UNO Gets Easier for a Single Player

Palash Dey[†], Prachi Goyal[†], and Neeldhara Misra[†]

[†]Indian Institute of Science, Bangalore

{palash | prachi.goyal | neeldhara}@csa.iisc.ernet.in

Abstract

This work is a follow up to [1, FUN 2010], which initiated a detailed analysis of the popular game of UNO®. We consider the solitaire version of the game, which was shown to be NP-complete. In [1], the authors also demonstrate a $n^{O(c^2)}$ algorithm, where c is the number of colors across all the cards, which implies, in particular that the problem is polynomial time when the number of colors is a constant.

In this work, we propose a kernelization algorithm, a consequence of which is that the problem is fixed-parameter tractable when the number of colors is treated as a parameter. This removes the exponential dependence on c and answers the question stated in [1] in the affirmative. We also introduce a natural and possibly more challenging version of UNO that we call "All Or None UNO". For this variant, we prove that even the single-player version is NP-complete, and we show a single-exponential FPT algorithm, along with a cubic kernel.

1 Introduction

UNO® is a popular American card game invented in the year 1971, by Merle Robbins. It is a shedding game, where the goal is to get rid of all the cards at hand, while constrained by some simple rules[I].

This paper is motivated largely by the work in [1], which formalizes the game of UNO and studies it from an algorithmic combinatorial game theory perspective. It is also motivated to a smaller extent by the plight of our friend Sheldon who, as it turns out, is a devoted UNO addict. On his birthday this year, Amy gifts him a painstakingly collected set of UNO cards from different parts of the world.

The gift, however, is accompanied by a challenge — Amy asks Sheldon to demolish the entire collection in a single solitaire game. Sheldon is confident from several hours of focused practice. However, this won't be easy, for several reasons: first, the cards were mostly second-hand, and so several cards from several decks are simply missing. Further, while all cards were either red, green, blue, or yellow, the numbers were often printed in the local language. Her collection involved over a thousand and seven hundred and thirty six distinct number symbols. Even though Sheldon could stare at the entire deck all he liked, he could never keep track of all of them!

Having accepted the challenge, and considering a precious lot was at stake (we regret that we are unable to share the exact details here, but we are confident that the interested reader will be able to guess), Sheldon ultimately found it in his best interests to turn to the theory of algorithms for help. This work investigates the game of UNO, and is especially relevant to those who are put in precarious positions by solitaire game challenges.

The UNO game was formally addressed in [1], and several variations were considered. For example, for the multiplayer versions, the authors suggest co-operative versions, where all players help one player finish his (or her) deck; and un-co-operative versions where everyone plays to win. On the other hand, the simplest non-trivial style of playing UNO — involving only a single player trying to discard a given deck — case turned out to be (somewhat

[I]For the non-UNO-player reader: every card has a number and a color, and when one card is discarded, it must have either the same number or the same color as the previous one.

surprisingly) intricate. To begin with, it is already NP-complete. One of the results in [1] shows an intricate $n^{O(c^2)}$ algorithm for the problem, where c is the number of distinct colors in the deck. This does imply that the problem is solvable in polynomial time if the number of colors is a constant. The algorithm has a geometric flavor — it treats every card as a point on a grid, and then performs dynamic programming.

We pursue this question further, and in particular, address the question of whether the problem is *fixed-parameter tractable* when parameterized by c — that is, is there an algorithm whose running time is of the form $f(c) \cdot n^{O(1)}$, so that the exponent of n is independent of c. This a natural direction of improvement, and we are able to answer this question in the affirmative.

Our approach to obtaining a FPT algorithm is kernelization. The algorithm involves an analysis of the structure of a winning sequence, where we first demonstrate that any winning sequence can be remodeled into another one with a limited number of "color chunks", which are maximal subsequences of cards with the same color. Based on this, we are able to formulate a reduction rule that safely removes cards from the game when there are enough involving a particular number. With this, we obtain that any instance of a single-player UNO game can be turned into one where the number of numbers is $O(c^2)$, significantly reducing the size of the game (note that with c colors and only c^2 numbers, we can have at most c^3 distinct cards). Combined with the algorithm in [1], we remark that the problem also admits an algorithm with running time $2^{O(c^2 \log c)} \cdot n^{O(1)}$.

Next, we introduce and study a more challenging form of UNO (we believe it to be more challenging for the player attempting it), which we call "All-Or-None" UNO. Here, the first time a color turns up in a discarding sequence, the player has to either commit to the color or choose to destroy it. If he commits, he must exhaust all cards of the said color right away. Otherwise, the cards that bear this color will be effectively rendered colorless, and they have to be discarded in the future only with the help of the numbers on them. The single-player version of this game turns out to be NP-complete as well, and in this setting we show a single-exponential algorithm and a cubic kernel when parameterized by the number of colors.

