_{\min}$. For any constants $\gamma > 1$ and $\delta > 0$ satisfying $\frac{\mathtt{obj}_{\mathcal{I}}\,\chi}{\ell^{\mathsf{Q}}_{\min}} \geqslant \gamma$ and $\mathtt{opt}_{\mathsf{Q}}(\mathcal{I}) \leqslant \delta$, we have that*

$$\mu_{\mathsf{Q}}(\chi) \;\leqslant\; \widetilde{\mu}_{\mathsf{Q}}(\chi) \;\leqslant\; \frac{1-\omega}{1-\frac{1}{\gamma}}\,\mu_{\mathsf{Q}}(\chi) \;\leqslant\; \frac{\frac{1}{\omega}-1}{\gamma-1},$$

where $\omega := \ell^{\mathsf{Q}}_{\min}/\delta$.

Proof. For convenience, let us denote $c_1 = \mathtt{opt}_{\mathsf{Q}}(\mathcal{I})$, $c_2 = \mathtt{obj}_{\mathcal{I}}\,\chi$ and $c_3 = \ell^{\mathsf{Q}}_{\min}$. We have $0 < c_3 < c_2 \leqslant c_1$. The first inequality in the above claim is trivial: $\mu_{\mathsf{Q}}(\chi) \leqslant \widetilde{\mu}_{\mathsf{Q}}(\chi)$ is equivalent to $c_1(c_2 - c_3) \leqslant c_2(c_1 - c_3)$, which is equivalent to $c_2 \leqslant c_1$. For the second inequality, we wish to find $\mathcal{C} > 1$ such that $\widetilde{\mu}_{\mathsf{Q}}(\chi) \leqslant \mathcal{C}\mu_{\mathsf{Q}}(\chi)$, and towards this end, we argue that it suffices to have $\mathcal{C} = \mathcal{C}(\gamma,\delta) := \frac{\gamma}{\gamma-1}(1 - \frac{\ell^{\mathsf{Q}}_{\min}}{\delta})$. Note the following equivalence:

$$\widetilde{\mu}_{\mathsf{Q}}(\chi) \leqslant \mathcal{C}\mu_{\mathsf{Q}}(\chi) \iff c_2(c_1-c_3) \leqslant \mathcal{C}c_1(c_2-c_3) \iff \mathcal{C} \geqslant \frac{c_2(c_1-c_3)}{c_1(c_2-c_3)} = \frac{1}{1-\frac{c_3}{c_2}}\left(1-\frac{c_3}{c_1}\right).$$

The assumption $\frac{c_2}{c_3} \geqslant \gamma$ gives us $\frac{c_3}{c_2} \leqslant \frac{1}{\gamma}$, which implies $\frac{1}{1-\frac{c_3}{c_2}} \leqslant \frac{\gamma}{\gamma-1}$. The assumption $c_1 \leqslant \delta$ gives us $1-\frac{c_3}{c_1} \leqslant 1-\frac{c_3}{\delta}$. Combining the two upper bounds with the above equivalence yields the claim $\widetilde{\mu}_{\mathsf{Q}}(\chi) \leqslant \mathcal{C}(\gamma,\delta)\,\mu_{\mathsf{Q}}(\chi)$. This inequality implies the last inequality due to $c_1 \leqslant \delta$ and $\frac{c_2}{c_3} \geqslant \gamma$ giving us $\mu_{\mathsf{Q}}(\chi) = \frac{c_1}{c_2} \leqslant \frac{\delta}{\gamma c_3}$. □

By taking $\gamma \downarrow 1$ in the proof of the above lemma, we obtain the contradiction $\mathtt{opt}_{\mathsf{Q}}(\mathcal{I}) \leqslant \ell^{\mathsf{Q}}_{\min}$, implying that there does not exist any constant $c > 1$ such that $\widetilde{\mu}_{\mathsf{Q}}(\chi) \leqslant c\,\mu_{\mathsf{Q}}(\chi)$. A particular consequence of Lemma 4 useful for our purposes is the following.

Lemma 5. *Assume the conditions of Lemma 4. If $\widetilde{\mu}_{\mathsf{Q}}$ is hard to approximate within some factor $\alpha > 1$, then μ_{Q} is hard to approximate within factor $\frac{\alpha}{1+(\alpha-1)\omega}$.*

Proof. Let χ be a feasible solution and denote $\mathtt{obj}_{\mathcal{I}}\,\chi = \delta/\beta$ for some $\beta > 1$, where we recall that δ is assumed to be some upper bound on $\mathtt{opt}_{\mathsf{Q}}(\mathcal{I})$. This implies $\mu_{\mathsf{Q}}(\chi) \leqslant \beta$. Choose $\gamma = \mathtt{obj}_{\mathcal{I}}\,\chi/\ell^{\mathsf{Q}}_{\min}$. Then, $\gamma = \delta/(\beta\ell^{\mathsf{Q}}_{\min}) = 1/(\beta\omega)$. Now, $\widetilde{\mu}_{\mathsf{Q}}(\chi) \leqslant \frac{1-\omega}{1-1/\gamma}\mu_{\mathsf{Q}}(\chi)$ from Lemma 4 implies that $\widetilde{\mu}_{\mathsf{Q}}(\chi) \leqslant \frac{(1-\omega)\beta}{1-\omega\beta}$. Hence, $\mu_{\mathsf{Q}}(\chi)$ is inapproximable within

$$\beta^* \coloneqq \sup\left\{\beta : \frac{(1-\omega)\beta}{1-\omega\beta} \leqslant \alpha,\ \beta > 1\right\}.$$

Simple algebra gives us $\beta^* = \alpha/(1+(\alpha-1)\omega)$. □

These lemmas will be used to prove hardness of approximating μ_{INTORD}, based on the hardness of $\widetilde{\mu}_{\mathsf{INTORD}}$.

4 Optimization

4.1 Deciding Feasibility

The feasibility of an instance $\mathcal{I} = (n, S_1, S_2)$ is not a trivial question. Not all instances are feasible since the $\leqslant$ and $\geqslant$ relationships imposed by S_1 and S_2 may contradict each other. Hence, the first step towards understanding the complexity of computing OPT is the decision problem of checking feasibility, denoted by FEAS. Note that if some $x \in \mathbb{N}$ satisfies (2), then there exists some $\lambda \in \mathbb{N}$ such that (λ, x) is feasible to INTORD (if $x \in \{0,1\}$, pick $\lambda = 0$, otherwise $\lambda = 1$). Thus, the feasibility problem for INTORD is

$$\mathsf{FEAS}\colon \text{ decide whether } \exists\, x \in \mathbb{N} \text{ satisfying (2).}$$

When there are only $\leqslant$ or only $\geqslant$ constraints in (2), then the problem is trivially feasible.

Observation 2. *$(n, S_1, \emptyset)$ and $(n, \emptyset, S_2)$ are feasible instances. Moreover, for the instance $(n, \emptyset, S_2)$, we have $\mathsf{OPT} = n - 1$.*

Proof. When $S_2 = \emptyset$, then $(\lambda, x) = (0, 0)$ is feasible due to $\xi_\sigma(0) = 0$ for all σ and $0 \in I_n \setminus I_0 = I_n$. When $S_1 = \emptyset$, then $(\lambda, x) = (n-1, 2^n - 1)$ is feasible and optimal due to $\xi_\sigma(2^n - 1) = 2^n - 1$ for all σ and $2^n - 1 \in I_n \setminus I_{n-1}$. □

We show next that deciding feasibility of INTORD is equivalent (in polynomial time) to deciding whether there exists a 0/1 vector satisfying many lexicographic constraints.

Lemma 6. *$\mathsf{FEAS} \equiv \mathsf{FEAS}_X$, where FEAS_X is the problem of deciding feasibility of*

$$X \coloneqq X(T_1, T_2) = \left\{y \in \{0,1\}^n : y \preccurlyeq_\sigma \theta \ \ \forall(\theta,\sigma) \in T_1,\ y \succcurlyeq_\sigma \theta \ \ \forall(\theta,\sigma) \in T_2\right\}, \tag{10}$$

for $T_1, T_2 \subset \{0,1\}^n \times \mathfrak{S}_n$.