The rest of the paper is organized as follows. We first introduce some general notation and establish the setup. Next, we illustrate the fixed-parameter tractable algorithm, and finally describe the kernel. Due to space constraints, some proofs have been omitted. Such statements are marked with a $\star$, and a full version of this work with complete details is appended at the end.

2 Preliminaries

In this section, we state some basic definitions related to our modeling of the game of UNO, and introduce terminology from graph theory and algorithms. We also establish some of the notation that will be used throughout.

We use $\mathbb{N}$ to denote the set of natural numbers. For a natural number n, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$.

To describe running times of we will use the $O^\star$ notation. Given $f : \mathbb{N} \to \mathbb{N}$, we define $O^\star(f(n))$ to be $O(f(n) \cdot p(n))$, where $p(\cdot)$ is some polynomial function. That is, the O^* notation suppresses polynomial factors in the running-time expression.

We will often be dealing with permutations of objects from an universe, and we represent permutations in a sequence notation. In fact, for ease of discussion, we refer to permutations as sequences over the relevant universe. When we want to emphasize the possibility that we have a string over an universe that allows for repetition, we distinguish the situation by calling such a sequence a *word*.

We typically reserve π and σ for sequences, and use the notation $\sigma \sqsubseteq \pi$ to indicate that σ is a contiguous subsequence of π. Our indexing starts at one, so one may speak of the first element of the sequence π and refer to it by $\pi[1]$. For a sequence π of n objects, the notation $\pi[i, j]$ for $1 \leqslant i \leqslant j \leqslant n$ refers to the subsequence starting at the i^{th} element of π and ending at the j^{th} element of π (note that these elements are included in $\pi[i, j]$). We use the notation $\pi \circ \sigma$ to denote the concatenation of π and σ.

The Game of UNO. An UNO card has two attributes — a *color* and a *number*. More formally we define a *card* to be an ordered pair $(x, y) \in C \times B$, where $C = \{1, 2, \ldots, c\}$ is a set of colors and $B = \{1, 2, \ldots, b\}$ is a set of numbers. A collection of cards, or a *deck* is usually denoted by Γ, and when we want to pick a subset of the deck, we use the letter $\wp$. We often refer to the cards by their ordered pair expression. Sometimes, the exact detail of the card is not of interest, in such situations we use variations of the letters $\mathfrak{g}$ and $\mathfrak{h}$ to refer to cards as a whole. We do use $\aleph(\mathfrak{g})$ and $\Im(\mathfrak{g})$ to refer to the color and the number of the card $\mathfrak{g}$, respectively. As a matter of informal convention, we use ℓ, ℓ', and so on, to refer to colors, while we use t, p, q and such to refer to numbers.

While in general, any reasonably finite number of players are welcome to join an UNO game, in this work we focus only on the single player or "solitaire" version of the game. Therefore, we only describe the model in this setting. By and large, we follow the setup suggested in [1]. At the beginning of the game the player is dealt with a set of n cards. We assume that no card is repeated (and in making this choice we deviate[II] from the model proposed in [1]). After playing a certain card in the i^{th} round, for the $(i+1)^{th}$ round, the only valid move that he can make is a card that has the same number or the same color as the one in the previous round.

Colorful UNO-1 (The Solitaire Version)	Parameter: c
Input:	A set Γ of n cards $\{(x_i, y_i) \mid i \in [n]\}$, where $x_i \in \{1, 2, \ldots, c\}$ and $y_i \in \{1, 2, \ldots, b\}$.
Question:	Determine whether the player can *play* all the cards.

We use the adjective "colorful" to imply that the problem is parameterized by the number of colors. We now introduce some notations that we use through out the paper. We note that this problem continues to be NP-complete when no duplicate cards are permitted (it is easy to verify that the reduced instance obtained in [1] indeed creates no repeated cards).

If we encounter, in a sequence of cards, two consecutive cards that have different colors *and* different numbers, then we refer to this unfortunate situation as a *match violation*. We say that a sequence of cards $\pi = \{x_1, x_2, \ldots x_l\}$ is a *feasible playing sequence*, if $\forall i$ where $1 \leqslant i < l$, the cards x_i and x_{i+1} either have a common color or a common number. We refer to a sequence of cards as a *winning sequence* if it is a feasible playing sequence that uses all the input cards.

For a color $\ell \in [c]$, we denote the set of cards whose color is ℓ by $\Gamma[\ell]$. Similarly, for a number $t \in [b]$, we denote the set of cards whose number is t by $\Gamma[t]$. Further, the degree of a color ℓ is the number of cards in the deck that have color ℓ (notationally, we say $d(\ell) := |\Gamma[\ell]|$). Similarly, the degree of a number t is the number of cards in the deck that have the number t (again, we write $d(t) := |\Gamma[t]|$).

UNO: The All-Or-None Version We also introduce an arguably more challenging version of the single-player UNO game. The revised rules require the player to treat cards of the same color in an "all or nothing" spirit. When a card of color r is played for the first time, then the color has to be either *committed* or *destroyed*. If the color is committed, then the player is required to exhaust all cards of color r in his playing sequence before playing a card whose color is different from r. If the color is destroyed, then the player is forbidden from playing two cards of color r consecutively in his playing sequence. Notice that when a color is destroyed, then it cannot be used together at all, so it is as if the card was effectively without a color (hence the terminology). We show that this single-player version of this game is also NP-complete, and it admits a single-exponential FPT algorithm, and a cubic kernel as well.

Graphs In the following, let $G = (V, E)$ and $G' = (V', E')$ be graphs, and $U \subseteq V$ some subset of vertices of G. We introduce only the definitions that we will be using. For any non-empty subset $W \subseteq V$, the subgraph of G induced by W is denoted by $G[W]$; its vertex set is W and its edge set consists of all those edges of E with both endpoints in W. A *path* in a graph is a sequence of distinct vertices $v_0, v_1, \ldots, v_k$ such that (v_i, v_{i+1}) is an edge for all $0 \leqslant i \leqslant (k-1)$. A *Hamiltonian path* of a graph G is a path featuring every vertex of G.

A set U is said to be an *independent set* in G if no two elements of U are adjacent to each other. A set U is said to be a *dominating set* in G if every vertex in $V \setminus U$ is adjacent to some vertex in U. An *independent dominating set* is a set that is both independent and dominating. We use $K_{i,j}$ to denote the complete bipartite graphs where the bipartitions have sizes i and j, respectively. In other words, the vertex set of $K_{i,j}$ is the disjoint union of two independent sets X and Y on i and j vertices, respectively, and for every $x \in X$ and $y \in Y$, (x, y) is an edge. A graph

[II] Having said that, our results scale easily enough when the number of duplicates is bounded by a constant.

is $K_{i,j}$-free if it does not contain $K_{i,j}$ as an induced subgraph. We refer the reader to [2] for details on standard graph theoretic notation and terminology we use in the paper.

Parameterized Complexity A parameterized problem Π is a subset of $\Gamma^* \times \mathbb{N}$, where Γ is a finite alphabet. An instance of a parameterized problem is a tuple (x, k), where k is called the parameter. A central notion in parameterized complexity is *fixed-parameter tractability (FPT)* which means, for a given instance (x, k), decidability in time $f(k) \cdot p(|x|)$, where f is an arbitrary function of k and p is a polynomial in the input size. The notion of a *kernelization* algorithm is formally defined as follows.

Definition 1. **[Kernelization]** *[3, 4] A kernelization algorithm for a parameterized problem $\Pi \subseteq \Gamma^* \times \mathbb{N}$ is an algorithm that, given $(x, k) \in \Gamma^* \times \mathbb{N}$, outputs, in time polynomial in $|x| + k$, a pair $(x', k') \in \Gamma^* \times \mathbb{N}$ such that (a) $(x, k) \in \Pi$ if and only if $(x', k') \in \Pi$ and (b) $|x'|, k' \leqslant g(k)$, where g is some computable function. The output instance x' is called the kernel, and the function g is referred to as the size of the kernel. If $g(k) = k^{O(1)}$ (resp. $g(k) = O(k)$) then we say that Π admits a polynomial (resp. linear) kernel.*

3 The Standard UNO Game

In this section, we argue a polynomial kernel for Colorful UNO-1.

Let (Γ, c, b) be an instance of Colorful UNO-1. We will first need some terminology. Let π be a feasible playing sequence of $\wp \subseteq \Gamma$. For a color $\ell \in [c]$, we say that a maximal contiguous subsequence of cards in π of color ℓ is a ℓ-*chunk* (see Figure 1). The *length* of a chunk is simply the number of cards in the chunk. For a feasible playing sequence π and a color $\ell \in [c]$, the *frequency* of ℓ in π is the number of distinct ℓ-chunks in π. Further, we say that ℓ is *fragmented* in π if its frequency in π is more than c. We use the notation $\sigma \sqsubseteq \pi$ to denote a contiguous subsequence of π.