Proof. Equation (8) tells us that $\xi_\sigma(x) \leqslant a$ is equivalent to $\mathtt{bits}(\xi_\sigma(x)) \preccurlyeq \mathtt{bits}(a)$. Since $\mathtt{bits}(\xi_\sigma(x)) = \mathtt{bits}(\mathtt{bits}^{-1}(\sigma \cdot \mathtt{bits}(x))) = \sigma \cdot \mathtt{bits}(x)$, we have that

$$\xi_\sigma(x) \leqslant a \iff \sigma \cdot \mathtt{bits}(x) \preccurlyeq \mathtt{bits}(a) \iff \mathtt{bits}(x) \preccurlyeq_\sigma \sigma^{-1} \cdot \mathtt{bits}(a). \tag{11a}$$

Analogously, $\xi_\sigma(x) \geqslant a$ is equivalent to $\sigma \cdot \mathtt{bits}(x) \succcurlyeq \mathtt{bits}(a)$. Therefore,

$$\begin{aligned} &x \in I_n \text{ satisfies (2)} \iff \mathtt{bits}(x) \in X(T_1, T_2), \\ &\text{for } T_i = \{(\sigma^{-1} \cdot \mathtt{bits}(a), \sigma) : (a, \sigma) \in S_i\},\ i = 1, 2. \end{aligned} \tag{11b}$$

Since the bijection $x \mapsto \mathtt{bits}(x)$ is computable in polynomial time, we have shown that $\mathsf{FEAS} \leqslant_{\mathrm{K}} \mathsf{FEAS}_X$. The arguments are similar for the other direction $\mathsf{FEAS}_X \leqslant_{\mathrm{K}} \mathsf{FEAS}$. □

This enables us to prove that the feasibility question is *strongly NP-hard* and hence, in general, cannot be answered in polynomial time, assuming $P \neq NP$.

Proposition 7. FEAS *is strongly NP-complete when all the integers in S_1 and S_2 are odd.*

Proof. Membership in *NP* is straightforward. For both problems, a certificate is $(\lambda, x) \in \{0, \ldots, n-1\} \times I_n$, whose encoding size is $\mathcal{O}(n)$. The inequalities (2) can be verified in polynomial time since $\xi_\sigma(x) = \langle \mathbf{2}, \sigma \cdot \mathtt{bits}(x) \rangle$ is a composition of three polynomial time computable functions. For the decision version of INTORD, we must also verify that $\lambda \geqslant \underline{\lambda}$ for some $\underline{\lambda} \in \mathbb{N}$ that is part of the input of an instance, but this is trivial to check.

To show that feasibility, and hence also optimization, is *strongly NP-hard*, we argue that 3-SAT $\leqslant_{\mathrm{K}}$ FEAS, where 3-SAT is the boolean satisfiability problem with exactly three literals in each clause. Due to Lemma 6, it suffices to prove that 3-SAT $\leqslant_{\mathrm{K}} \mathsf{FEAS}_X$. Consider a CNF formula $\varphi(\chi) = \wedge_{i=1}^{p} C_i(\chi)$ in p clauses $C_1(\chi), \ldots, C_p(\chi)$ and q booleans $\chi_1, \ldots, \chi_q$. Construct an instance of $X(T_1, T_2)$ as follows. Set $n = 2q$ with the elements of a binary vector $y \in \{0,1\}^{2q}$ ordered as $y = (\chi_1, \chi_{\neg 1}, \ldots, \chi_q, \chi_{\neg q})$. Using the compressed notation (7) for lex constraints, we set the constraints in T_1 to be

$$(\chi_i, \chi_{\neg i}) \preccurlyeq (0,1), \quad i = 1, \ldots, q, \qquad (\neg z_1^i, \neg z_2^i, \neg z_3^i) \preccurlyeq (0,1,1), \quad i = 1, \ldots, p, \tag{12a}$$

where (z_1^i, z_2^i, z_3^i) are the three literals (positive or negative) in clause C_i (if $z_j^i = \neg\chi_t$, then $\neg z_j^i = \chi_t$), and those in T_2 to be

$$(\chi_i, \chi_{\neg i}) \succcurlyeq (1,0), \quad i = 1, \ldots, q. \tag{12b}$$

Thus we have $|T_1| = p + q$ and $|T_1| = q$. The values of θ and σ for T_1 and T_2 can be found easily from (7) — the first set of lex constraints in T_1 have $\theta = \mathbf{1} - \mathbf{e}_{2i-1}$ and the lex constraints in T_2 have $\theta = \mathbf{1} - \mathbf{e}_{2i}$. Clearly, for any $\chi \in \{0,1\}^q$ we have that $\chi_i + \neg\chi_i = 1$ if and only if $(\chi_i, \chi_{\neg i}) \preccurlyeq (0,1)$ and $(\chi_i, \chi_{\neg i}) \succcurlyeq (1,0)$. A clause C_i is satisfied if and only if at least one of its three literals z_1^i, z_2^i, z_3^i is equal to 1, or equivalently, if one of $\neg z_1^i, \neg z_2^i, \neg z_3^i$ is equal to 0, and it is easy to verify that this is equivalent to $(\neg z_1^i, \neg z_2^i, \neg z_3^i) \preccurlyeq (0,1,1)$. Thus, the CNF formula is satisfiable if and only if $X(T_1, T_2)$ is feasible.

The right hand sides in (12a) and (12b) share the common property that each of them has exactly one 0 in it. Each right hand side is of the form $\sigma \cdot \theta$ for some $\theta \in \{0,1\}^{2q}$ with exactly one 0 in it and some $\sigma \in \mathfrak{S}_{2q}$. The unique 0 implies that $\mathtt{bits}^{-1}(\sigma \cdot \theta) = \langle \mathbf{2}, \sigma \cdot \theta \rangle =$

$\langle \mathbf{2}, \mathbf{1} \rangle - 2^k = 2^{2q} - 1 - 2^k$, where k equals $2q-2$ for the first lex constraint in (12a), equals $2q-3$ for the second lex constraint in (12a), and equals $2q-1$ for the lex constraint in (12b). Thus, each $\mathtt{bits}^{-1}(\sigma \cdot \theta)$ is an odd integer. By setting $a = \mathtt{bits}^{-1}(\sigma \cdot \theta)$ and invoking the reduction $\mathsf{FEAS}_X \leqslant_{\mathrm{K}} \mathsf{FEAS}$ from Lemma 6, we obtain that all the a's in the instance of INTORD are odd. □

4.2 Polynomial case

Proposition 7 obviously implies that it is *strongly NP-hard* to solve INTORD exactly. This holds if the number of orderings is allowed to be arbitrary since that is essential in the reduction from 3-SAT (cf. (12)). However, if the number of orderings is bounded by a constant, then we note that enumeration solves the problem in polynomial time.

Proposition 8. *Enumeration solves* INTORD *in polynomial time if either $m_1 + m_2$ or n is bounded by some given constant.*

Proof. Equation (11b) in Lemma 6 and the fact that $\mathtt{size}(\cdot)$ is a nondecreasing function implies that

$$\mathsf{OPT} = \mathtt{size}(x^*), \quad \text{where} \quad x^* = \mathtt{max}\{x : \ \mathtt{bits}(x) \in X(T_1, T_2)\}.$$

Polynomial computability of $\mathtt{size}(\cdot)$ and the equivalence of integer ordering and lex ordering in (8) implies that computing OPT polynomially reduces to the problem of computing the lex maximum integer vector in $X(T_1, T_2)$. Since $X(T_1, T_2) \subset \{0,1\}^n$, there are at most 2^n feasible vectors and so if n is bounded by a constant, then enumeration trivially solves in $\mathcal{O}(1)$ time. Suppose $m_1 + m_2$ is bounded by a constant. Observe that for any $y, \theta \in \{0,1\}^n$ and $\sigma \in \mathfrak{S}_n$, we have the lex relation $y \preccurlyeq_\sigma \theta$ if and only if $y = \theta$ or $y \in \cup_{j=1}^n \mathcal{B}_j$, where $\mathcal{B}_j = \{z \in \{0,1\}^n : \ z_{\sigma(j)} \leqslant \theta_{\sigma(j)} - 1, \ z_{\sigma(k)} = \theta_{\sigma(k)}, k = j, \ldots, n\}$ represents the j^{th} fixing. Applying this observation repeatedly to $X(T_1, T_2)$ which has $m_1 + m_2$ lex relations we obtain that there are $\mathcal{O}(n^{m_1+m_2})$ fixings. When $m_1 + m_2$ is bounded by a constant, we can enumerate over all the fixings in polynomial (in n) time. □

5 Hardness of Approximation

5.1 An Equivalent Problem

Consider the following maximization problem over the set $X(T_1, T_2)$ defined in (10).