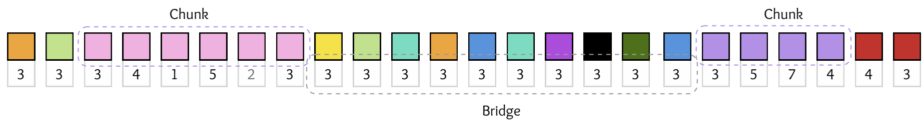

Figure 1: Color chunks and number bridges (see section 4.2 for the notion of a bridge) in a feasible playing sequence.

Our first observation is that fragmented colors can be "fixed", in the following sense. Given a feasible playing sequence of $\wp$ where a color ℓ is fragmented, we claim that there exists another feasible playing sequence where ℓ is not fragmented.

Lemma 1. *Let (Γ, c, b) be an instance of Colorful UNO-1, and let $\ell \in [c]$. Further, let π be a feasible playing sequence of $\wp \subseteq \Gamma$ where ℓ is fragmented. Then, there exists a feasible playing sequence $\pi^{\circlearrowleft}$ where ℓ is not fragmented.*

Proof. Since ℓ is fragmented in π, there are at least $(c + 1)$ ℓ-chunks in π. Let $\pi[x_1, y_1], \ldots, \pi[x_{c+1}, y_{c+1}]$ denote the first $c + 1$ ℓ-chunks in π. Notice that these chunks must be non-overlapping (see Figure 2), that is,

$$\pi[y_i + 1] \neq \pi[x_{i+1} - 1], \text{for all } 1 \leqslant i \leqslant c.$$

Indeed, let $\mathfrak{g}$ and $\mathfrak{h}$ denote the cards $\pi[y_i + 1]$ and $\pi[x_{i+1} - 1]$, respectively. Further, let $\mathfrak{g}^\star$ denote the last card of the i^{th} ℓ-chunk, and let $\mathfrak{h}^\star$ denote the first card of the $(i + 1)^{st}$ ℓ-chunk. Since a chunk is maximal subsequence of cards of the same color, the numbers on the cards $\mathfrak{g}$ and $\mathfrak{h}$ must be equal to the numbers of the cards at $\mathfrak{g}^\star$ and $\mathfrak{h}^\star$, respectively. Therefore, if $\mathfrak{g} = \mathfrak{h}$, then we have a contradiction because we know that the numbers on the cards $\mathfrak{g}^\star$ and $\mathfrak{h}^\star$ are different.

For a ℓ-chunk $\sigma \sqsubset \pi$, denote by $E(\sigma)$ the cards that appear just before and just after σ in π. Notice that $|E(\sigma)| \leqslant 2$, where $|E(\sigma)| = 1$ exactly when σ coincides with the start or end of π. Now consider the set:

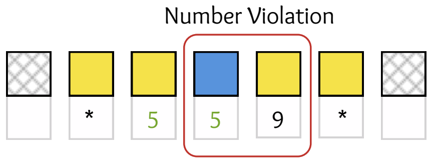

Figure 2: This depicts why two ℓ chunks cannot overlap

$$\mathcal{B} = E(\pi[x_1, y_1]) \cup \ldots \cup E(\pi[x_{c+1}, y_{c+1}])$$

At most two of the chunks under the consideration have only one card in $\mathcal{B}$, and since we have that $E(\pi[x_i, y_i]) \cap E(\pi[x_j, y_j])$ for all $1 \leqslant i \neq j \leqslant c+1$, it follows that $|\mathcal{B}| \geqslant 2(c+1) - 2 = 2c$. Every card in $\mathcal{B}$ has a color from $[c] \setminus \{\ell\}$, therefore, we have $(c-1)$ colors distributed over $2c$ cards. Therefore, there exists a color $\ell^\star \in [c] \setminus \{\ell\}$ such that at least three cards in $\mathcal{B}$ have color $\ell^\star$.

Let g_1, g_2, g_3 be three cards in $\mathcal{B}$ that have color $\ell^\star$, which we will now refer to as *friendly* cards. Let the associated chunks be $\pi[a_1, b_1], \pi[a_2, b_2], \pi[a_3, b_3]$. One of the following scenarios must hold:

- At least two among these three chunks have a friendly card appearing before the chunk, or,
- At least two among these three chunks have a friendly card appearing after the chunk.