Problem (LOGLEX)**.** Given $n \in \mathbb{N}$, $\emptyset \neq Y \subseteq \{0,1\}^n$ and $T_1, T_2 \subset \{0,1\}^n \times \mathfrak{S}_n$, find a nonzero $y \in X(T_1, T_2) \cap Y$ that maximizes $\lfloor \mathtt{log}_2 \langle \mathbf{2}, y \rangle \rfloor$.

We assume that when $X(T_1, T_2)$ is nonempty, it is also nontrivial in the sense that it contains some nonzero point (note that $X = \{\mathbf{0}\}$ if and only if all the θ's are equal to $\mathbf{0}$). The problem class LOGLEX with $Y = \{0,1\}^n$ is denoted by $\mathsf{LOGLEX}^{0/1}$. Checking feasibility of these problems is checking whether $X(T_1, T_2)$ is nonempty, and so $\mathsf{FEAS}_{\mathsf{LOGLEX}^{0/1}} \equiv \mathsf{FEAS}_X$, and then Lemma 6 gives us that $\mathsf{LOGLEX}^{0/1}$ is equivalent to INTORD in feasibility.

Observation 3. $\mathsf{FEAS}_{\mathsf{LOGLEX}^{0/1}} \equiv \mathsf{FEAS}$.

In the sense of optimization, $\mathsf{LOGLEX}^{0/1}$ reduces to INTORD.

Lemma 9. $\mathsf{LOGLEX}^{0/1} \propto_{\widetilde{\mu}} \mathsf{INTORD}$ *and* $\mathsf{LOGLEX}^{0/1} \propto_{\mu} \mathsf{INTORD}$.

Furthermore, if $\mu_{\mathsf{LOGLEX}^{0/1}}$ (resp. $\widetilde{\mu}_{\mathsf{LOGLEX}^{0/1}}$) is hard to approximate within some factor $\alpha(n)$, then μ_{INTORD} (resp. $\widetilde{\mu}_{\mathsf{INTORD}}$) is also hard to approximate within $\alpha(n)$.

Proof. Consider any instance $\mathcal{I} = (n, T_1, T_2)$ of $\mathsf{LOGLEX}^{0/1}$ and let ℓ_1 be the lower bound used in calculating $\widetilde{\mu}_{\mathsf{LOGLEX}^{0/1}}$. Using the equivalence (11a) in Lemma 6, we can map $\mathcal{I}$ in polynomial time to an instance $\mathcal{I}' = (n, S_1, S_2)$ of INTORD by setting

$$S_i \;=\; \{(\mathsf{bits}^{-1}(\sigma \cdot \theta),\, \sigma) : \; (\theta, \sigma) \in T_i\}, \qquad i = 1, 2.$$

For $\mathcal{I}'$, set $\ell_{\min} = \ell_1$. The two feasible sets are in bijection. In particular, y is feasible to $\mathcal{I}$ if and only if $\mathsf{bits}^{-1}(y)$ is feasible to $\mathcal{I}'$ for some $\lambda \in \mathbb{N}$. Equivalently, x satisfies (2) if and only if $\mathsf{bits}(x) \in X(T_1, T_2)$. Henceforth assume feasible instances. We argue that $\mathsf{obj}_{\mathcal{I}'}(\lambda, x) \leqslant \mathsf{obj}_{\mathcal{I}}\, \mathsf{bits}(x)$ and $\mathsf{opt}_{\mathsf{INTORD}}(\mathcal{I}') = \mathsf{opt}_{\mathsf{LOGLEX}^{0/1}}(\mathcal{I})$. This implies that $\mu_{\mathsf{LOGLEX}^{0/1}}(\mathsf{bits}(x)) \leqslant \mu_{\mathsf{INTORD}}(\lambda, x)$, giving us $\mathsf{LOGLEX}^{0/1} \propto_{\mu} \mathsf{INTORD}$. Since $\ell_{\min} = \ell_1$, it also implies that $\widetilde{\mu}_{\mathsf{LOGLEX}^{0/1}}(\mathsf{bits}(x)) \leqslant \widetilde{\mu}_{\mathsf{INTORD}}(\lambda, x)$, giving us $\mathsf{LOGLEX}^{0/1} \propto_{\widetilde{\mu}} \mathsf{INTORD}$.

For the optimal values, we have

$$\begin{aligned}
\mathsf{opt}_{\mathsf{INTORD}}(\mathcal{I}') \;&=\; \max_{\lambda, x \in \mathbb{N}^2} \Big\{\lambda : \; 2^\lambda \leqslant x \leqslant 2^n - 1,\; x \text{ satisfies (2)}\Big\} \\
&=\; \max_{\lambda, x \in \mathbb{N}^2} \big\{\lambda : \; \lambda \leqslant \lfloor \log_2 \langle \mathbf{2},\, \mathsf{bits}(x)\rangle \rfloor \leqslant n - 1,\; \mathsf{bits}(x) \in X(T_1, T_2)\big\} \\
&=\; \max_{x \in \mathbb{N}} \big\{\lfloor \log_2 \langle \mathbf{2},\, \mathsf{bits}(x)\rangle \rfloor : \; \mathsf{bits}(x) \in X(T_1, T_2)\big\} \\
&=\; \max_{y \in \{0,1\}^n} \big\{\lfloor \log_2 \langle \mathbf{2},\, y\rangle \rfloor : \; y \in X(T_1, T_2)\big\} \\
&=:\; \mathsf{opt}_{\mathsf{LOGLEX}^{0/1}}(\mathcal{I}).
\end{aligned}$$

The first equality comes from Observation 1. The second equality is from equivalence of x satisfying (2) with $\mathsf{bits}(x) \in X(T_1, T_2)$, equivalence of $2^\lambda \leqslant x \leqslant 2^n - 1$ with $\lambda \leqslant \lfloor \log_2 x \rfloor \leqslant n - 1$, and $x = \langle \mathbf{2}, \mathsf{bits}(x)\rangle$. The third equality is obvious since we are maximizing λ and because $X(T_1, T_2) \subset \{0, 1\}^n$ implies $\langle \mathbf{2}, \mathsf{bits}(x)\rangle \leqslant 2^n - 1$, making the upper bound of $n - 1$ redundant. The fourth equality is variable substitution $y = \mathsf{bits}(x)$, and the last equality is just definition of the optimal value. It is clear from above arguments that if we were not optimizing and just considering the objective value for any feasible (λ, x), then we have $\lambda \leqslant \lfloor \log_2 \langle \mathbf{2}, \mathsf{bits}(x)\rangle \rfloor$, and hence $\mathsf{obj}_{\mathcal{I}'}(\lambda, x) \leqslant \mathsf{obj}_{\mathcal{I}}\, \mathsf{bits}(x)$.

Since the above reduction preserves the value of n in the reduction from $\mathsf{LOGLEX}^{0/1}$ to INTORD, it follows that the inapproximability factor carries over exactly the same. □

5.2 Reduction from Cardinality Maximization

The cardinality of a vector $y \in \mathbb{R}^n$, also called its ℓ_0-norm, is

$$\begin{aligned}
\|y\|_0 \;&\coloneqq\; \#\{j : \; y_j \neq 0\} \\
&=\; \sum_{j=1}^{n} y_j, \quad \text{if } y \in \{0, 1\}^n.
\end{aligned}$$

The subvector of y corresponding to a subset J is written as y_J. The problem of optimizing the cardinality measure over 0/1 vectors is formally stated as

Problem (CARD)**.** Given $m \in \mathbb{N}$, $J \subseteq \{1, \ldots, m\}$ and $\emptyset \neq \mathcal{Y} \subset \{0,1\}^m$, find $y \in \mathcal{Y}$ that maximizes $\|y_J\|_0$.

Remark 1. We assume, wlog, that any instance $\mathcal{I}$ of CARD has $\mathsf{opt}_{\mathsf{CARD}}(\mathcal{I}) \geqslant 1$, since if the optimal value is zero, then $y_J = \mathbf{0}$ for every $y \in \mathcal{Y}$ and then we could simply reduce the problem dimension by fixing $y_j = 0$ for $j \in J$.

CARD is a *family* of optimization problems, since $\mathcal{Y}$ is allowed to be arbitrary and so each type of $\mathcal{Y}$ defines one problem in this family. Some common examples are maximum matching in graphs, maximum independent set in graphs, maximum satisfiability of a conjunctive normal formula, etc. Any instance of a problem in CARD involves an appropriate encoding of the set $\mathcal{Y}$, for e.g., if the constraints of $\mathcal{Y}$ correspond to a graph, then the input for $\mathcal{Y}$ would be the vertex and edge sets for a graph, the implicit assumption is that the number of constraints in $\mathcal{Y}$ is polynomial in the graph input. More generally, for complexity purposes, $\mathcal{Y}$ can be input through a polynomial time membership oracle.

The main ingredient of our analysis is to establish that any cardinality maximizing combinatorial optimization problem has a factor preserving reduction to LOGLEX.

Theorem 10. $\mathsf{Q} \propto_{\widetilde{\mu}} \mathsf{LOGLEX}$ *for every* $\mathsf{Q} \in \mathsf{CARD}$.

In the above reduction, we use $\ell^{\mathsf{Q}}_{\min} = 0$, which is a lower bound on $\|y_J\|_0$ for every $y \in \mathcal{Y}$. The proof of this theorem has several steps, the first of which is to map any instance of problem $\mathsf{Q} \in \mathsf{CARD}$ to an instance of LOGLEX. We will present an algorithm for this reduction, then analyze how this mapping maps feasible solutions and optimal values of the two problems, leading us to proving Theorem 10. To state this reduction, we recall the compressed notation for lexicographic constraints from equations (7). Also, let $\mathbf{1}$ be a vector of ones of conformable length.

Algorithm 1: A map ϕ that reduces any problem in CARD to LOGLEX.
Input: Instance $\mathcal{I} = (m, J, \mathcal{Y})$ of any $\mathsf{Q} \in \mathsf{CARD}$.
Output: Instance $\phi(\mathcal{I})$ of LOGLEX.
Denote

$$N \coloneqq |J| \,.$$

The mapped instance is created as follows.

— $n = N^2 + m + N$ variables, denoted as
  - u_{ik} for $i \in J$, $1 \leqslant k \leqslant N$,
  - v_i for $i = 1, \ldots, m$,
  - w_k for $1 \leqslant k \leqslant N$.

— $|T_1| = (N^2 + 3N)/2$ with the following inequalities
  1. $(w_k, \{u_{ij}\}_{i \in J}) \preccurlyeq (0, \mathbf{1})$ for $1 \leqslant j \leqslant k \leqslant N$,
  2. $(v_i, \{u_{ik}\}_{k=1\ldots N}) \preccurlyeq (0, \mathbf{1})$ for $i \in J$.

— $|T_2| = 3N^2 + N + 1$ with the following inequalities

3. $(w_k, u_{ik}) \succcurlyeq (1, 0)$ for $i \in J$, $1 \leqslant k \leqslant N$,
4. $(\{u_{ik}\}_{k=1\ldots N}, v_i) \succcurlyeq (\mathbf{1}, 0)$ for $i \in J$,
5. $(\{u_{i'k}\}_{\substack{i' \in J \\ i' \neq i}}, u_{ik}) \succcurlyeq (\mathbf{1}, 0)$ for $i \in J$, $1 \leqslant k \leqslant N$,
6. $(\{u_{ij}\}_{\substack{j=1\ldots N \\ j \neq k}}, u_{ik}) \succcurlyeq (\mathbf{1}, 0)$ for $i \in J$, $1 \leqslant k \leqslant N$,
7. $w_1 \succcurlyeq 1$.

— $Y = \{(u, v, w) \in \{0, 1\}^{N^2+m+N} : v \in \mathcal{Y}\}$.

— $\ell_{\min}^{\mathsf{LOGLEX}} = N^2 + m$.

It is clear that ϕ is computable in polynomial time (note that $\mathcal{Y}$ is provided as input through some polynomial time membership oracle). We now prove properties of the mapped instance $\phi(\mathcal{I})$ created by Algorithm 1 that will help us to argue a factor preserving reduction. The constraints in T_1 and T_2 will be referred to by their numbering, first to seventh. It will be useful to note the logical implications for these constraints.

Observation 4. *The lex constraints in Algorithm 1 have the following combinatorial statements:*

1. $u_{ij} = 1$ *for all* $i \in J \implies w_k = 0$ *for all* $k \geqslant j$,
2. $u_{ik} = 1$ *for all* $k = 1, \ldots, N \implies v_i = 0$,
3. $w_k = 0 \implies u_{ik} = 1$ *for all* i,
4. $v_i = 0 \implies u_{ik} = 1$ *for all* k,
5. $u_{ik} = 0 \implies u_{i'k} = 1$ *for all* $i' \neq i$,
6. $u_{ik} = 0 \implies u_{ij} = 1$ *for all* $j \neq k$.

A solution (u, v, w) is called trivial if $v_J = \mathbf{0}$, otherwise it is a nontrivial solution. We argue that $\phi(\mathcal{I})$ has only nontrivial solutions, each of which has consecutive ones in w.

Lemma 11. *Every feasible solution of* $\phi(\mathcal{I})$ *is nontrivial and has* $w_j = 1$ *for* $1 \leqslant j \leqslant \|v_J\|_0$ *and* $w_j = 0$ *for* $\|v_J\|_0 + 1 \leqslant j \leqslant N$.

Proof. If $v_J = \mathbf{0}$, applying the fourth constraint for every $i \in J$ leads to $u = \mathbf{1}$. If $u = \mathbf{1}$ then the first constraint gives us $w = \mathbf{0}$. If $w = \mathbf{0}$ then the third constraint implies that $u = \mathbf{1}$, and then the second constraint implies that $v_J = \mathbf{0}$. Thus, we have characterized the trivial solutions as follows: $v_J = \mathbf{0} \iff u = \mathbf{1} \iff w = \mathbf{0}$. Now, the seventh constraint in T_2 cuts off trivial solutions.

A nontrivial (u, v, w) has $v_J \neq \mathbf{0}$, so that $\|v_J\|_0 \geqslant 1$, and then the first part implies that $u_{i,j} = 0$ for some i, j. Let $k = \max\{j : \exists\, i, u_{ij} = 0\}$. The third constraint gives us $w_k = 1$. The first constraint tells us that for every $j = 1, \ldots, k$, there exists some i_j such that $u_{i_j,j} = 0$. The fifth constraint guarantees that each i_j is unique and the sixth constraint guarantees that $i_1, i_2, \ldots, i_k$ are distinct. The third constraint now gives us $w_j = 1$ for $j \leqslant k$. The maximality of k means that for $j > k$, we have $u_{ij} = 1$ for all $i \in J$. Then the

first constraint gives us $w_j = 0$ for $j > k$. Thus we have shown that $w = (\mathbf{1}, \mathbf{0})$ where the ones are of length k. To finish the proof, we claim that $k = \|v_J\|_0$. For $j = 1, \ldots, k$, since $u_{i_j,j} = 0$, the fourth constraint implies that $v_{i_j} = 1$. Since $\{i_1, i_2, \ldots, i_k\} \subset J$ and the i_j's are distinct, we have $\|v_J\|_0 \geqslant k$. Since $u_{ij} = 1$ for $i \in J$ and $j > k$, for $i \in J \setminus \{i_1, i_2, \ldots, i_k\}$, we have $v_i = 0$ from the second constraint. Hence, $\|v_J\|_0 = k$ and we are done. □

Next, we establish a one-one correspondence between feasible solutions of $\mathcal{I}$ and $\phi(\mathcal{I})$. The feasible set of CARD is $\mathcal{Y}$, and that of the instance $\phi(\mathcal{I})$ of LOGLEX we denote by $(X \cap Y)_{\phi(\mathcal{I})}$.