Assume that the first scenario holds. Without loss of generality, let the chunks that have a friendly card appearing before them be $\pi[a_1, b_1]$ and $\pi[a_2, b_2]$. Then, consider the playing sequence (see Figure 3):

$$\pi^{\circlearrowleft} := \pi[1, a_1 - 1] \circ \pi[a_1 - 1] \circ \pi[a_2 - 1] \circ \overleftarrow{\pi}[a_2 - 2, a_1] \circ \pi[a_2, n].$$

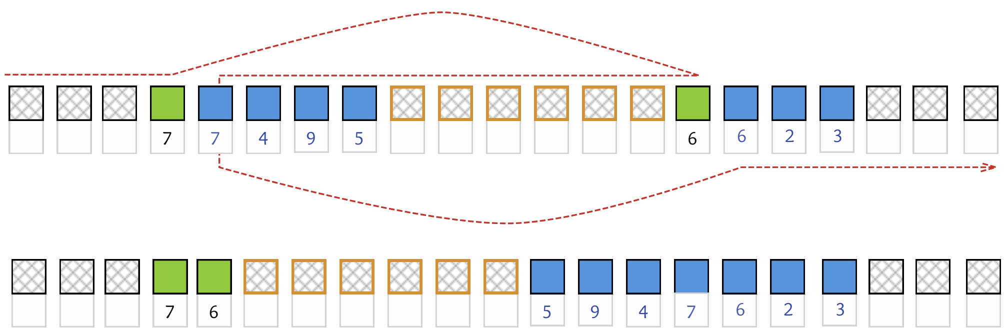

Figure 3: Remodeling a playing sequence to merge far-apart color chunks.

Note that the two ℓ-chunks that we were considering in π are merged in $\pi^{\circlearrowleft}$. This rearrangement, therefore, strictly decreases the frequency of ℓ. Observe that the second scenario is similar to the first, for instance, we may begin with the permutation π in reverse and rearrange the sequence as above.

We conclude the proof by repeating this argument for as long as the frequency of ℓ is greater than c. □

Lemma 1 has the following consequence: if we have a YES-instance of COLORFUL UNO-1, then there exists a feasible playing sequence exhausting all the cards where no color is fragmented. Indeed, starting with an arbitrary winning sequence, we may appeal to lemma 1 for every fragmented color separately, by observing that when the lemma is invoked for a color ℓ, we do not increase the frequency of any other color. This brings us to the following corollary.

Corollary 1. *Let* (Γ, c, b) *be a* YES*-instance of* COLORFUL UNO-1*. Then, there exists a winning sequence where the frequency of* ℓ *is at most* c *for all* $\ell \in [c]$*. In particular, this sequence admits at most* c^2 *chunks.*

Note that in the setting where there are no duplicate cards, the number of cards in any instance (Γ, c, b) of COLORFUL UNO-1 is at most bc. Therefore, to obtain a polynomial kernel, it suffices to obtain a bound on b in terms of c.

We first introduce some terminology. For $\ell \in [c]$, the *degree* of ℓ is simply the number of cards in Γ that have color ℓ. Notationally, the degree of ℓ, which we will denote by $d(\ell)$, is simply $|\Gamma[\ell]|$. Note that $d(\ell) \leqslant b$ for all $\ell \in [c]$. For any feasible sequence, the cards that occur at the beginning and end of chunks are called *critical cards*, while all remaining cards are called *wildcards*. We begin with an easy observation that is based on the fact that the predecessor and successor of a wildcard necessarily match with each other. Therefore, the removal of such a card does not "break" the associated feasible sequence. Similarly, we have that a wildcard at the start or end of a sequence can always be removed without affecting the sequence. This allows us to conclude the following.

Proposition 1. *Let* $\wp \subseteq \Gamma$ *be a set of cards, and let* π *be a feasible sequence of* $\wp$*. If we remove a wildcard* $\mathfrak{g}$ *from* π*, then we still have a feasible sequence of* $\wp \setminus \{\mathfrak{g}\}$*.*

We now propose the following kernelization algorithm. We will consider pairs of cards. Initially, we say that all cards are *unlabeled*. Now, for $1 \leqslant \ell, \ell' \leqslant c$, we consider

$$\mathcal{S}_{\ell,\ell'} := \{\langle \mathfrak{g}, \mathfrak{h} \rangle \mid \mathfrak{I}(\mathfrak{g}) = \mathfrak{I}(\mathfrak{h}) \text{ and } \mathfrak{g} \in \Gamma[\ell], \mathfrak{h} \in \Gamma[\ell']\}.$$

If the number of unlabeled card pairs in $\mathcal{S}_{\ell,\ell'}$ is more than $5c$, then we arbitrarily chose $5c$ unlabeled card pairs and given them the label $[\ell, \ell']$. Otherwise, we mark all the available unlabeled card pairs with the label $[\ell, \ell']$. At the end of this procedure, we say that *a card is unlabeled* if it doesn't belong to any labeled pair. We now propose the following reduction rule.