Lemma 12. *There exists an injective map $g\colon \mathcal{Y} \to (X \cap Y)_{\phi(\mathcal{I})}$.*

Proof. For any $y \in \{0,1\}^m$, define $T_y := \{j \in J\colon y_j = 1\}$, so that the cardinality of the subvector y_J is $\|y_J\|_0 = |T_y|$. The elements of T_y can be sorted in increasing order. For $i \in J$, define the rank function $\rho_y\colon J \to \{1, \ldots, |T_y|\} \cup \{+\infty\}$ with respect to y as follows: if $i \in T_y$, then $\rho_y(i) = k$ where k is such that i is the k^{th} smallest element among the sorted elements of T_y, otherwise $\rho_y(i) = +\infty$. For any $y \in \{0,1\}^m$, define (u^y_{ik}, w^y_k) for $i \in J$ and $1 \leqslant k \leqslant N$ as follows :

$$u^y_{ik} = \begin{cases} 0 & 1 \leqslant k \leqslant |T_y|\,,\ \rho_y(i) = k \\ 1 & \text{otherwise} \end{cases} \qquad w^y_k = \begin{cases} 1 & 1 \leqslant k \leqslant |T_y| \\ 0 & |T_y| < k \leqslant N. \end{cases} \tag{13a}$$

Define $g\colon \{0,1\}^n \to \{0,1\}^{N^2+m+N}$ to be the map

$$g\colon y \mapsto (u^y, y, w^y). \tag{13b}$$

Clearly, $g(y) \neq g(y')$ for $y \neq y'$. It remains to verify that $g(\mathcal{Y}) \subset (X \cap Y)_{\phi(\mathcal{I})}$. To argue this we must check that for any $y \in \mathcal{Y}$, (u^y, y, w^y) satisfies the $\preccurlyeq$ and $\succcurlyeq$ constraints in $\phi(\mathcal{I})$. If $y = \mathbf{0}$ then $T_y = \emptyset$ implies $u^{\mathbf{0}} = \mathbf{1}, w^{\mathbf{0}} = \mathbf{0}$, so that the first, second, and fourth constraints are equality and the other three $\succcurlyeq$ inequalities are strict. For $y \neq \mathbf{0}$, w^y satisfies the necessary condition of consecutive ones from Lemma 11 and we have $T_y = \{j_1, \ldots, j_t\}$ for some $1 \leqslant t \leqslant N$. Then, if we consider u^y to be partitioned into N blocks, each of length N, as $u^y = (\{u^y_{i1}\}_{i\in J}, \{u^y_{i2}\}_{i\in J}, \ldots, \{u^y_{iN}\}_{i\in J})$, then each of the first t blocks has exactly one zero in it and the remaining blocks are all ones, so that we have

$$u^y = (\underbrace{(\mathbf{1},0,\mathbf{1})}_{k=1}, \ldots, \underbrace{(\mathbf{1},0,\mathbf{1})}_{k=t}, \underbrace{\mathbf{1}}_{t<k\leqslant N}), \qquad w^y = (\underbrace{\mathbf{1}}_{t}, \underbrace{\mathbf{0}}_{N-t}).$$

Since u^y has at most one zero in each block, the fifth and sixth constraints of $\phi(\mathcal{I})$ are satisfied. Since $u^y_{ik} = 1$ and $v_i = y_i = 0$ for $i \notin T_y$, the fourth constraint is satisfied. Since $u^y_{ik} = 0$ only if $k \leqslant |T_y|$, which is when $w^y_k = 1$, the third constraint is satisfied. The rank function ρ_y assigns a unique integer to each $i \in T_j$ and so for any j, we have $u^y_{ij} = 1$ for all $i \in J$ only if $j > |T_y|$, which is when $w^y_k = 0$, implying that the first constraint is satisfied for all $k \geqslant j$. The second constraint holds because $u^y_{ik} = 1$ for all k only if $i \notin T_y$, which is when $y_i = 0$. □

The sets $\mathcal{Y}$ and $(X \cap Y)_{\phi(\mathcal{I})}$ would be in bijection by the Cantor-Schröder-Bernstein theorem if there existed an injective map from $(X \cap Y)_{\phi(\mathcal{I})} \to \mathcal{Y}$. However, such a map does not exist since it is not hard to see that $(X \cap Y)_{\phi(\mathcal{I})}$ has distinct points that differ only in u but not in v and w. Moreover, the map g is not surjective on $(X \cap Y)_{\phi(\mathcal{I})}$ since points (u, v, w) in which indices where u_{ik} equals 0 does not satisfy the sorting condition in equation (13a) are feasible to $\phi(\mathcal{I})$ but not in the image set $g(\mathcal{Y})$. However, we don't need g or the two sets to be bijective; we only need the inverse of g which exists since g is injective. The left inverse of g with domain $(X \cap Y)_{\phi(\mathcal{I})}$ is

$$g^{-1} \colon (u, v, w) \in (X \cap Y)_{\phi(\mathcal{I})} \;\mapsto\; v \in \mathcal{Y}. \tag{13c}$$

For proving Theorem 10, we must relate the objective values of instance $\mathcal{I}$ of CARD and instance $\phi(\mathcal{I})$ of LOGLEX. For convenience, let us denote these objective functions by

$$\mathsf{obj}_{\mathcal{I}}\, y \coloneqq \|y_J\|_0 = \sum_{j \in J} y_j, \quad \mathsf{obj}_{\phi(\mathcal{I})}(u, v, w) \coloneqq \lfloor \log_2 \langle \mathbf{2},\, (u, v, w) \rangle \rfloor.$$

Remark 2. We assume that the ordering of variables in $\phi(\mathcal{I})$ under the identity permutation is $y = (u, v, w)$. Replacing this with (v, u, w), or anything else that keeps w as the last N variables, does not change our analysis.

Next, we obtain an explicit expression for the objective value of each feasible solution.

Lemma 13. *Any feasible solution* (u, v, w) *has* $\mathsf{obj}_{\phi(\mathcal{I})}(u, v, w) = \|v_J\|_0 - 1 + \ell_{\min}^{\mathsf{LOGLEX}}$.

Proof. Since $|J| = N$, for simplicity, assume that the elements of J are numbered as $\{1, \ldots, N\}$. Take any nontrivial (u, v, w). Its objective value is

$$\langle \mathbf{2},\, (u, v, w) \rangle = \sum_{i \in J} \sum_{j=1}^{N} 2^{N(i-1)+j-1} u_{ij} + \sum_{i=1}^{m} 2^{N^2+i-1} v_i + \sum_{j=1}^{N} 2^{N^2+m+j-1} w_j.$$

Let $k = \|v_J\|_0$. Lemma 11 tells us that the solution is nontrivial and $w = (\mathbf{1}, \mathbf{0})$ where the vector of ones is of length $k \geqslant 1$. So $w_k = 1$, and then nonnegativity of each term in above implies that $\langle \mathbf{2},\, (u, v, w) \rangle \;\geqslant\; 2^{N^2+m+k-1}$. The vector $(\mathbf{0}, 1, \mathbf{0})$ with the solitary 1 at the $k+1^{th}$ position, satisfies $w \prec (\mathbf{0}, 1, \mathbf{0})$, which implies that $(u, v, w) \prec (\mathbf{0}, \mathbf{0}, (\mathbf{0}, 1, \mathbf{0}))$. Then equation (6) for lex relation implies that $\langle \mathbf{2},\, (u, v, w) \rangle \;<\; \langle \mathbf{2},\, (\mathbf{0}, \mathbf{0}, (\mathbf{0}, 1, \mathbf{0})) \rangle \;=\; 2^{N^2+m+k}$. Combining the lower and upper bound on $\langle \mathbf{2},\, (u, v, w) \rangle$ leads to

$$N^2 + m + k - 1 \;\leqslant\; \log_2 \langle \mathbf{2},\, (u, v, w) \rangle \;<\; N^2 + m + k.$$

Rounding down gives us $\mathsf{obj}_{\phi(\mathcal{I})}(u, v, w) = N^2 + m + k - 1 = \ell_{\min}^{\mathsf{LOGLEX}} + k - 1$. □

This characterization of feasible values has two implications. First, we have that $\ell_{\min}^{\mathsf{LOGLEX}}$ is a lower bound on the minimization version of LOGLEX and so can be used for computing the approximation ratio $\widetilde{\mu}_{\mathsf{LOGLEX}}$.