Reduction Rule 1. *Let* $\mathfrak{g}$ *be a an unlabeled card, where* $\aleph(\mathfrak{g}) = \ell$*. If* $d(\ell) > c + 1$*, then delete* $\mathfrak{g}$ *from* Γ*.*

Note that with one application of Reduction Rule 1, we have that if a card $\mathfrak{g}$ of color ℓ was removed, then $d(\ell)$ in the reduced game is at least $(c + 1)$. Indeed, if not, then the degree of ℓ in the original game was at most $(c + 1)$, which contradicts the pre-requisite for removing $\mathfrak{g}$ from the game in Reduction Rule 1. Thus, it is easy to arrive at the following.

Proposition 2. *Let* $(\Gamma^\star, c^\star, b^\star)$ *be the reduced instance corresponding to* (Γ, c, b)*, and let* $\Upsilon \subseteq [c]$ *denote the set of colors on the cards that were deleted by the application of Reduction Rule 1. If* $d(\ell) < c + 1$ *in* $\Gamma^\star$*, then* $\ell \notin \Upsilon$*.*

We are now ready to establish the correctness of this reduction rule.

Lemma 2. *Let* $(\Gamma^\star, c^\star, b^\star)$ *be the reduced instance corresponding to* (Γ, c, b)*. We claim that* $(\Gamma^\star, c^\star, b^\star)$ *is a* YES *instance if, and only if,* (Γ, c, b) *is a* YES *instance.*

Proof. In the forward direction, let π be a winning sequence for (Γ, c, b). By Corollary 1 we may assume, without loss of generality, that π has at most c^2 chunks. Consider $\pi^\star$ obtained by projecting π onto $\Gamma^\star$. Suppose now that $\pi^\star$ is not a winning sequence. Clearly, $\pi^\star$ exhausts all cards in $\Gamma^\star$, therefore if the sequence is not winning, it must be because of a match violation. Let $\pi^\star[i]$ and $\pi^\star[i + 1]$ be such a violation. Note that these two cards must have been in different and adjacent chunks in π. Let ℓ be the color in the chunk of $\pi^\star[i]$ in π, and let ℓ' be the color in the chunk of $\pi^\star[i + 1]$ in π. Note that both these chunks in π contained cards that were deleted by Reduction Rule 1 (otherwise this violation would be present in π).

This implies that there exist, in Γ, at least $5c$ card pairs $\langle \mathfrak{g}, \mathfrak{h} \rangle$ such that each pair of cards share the same number, and these cards were labeled with the pair $[\ell, \ell']$ by the kernelization algorithm. Since Reduction Rule 1 never deletes a labeled pair, these card pairs, say $\wp$, are also present in $\Gamma^\star$. Observe that $\pi^\star$ has at most c ℓ-chunks, and at most c ℓ' chunks. Therefore, at most $2c$ cards in $\Gamma^\star(\ell)$ are critical, and similarly, at most $2c$ cards in $\Gamma^\star(\ell')$ are critical, with respect to $\pi^\star$. Therefore, in $\wp$, there is at least one pair of wildcards, say (x, y). We now use this pair to fix the match violation by inserting x after $\pi^\star[i]$ and inserting y before $\pi^\star[i+1]$. This reduces the total number of violations in $\pi^\star$ by one, and doesn't create any new violations by Proposition 1.

This shows that every violation can be iteratively fixed to obtain a winning sequence $\pi^\star$ for the instance $(\Gamma^\star, c^\star, b^\star)$.

We now turn to the reverse direction. Let $\pi^\star$ be a winning sequence for the instance $(\Gamma^\star, c^\star, b^\star)$. We may assume without loss of generality, by Lemma 1, that for every $\ell \in [c^\star]$, there are at most $c^\star \leqslant c$ ℓ-chunks in $\pi^\star$.

To obtain a winning sequence for (Γ, c, b), we have to insert all the cards in $\Gamma \setminus \Gamma^\star$ into the sequence π. Consider $\mathfrak{g} \in \Gamma \setminus \Gamma^\star$, and assume that the color of $\mathfrak{g}$ is ℓ. Notice that $\mathfrak{g}$ is a card that was deleted by Reduction Rule 1. This implies ℓ has degree at least $c+1$ in $\Gamma^\star$. Since there are at most c ℓ-chunks, by the pigeon-hole principle, there is a ℓ-chunk of length at least two. Now, since $\pi^\star$ admits some ℓ-chunk of length at least two, then $\mathfrak{g}$ can be inserted inside this chunk (specifically, making it a wildcard in the resulting sequence). Repeating this argument for every card in $\Gamma \setminus \Gamma^\star$, we are done in the reverse direction. □

We now argue the size of the kernel obtained after the application of Reduction Rule 1.

Theorem 1. COLORFUL UNO-1 *admits a kernel on* $O(c^3)$ *cards.*

Proof. Let $(\Gamma^\star, c^\star, b^\star)$ be the reduced instance corresponding to (Γ, c, b). The equivalence of these instances is given by Lemma 2. Now we analyze $|\Gamma^\star|$. Fix $\ell \in [c^\star]$. If $d(\ell)$ in Γ was at most $(c+1)$, then there are at most $(c+1)$ cards of color ℓ in $\Gamma^\star$ as well. Otherwise, observe that the number of labeled cards in $\Gamma[\ell]$ is at most $5c(c-1)$. Since we are considering the case when $d(\ell)$ is strictly greater than $(c+1)$ in Γ, we have that all unlabeled cards in $\Gamma[\ell]$ are deleted by Reduction Rule 1. Therefore, the degree of ℓ in $\Gamma^\star$ is at most $5c(c-1)$. Since there are $c^\star \leqslant c$ colors in total, evidently the total number of cards is bounded by $5c^2(c-1) = O(c^3)$. □

4 The All-Or-None UNO Game

In this section, we consider the game of ALL-OR-NONE UNO. We begin with a mathematical formulation of the game. An instance of ALL-OR-NONE UNO consists of a set of cards χ from $[c] \times [b]$, and a sequence of cards $\pi := \mathfrak{g}_1 \cdots \mathfrak{g}_n$ is a valid playing sequence if, for every $\ell \in [c]$, one of the following is true (see also fig. 4):

- All cards of color ℓ appear as a contiguous subsequence of π, or
- No two cards of color ℓ appear consecutively in π.

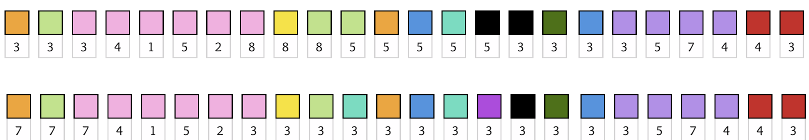

Figure 4: Top: An invalid run of All-Or-None UNO (the green card is violating the rules). Below: A valid run of All-Or-None UNO

We say that the player wins his game if there is a valid playing sequence that exhausts all the cards, and as before, such a sequence is called a winning sequence. It turns out that determining if a player can win in ALL-OR-NONE UNO is NP-hard, and we show this first. Thereafter, we prove that the question admits a single-exponential FPT algorithm and a cubic kernel when parameterized by the number of colors.

4.1 NP-Hardness

In this section, we show that ALL-OR-NONE UNO is NP-complete, by a simple reduction from the HAMILTONIAN PATH problem on general graphs.

Lemma 3. ALL-OR-NONE UNO *is* NP-complete.

Proof. Clearly the problem is in NP. To prove NP-hardness, we reduce it from the Hamiltonian Path (HP) problem.

Given an HP instance G=(V,E), we construct an ALL-OR-NONE UNO instance as follows. For each edge $e = \{u, v\} \in E$, we introduce two cards $(u, e), (v, e)$ (see Figure 5).