Corollary 14. $\ell_{\min}^{\mathsf{LOGLEX}}$ *is a valid lower bound in the computation of* $\widetilde{\mu}_{\mathsf{LOGLEX}}(u, v, w)$.

Proof. Immediate from Lemma 13 since $\|v_J\|_0 \geqslant 1$ by Lemma 11. □

Second, we have a precise relation between the optimal values.

Corollary 15. $\mathtt{opt}_{\mathsf{Q}}(\mathcal{I}) + \ell_{\min}^{\mathsf{LOGLEX}} = \mathtt{opt}_{\mathsf{LOGLEX}}(\phi(\mathcal{I})) + 1$.

Proof. By assumption, we have $\mathtt{opt}_{\mathsf{Q}}(\mathcal{I}) = \|y_J^*\|_0$ for some $y_J^* \neq \mathbf{0}$. Take the map g from Lemma 12 that yields $g(y^*) = (u^{y^*}, y^*, w^{y^*})$ to be a nontrivial feasible solution to $\phi(\mathcal{I})$. This implies $\mathtt{opt}_{\mathsf{LOGLEX}}(\phi(\mathcal{I})) \geqslant \mathtt{obj}_{\phi(\mathcal{I})}\, g(y^*) = \|y_J^*\|_0 - 1 + \ell_{\min}^{\mathsf{LOGLEX}} \geqslant 0$, where the equality is from Lemma 13. Suppose, for sake of contradiction, the first inequality is strict, so that there exists some (u, v, w) feasible to $\phi(\mathcal{I})$ with $\mathtt{obj}_{\phi(\mathcal{I})}(u, v, w) > \|y_J^*\|_0 - 1 + \ell_{\min}^{\mathsf{LOGLEX}}$. By Lemma 13, (u, v, w) is nontrivial and has $\mathtt{obj}_{\phi(\mathcal{I})}(u, v, w) = \|v_J\|_0 - 1 + \ell_{\min}^{\mathsf{LOGLEX}}$, thereby leading to $\|v_J\|_0 - 1 > \|y_J^*\|_0 - 1$, which implies $\|v_J\|_0 \geqslant \|y_J^*\|_0 + 1$. We have $g^{-1}(u, v, w) = v \in \mathcal{Y}$. Since $\|v_J\|_0 \geqslant \|y_J^*\|_0 + 1$, we have reached a contradiction $\mathtt{obj}_{\mathcal{I}}\, g^{-1}(u, v, w) > \mathtt{opt}_{\mathsf{Q}}(\mathcal{I})$. Hence, $g(y^*)$ is optimal to instance $\phi(\mathcal{I})$ of LOGLEX. Since $\mathtt{opt}_{\mathsf{Q}}(\mathcal{I}) = \|y_J^*\|_0$ and $\mathtt{obj}_{\phi(\mathcal{I})}\, g(y^*) = \|y_J^*\|_0 - 1 + \ell_{\min}^{\mathsf{LOGLEX}}$, we have the desired claim. □

We have now established everything needed to prove Theorem 10.

Proof of Theorem 10. Algorithm 1 maps an instance of $\mathsf{Q} \in \mathsf{CARD}$ to an instance of LOGLEX, and this mapping runs in polynomial time. By Lemma 13, a feasible (u, v, w) has $\mathtt{obj}_{\phi(\mathcal{I})}(u, v, w) = \|v_J\|_0 - 1 + \ell_{\min}^{\mathsf{LOGLEX}}$. The inverse map from (13c) has $g^{-1}(u, v, w) = v \in \mathcal{Y}$, and since $\mathtt{obj}_{\mathcal{I}}\, v = \|v_J\|_0$, we have $\mathtt{obj}_{\mathcal{I}}\, g^{-1}(u, v, w) + \ell_{\min}^{\mathsf{LOGLEX}} = \mathtt{obj}_{\phi(\mathcal{I})}(u, v, w) + 1$. Note the fact that any two reals $1 < c_2 \leqslant c_1$ satisfy $c_1/c_2 \leqslant (c_1 - 1)/(c_2 - 1)$. Applying this fact with $c_1 = \mathtt{opt}_{\mathsf{Q}}(\mathcal{I})$ and $c_2 = \mathtt{obj}_{\mathcal{I}}\, g^{-1}(u, v, w)$ and using $\ell_{\min}^{\mathsf{Q}} = 0$ yields

$$\begin{aligned}
\widetilde{\mu}_{\mathsf{Q}}(g^{-1}(u, v, w)) \coloneqq \frac{\mathtt{opt}_{\mathsf{Q}}(\mathcal{I})}{\mathtt{obj}_{\mathcal{I}}\, g^{-1}(u, v, w)} &\leqslant \frac{\mathtt{opt}_{\mathsf{Q}}(\mathcal{I}) - 1}{\mathtt{obj}_{\phi(\mathcal{I})}(u, v, w) - \ell_{\min}^{\mathsf{LOGLEX}}} \\
&= \frac{\mathtt{opt}_{\mathsf{LOGLEX}}(\phi(\mathcal{I})) - \ell_{\min}^{\mathsf{LOGLEX}}}{\mathtt{obj}_{\phi(\mathcal{I})}(u, v, w) - \ell_{\min}^{\mathsf{LOGLEX}}} \\
&\eqqcolon \widetilde{\mu}_{\mathsf{LOGLEX}}(u, v, w),
\end{aligned}$$

where the equality is by Corollary 15. Thus, Algorithm 1 preserves the approximation ratio $\widetilde{\mu}$. Since g^{-1} runs in polynomial time, we conclude that $\mathsf{Q} \propto_{\widetilde{\mu}} \mathsf{LOGLEX}$. □

Note that the inequality in the above proof holds only for the shifted approximation ratio $\widetilde{\mu}$ and same arguments do not work for the ratio μ.

5.3 Proof of Main Result

We prove Theorem 1 here. Let $\mathsf{CARD}^{\preccurlyeq}$ be the following subclass of CARD,

$$\mathsf{CARD}^{\preccurlyeq} \coloneqq \{\mathsf{Q} \in \mathsf{CARD} : \mathsf{FEAS}_{\mathsf{Q}} \in P,\ \mathsf{FEAS}_{\mathsf{Q}} \leqslant_{\mathrm{K}} \mathsf{FEAS}_X\},$$

where we recall that $\mathsf{FEAS}_X(T_1, T_2)$ is the problem of deciding feasibility of the set $X(T_1, T_2)$ from equation (10). From Lemma 6 and Proposition 7, we have that $\mathsf{FEAS}_X \in$ *NP-hard*, and so the reduction $\mathsf{FEAS}_{\mathsf{Q}} \leqslant_{\mathrm{K}} \mathsf{FEAS}_X$ seems to be in the opposite direction than what one could use. However, if we combine this reduction with the assumption $\mathsf{FEAS}_{\mathsf{Q}} \in P$ and employ Algorithm 1, then we obtain that

Lemma 16. *For any feasible instance $\mathcal{I}$ of $\mathsf{Q} \in \mathsf{CARD}^{\preccurlyeq}$,*

1. *Algorithm 1 creates an instance of $\mathsf{LOGLEX}^{0/1}$,*
2. *it is possible to create in polynomial time an instance of INTORD for which a feasible solution can be found in polynomial time.*

Proof. If $\mathsf{FEAS}_{\mathsf{Q}} \in P$ and $\mathcal{I}$ is a feasible instance of Q, then a point $y \in \mathcal{Y}$ can be found in polynomial time. Take the injective map $g\colon \mathcal{Y} \to (X \cap Y)_{\phi(\mathcal{I})}$ of Lemma 12. Computing this map includes sorting of T_y, computing the rank function $\rho_y(\cdot)$ and setting values as per (13a). All of these steps can be carried out in polynomial time. Hence, the instance $\phi(\mathcal{I})$ created by Algorithm 1 has a feasible point $g(y)$ which can be found in polynomial time, meaning that $\mathsf{FEAS}_{\mathsf{LOGLEX}}(\phi(\mathcal{I})) \in P$. If $\mathsf{FEAS}_{\mathsf{Q}} \leqslant_{\mathrm{K}} \mathsf{FEAS}_X$, then it means that there exist some $\tilde{T}_1, \tilde{T}_2 \subset \{0,1\}^m \times \mathfrak{S}_m$ whose cardinality is polynomial in m and N (input size of Q) such that for the set $\mathcal{Y}$ in problem Q we have $\mathcal{Y} = X(\tilde{T}_1, \tilde{T}_2)$. Since $X(T_1, T_2) \cap X(\tilde{T}_1, \tilde{T}_2) = X(T_1 \cup \tilde{T}_1, T_2 \cup \tilde{T}_2)$, the instance $\phi(\mathcal{I})$ of LOGLEX created by Algorithm 1 has $Y = \{0,1\}^{N^2+m+N}$ and feasible set is $X(T_1 \cup \tilde{T}_1, T_2 \cup \tilde{T}_2)$, and so $\phi(\mathcal{I})$ is an instance of $\mathsf{LOGLEX}^{0/1}$. Lemma 6 (and its proof) tells us that $\mathcal{I}' = (n, S_1, S_2)$ is an instance of INTORD for

$$S_i = \{(\mathtt{bits}^{-1}(\sigma \cdot \theta), \sigma) : (\theta, \sigma) \in T_i \cup \tilde{T}_i\}, \ i = 1, 2, \qquad n = N^2 + m + N. \tag{14}$$

This instance has $x = \mathtt{bits}^{-1}(g(y))$ as a feasible solution due to

$$\xi_\sigma(x) \ = \ \xi_\sigma(\mathtt{bits}^{-1}(g(y))) \ = \ \mathtt{bits}^{-1}(\sigma \cdot \mathtt{bits}(\mathtt{bits}^{-1}(g(y)))) \ = \ \mathtt{bits}^{-1}(\sigma \cdot g(y)),$$

$g(y) \in X(T_1 \cup \tilde{T}_1, T_2 \cup \tilde{T}_2)$, and the equivalence between integer ordering and lex ordering from equation (8). Thus, $\mathsf{FEAS}_{\mathsf{INTORD}}(\mathcal{I}') \in P$. □

It follows that problems in $\mathsf{CARD}^{\preccurlyeq}$ reduce to INTORD and this reduction provides inapproximability of INTORD using known hardness results for *NP-hard* problems in $\mathsf{CARD}^{\preccurlyeq}$.

Theorem 17. *Consider any $\mathsf{Q} \in \mathsf{CARD}^{\preccurlyeq}$. We have $\mathsf{Q} \propto_{\widetilde{\mu}} \mathsf{INTORD}$. If Q is also NP-hard and inapproximable*[4] *within factor $f(m)$ for some monotone function $f(\cdot)$, then $\widetilde{\mu}_{\mathsf{LOGLEX}^{0/1}}$ and $\widetilde{\mu}_{\mathsf{INTORD}}$ are inapproximable within $f(\sqrt{n+1}-1)$.*

Proof. Theorem 10 and Lemma 16 give us $\mathsf{Q} \propto_{\widetilde{\mu}} \mathsf{LOGLEX}^{0/1}$, which combined with Lemma 9 and the fact that $\propto$ is transitive, implies that $\mathsf{Q} \propto_{\widetilde{\mu}} \mathsf{INTORD}$. Consider the instance $\mathcal{I}'$ of INTORD created in (14). Since $n = N^2 + m + N$ and $N \leqslant m$, we have $n \leqslant m^2 + 2m$. For any fixed n, the quadratic $m \mapsto m^2 + 2m - n$ has only one positive root $m = \sqrt{n+1} - 1$ and so is nonnegative on $\mathbb{R}_+$ if and only if $m \geqslant \sqrt{n+1} - 1$. Since $f(\cdot)$ is monotone, we have $f(m) \geqslant f(\sqrt{n+1} - 1)$ for $m \geqslant \sqrt{n+1} - 1$. Theorem 17 now implies that if $\mathsf{opt}_{\mathsf{Q}}(\mathcal{I})$ is inapproximable within $c(m)$ then $\mathsf{opt}_{\mathsf{INTORD}}(\mathcal{I}')$ is inapproximable within $f(\sqrt{n+1}-1)$. The arguments for $\widetilde{\mu}_{\mathsf{LOGLEX}^{0/1}}$ are exactly the same. □

To use the above theorem, we need to know whether there exist *NP-hard* problems in $\mathsf{CARD}^{\preccurlyeq}$. Let us point out two classical problems: MAXSAT, the maximum boolean satisfiability problem, and INDSET, the maximum independent set problem on graphs.

[4] In polynomial time, under some standard hypotheses such as $P \neq NP$, $NP \not\subset ZPP$, etc.

Lemma 18. $\mathsf{MAXSAT}, \mathsf{INDSET} \in \mathsf{CARD}^{\preccurlyeq} \cap$ *NP-hard.*

Proof. Both the problems are trivially feasible and well-known to be *NP-hard*. We show how their feasible sets can be reformulated using lex constraints, thereby giving us $\mathsf{FEAS}_{\mathsf{Q}} \leqslant_{\mathrm{K}} \mathsf{FEAS}_X$ and hence membership in $\mathsf{CARD}^{\preccurlyeq}$.

For a CNF formula $\varphi(\chi) = \wedge_{i=1}^{p} C_i(\chi) = \wedge_{i=1}^{p} \left(C_i^+(\chi) \vee C_i^-(\chi)\right)$ in p clauses and q booleans (C_i^+ has positive literals and C_i^- has negated literals), the optimization formulation of MAXSAT is to maximize $\sum_{i=1}^{p} y_i$ over p constraints $\sum_{j \in C_i^+} x_j + \sum_{j \in C_i^-} (1 - x_j) \geqslant y_i$, for $i = 1, \ldots, p$, in $p + q$ binary variables $(x, y) \in \{0,1\}^q \times \{0,1\}^p$. Consider any clause C_i and its corresponding inequality $\sum_{j \in C_i^+} x_j + \sum_{j \in C_i^-} (1 - x_j) \geqslant y_i$. Denote the literals to be ℓ_j^i for $j \in C_i$. Recalling the reduction from SAT in Proposition 7, we can write this inequality as $\sum_{j \in C_i^+} x_{\ell_j^i} + \sum_{j \in C_i^-} x_{\neg \ell_j^i} \geqslant y_i$. Now it is easy to verify that this inequality is equivalent to the lex constraint $\left(y_i, \{x_{\ell_j^i}\}_{j \in C_i^-}\}, \{x_{\neg \ell_j^i}\}_{j \in C_i^+}\right) \preccurlyeq (0, \mathbf{1})$. The lex constraints for $x_k + x_{\neg k} = 1$ are $(x_k, x_{\neg k}) \preccurlyeq (0,1)$ and $(x_k, x_{\neg k}) \succcurlyeq (1,0)$. Thus, $\mathsf{FEAS}_{\mathsf{MAXSAT}} \leqslant_{\mathrm{K}} \mathsf{FEAS}_X(T_1, T_2)$ for $T_i \subset \{0,1\}^{p+2q} \times \mathfrak{S}_{p+2q}$, $i = 1, 2$, with T_1 having $p + q$ elements and T_2 having q elements.

Finding independent sets in a simple graph $G = (V, E)$ is formulated as $x_i + x_j \leqslant 1$ for $(i, j) \in E$ and $x \in \{0,1\}^{|V|}$. Clearly, $x_i + x_j \leqslant 1$ is equivalent to $(x_i, x_j) \preccurlyeq (0,1)$ in the compressed notation from (7). Thus, $\mathsf{FEAS}_{\mathsf{INDSET}} \leqslant_{\mathrm{K}} \mathsf{FEAS}_X(T_1, \emptyset)$ for $T_1 \subset \{0,1\}^{|V|} \times \mathfrak{S}_{|V|}$ with $|E|$ elements. □

Finally, we are ready to prove our main result.

Proof of Theorem 1. We use INDSET, the independent set problem on simple graphs, which is equivalent to the CLIQUE problem in terms of both optimization and approximation. Since this problem is trivially feasible, Lemma 16 implies we can map any graph (an instance of INDSET) to instances of INTORD whose feasibility can be decided in polynomial time. Lemma 18 tells us that we can apply Theorem 17 with $\mathsf{Q} = \mathsf{INDSET}$ to obtain the claimed inapproximability factor. □

Both MAXSAT and INDSET are *APX-hard*: for former, the ratio μ is hard to approximate within factor $\frac{8}{7} - \epsilon$ for any $\epsilon > 0$ [Hås01, Theorem 6.1] and for latter, μ is hard to approximate within factor $\frac{p^*}{4p^{*3} - 3p^{*4}}$ for $p^* = \frac{3 - \sqrt{5}}{2}$ [DS05, Corollary 2.3]. Since Lemma 4 tells us that μ_{Q} is upper bounded by the shifted approximation ratio $\widetilde{\mu}_{\mathsf{Q}}$, it follows that these inapproximabilities also hold with respect to $\widetilde{\mu}_{\mathsf{MAXSAT}}$ and $\widetilde{\mu}_{\mathsf{INDSET}}$, respectively. Lemma 18 shows both these problems are in $\mathsf{CARD}^{\preccurlyeq}$. Applying Theorem 17 gives us $\widetilde{\mu}_{\mathsf{INTORD}} \leqslant \max\{\frac{8}{7} - \epsilon, \frac{p^*}{4p^{*3} - 3p^{*4}}\}$. Basic arithmetic simplifies $\frac{p^*}{4p^{*3} - 3p^{*4}} = \frac{1}{6\sqrt{5} - 13} \approx 2.4015 > \frac{8}{7}$. Hence, we have the following guarantee which tells us that INTORD is *APX-hard*.

Corollary 19. *It is NP-hard to approximate* $\widetilde{\mu}_{\mathsf{INTORD}}$ *within factor 2.4015.*

But this is obviously weaker than having factors that can be arbitrarily close to some function of n, as in Corollaries 2 and 3 which imply that INTORD is not in *APX*, i.e., is not constant factor approximable.

We end by arguing inapproximability of μ_{INTORD}.

Corollary 20. *For instances of* INTORD *whose feasibility can be checked in polynomial time, it is NP-hard to approximate* μ_{INTORD} *within factor* $\frac{n-1}{n+2-\sqrt{n+1}}$.

Proof. Choose $\mathsf{Q} = \mathsf{INDSET} \in \mathsf{CARD}^{\preccurlyeq}$ and consider Algorithm 1 which reduces any input graph G on m vertices. From Lemma 16 we know that this reduction creates an instance $\phi(G)$ of $\mathsf{LOGLEX}^{0/1}$. Corollary 14 gives us $\ell_{\min}^{\phi(G)}$, which is set to $N^2 + m$ in the reduction, to be a lower bound on objective values of feasible solutions of $\phi(G)$. This bound becomes $\ell_{\min}^{\phi(G)} = m^2 + m$ since for INDSET we have $N = m$. Lemma 13 gives an upper bound on objective values to be $N - 1 + \ell_{\min}^{\phi(G)}$, which equals $m^2 + 2m - 1$ since $N = m$. For INDSET, the ratio μ_{INDSET}, and hence also $\widetilde{\mu}_{\mathsf{INDSET}}$ by Lemma 4, is *NP-hard* to approximate within factor arbitrarily close to m [Hås99, Theorem 5.2]. Theorem 17 implies that $\widetilde{\mu}_{\mathsf{LOGLEX}^{0/1}}$ is *NP-hard* to approximate within factor $m = \sqrt{n+1} - 1$. Hence, we can apply Lemma 5 to $\mathsf{LOGLEX}^{0/1}$ with $\alpha = m$, $\ell_{\min} = m^2 + m$ and $\delta = m^2 + 2m - 1$. This gives us that $\mu_{\mathsf{LOGLEX}^{0/1}}$ is hard to approximate within factor

$$\frac{m}{1 + (m-1)\frac{m^2+m}{m^2+2m-1}} = \frac{m(m^2+2m-1)}{m^2+2m-1+(m-1)(m^2+m)} = \frac{m(m^2+2m-1)}{m^3+m^2+m-1},$$

which is lower bounded by

$$\frac{m(m^2+2m-1)}{m^3+m^2+m} = \frac{m^2+2m-1}{m^2+m+1} = 1 + \frac{m-2}{m^2+m+1} = 1 + o(1).$$

Substituting $m = \sqrt{n+1} - 1$ and simplifying gives us a factor of $\frac{n-1}{n+2-\sqrt{n+1}}$. □

6 Conclusion

We have introduced a bijection on integers that is then used to perform ordering comparisons for sorting integers. The combinatorial problem we consider seeks for the largest size of an integer that satisfies under our bijective mapping $\leqslant$ and $\geqslant$ relations with a given list of integers. We show this problem to be *strongly NP-hard* but polynomially solvable when the number of ordering relations is bounded by some given constant. Our main contribution is to show that when this problem is feasible, it is, in general, *NP-hard* to approximate its optimum within factor arbitrarily close to $\sqrt{n}$, where n is the maximum size among all integers input to the problem. In fact, we show that our problem is at least as hard to approximate as any cardinality maximization combinatorial problem whose constraints can be represented through linear lexicographic relationships. This includes the problem of finding maximum clique in a graph. Permutations play an important role in making our problem hard to solve exactly and to approximate. To the best of our knowledge, we are unaware of other studies on establishing bijections that make it hard to order integers and analyzing how hard this can be.

We mention some open questions that deserve consideration in the future. In terms of hardness results, we do not know whether the factor in Theorem 1 is optimal for our problem. Our proof establishes a general reduction framework from cardinality maximization problems whose constraints satisfy our assumption, and any of them with a worse inapproximability threshold than the clique problem would yield a worse hardness factor than

our theorem. It would be interesting to identify such other cardinality problems or improve upon the factor of $\sqrt{n}$ through other PTAS reductions. On the positive side (polynomial solvability), it would be good to have an approximation algorithm with either a $\mathcal{O}(\sqrt{n})$ guarantee, which would prove that our hardness factor is almost tight, or some other upper bound, even if it is for special cases such as having only $\leqslant$ comparisons to be made. A particular question we pose in this regard is the following: *for instances* $(n, S_1, \emptyset)$ *of* INTORD*, is there a PTAS reduction to some packing problem on graphs?* We mention this because $\preccurlyeq_\sigma$ relations can be represented as linear inequalities of $\leqslant$ form with variable coefficients in $\{0, 1\}$, meaning that for instances with $S_2 = \emptyset$, the constraint set can be reduced to a packing polytope (packing in a hypergraph). But it is not immediate whether such an approach will reduce the objective functions in a manner that will allow using approximation results for hypergraph packings. Finally, it may also be of interest to analyze this problem under other bijections on $\mathbb{Z}$.

References

[DS05] I. Dinur and S. Safra. "On the hardness of approximating minimum vertex cover". In: *Annals of Mathematics* 162.1 (2005), pp. 439–485.

[Gup16] A. Gupte. "Convex hulls of superincreasing knapsacks and lexicographic orderings". In: *Discrete Applied Mathematics* 201 (2016), pp. 150–163.

[GP18] A. Gupte and S. Poznanović. "On Dantzig figures from graded lexicographic orders". In: *Discrete Mathematics* 341.6 (2018), pp. 1534–1554.

[Hås99] J. Håstad. "Clique is hard to approximate withinn 1- ε". In: *Acta Mathematica* 182.1 (1999), pp. 105–142.

[Hås01] J. Håstad. "Some optimal inapproximability results". In: *Journal of the ACM* 48.4 (2001), pp. 798–859.

[KP06] S. Khot and A. K. Ponnuswami. "Better inapproximability results for MaxClique, Chromatic Number and Min-3Lin-Deletion". In: *Automata, Languages, and Programming. ICALP 2006*. Ed. by M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener. Vol. 4051. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2006, pp. 226–237.

[WS11] D. P. Williamson and D. B. Shmoys. *The design of approximation algorithms*. Cambridge University Press, 2011.