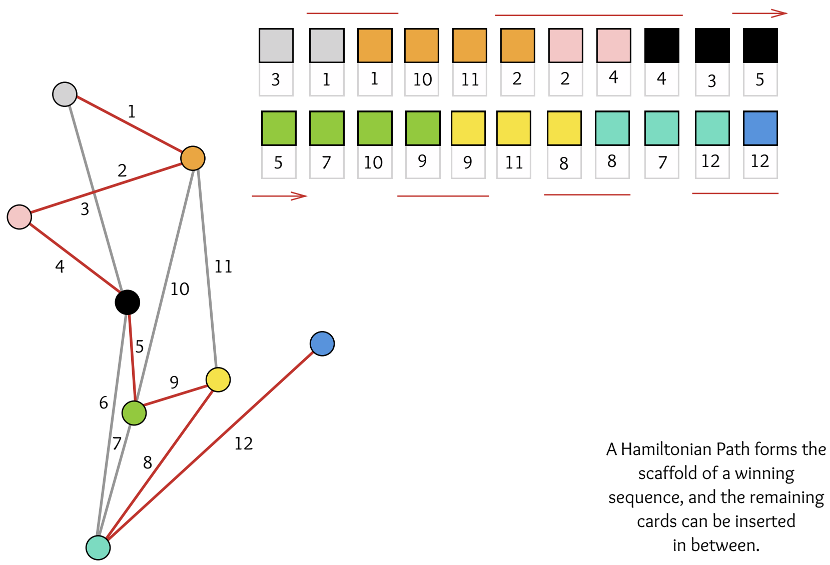

Figure 5: Illustrating the NP-hardness reduction for ALL-OR-NONE UNO

Now we show that HP is an YES instance if, and only if, the corresponding ALL-OR-NONE UNO is an YES instance. Suppose, G has a Hamiltonian path, say, $u_1, u_2, \ldots, u_n$. Let e_i be the edge between u_i and u_{i+1} for all $i \in \{1, \ldots, n-1\}$. Then a winning sequence is all cards with first coordinate u_1 except (u_1, e_1), followed by (u_1, e_1), followed by (u_2, e_1), followed by rest of the cards with first coordinate u_2 except (u_2, e_2), and so on.

For reverse direction, suppose there is a winning sequence. Then first notice that the player has to commit every color since an none of the cards can be a bridge. To see this, notice that if a card is a bridge, then it must have same number as the cards that are just before and after it in the sequence. But this is not possible because there is no three cards with same number, as the numbers correspond to edges in G and occur on exactly two cards. Hence the player will commit every color. Let the color chunks appears in the sequence $u_1, u_2, \ldots, u_n$. Then there must be a number e_i which bridges the color chunks u_i and u_{i+1} for every $i \in \{1, 2, \ldots, n-1\}$. Hence the vertex sequence $u_1, u_2, \ldots, u_n$ forms a Hamiltonian path in G. □

4.2 A Single-Exponential FPT Algorithm

As before, we analyze a YES-instance of ALL-OR-NONE UNO. Let (Γ, c, b) be an instance of ALL-OR-NONE UNO. Let π be a feasible playing sequence of $\wp \subseteq \Gamma$. The most striking feature of π is the following: for any $\ell \in [c]$, π has either exactly one ℓ-chunk, or all ℓ-chunks in π are of length one.

We will need some additional terminology, mostly to observe the behavior of maximal subsequences from the lens of numbers rather than colors. If σ is a sequence of cards, we use $\sigma^\dagger$ to refer to the sequence σ without the first and last cards in σ, that is, $\sigma^\dagger := \sigma[2, |\sigma| - 1]$. Note that if σ is a sequence of length at most two, $\sigma^\dagger$ is the empty sequence.

Fix $t \in [b]$, and let σ be a maximal contiguous subsequence of cards in π with number t. When $|\sigma| > 2$, we call $\sigma^\dagger$ a t-*bridge*. The *length* of a bridge is simply the number of cards in the bridge. For a feasible playing sequence π and a number $t \in [b]$, the *frequency* of t in π is the number of distinct t-bridges in π. Further, we say that t is *broken* in π if its frequency in π is more than one.

Our first task here is to fix broken numbers.

Lemma 4. *Let* (Γ, c, b) *be an instance of* ALL-OR-NONE UNO*, and let* $t \in [b]$*. Further, let* π *be a feasible playing sequence of* $\wp \subseteq \Gamma$ *where* t *is broken. Then, there exists a feasible playing sequence* $\pi^{\circlearrowleft}$ *where* t *is not broken.*

Proof. If t is broken in π, then π admits at least two t-bridges, say $\pi[i_1, j_1]$ and $\pi[i_2, j_2]$. Notice that the cards $\pi[i_1 - 1]$ and $\pi[j_1 + 1]$ have the same number, by the definition of a bridge. We may therefore "pluck out" the bridge $\pi[i_1, j_1]$ (without affecting feasibility) and insert it into the second t-bridge. In particular, we are considering the remodeled sequence (see Figure 6):

$$\pi^{\circlearrowleft} := \pi[1, i_1] \circ \pi[j_1] \circ \pi[j_1 + 1, i_2] \circ \pi[i_1 + 1, j_1 - 1] \circ \pi[i_2 + 1, n],$$

and repeating this operation for every pair of t-bridges eventually merges them all, and in the resulting sequence t is not broken any more.

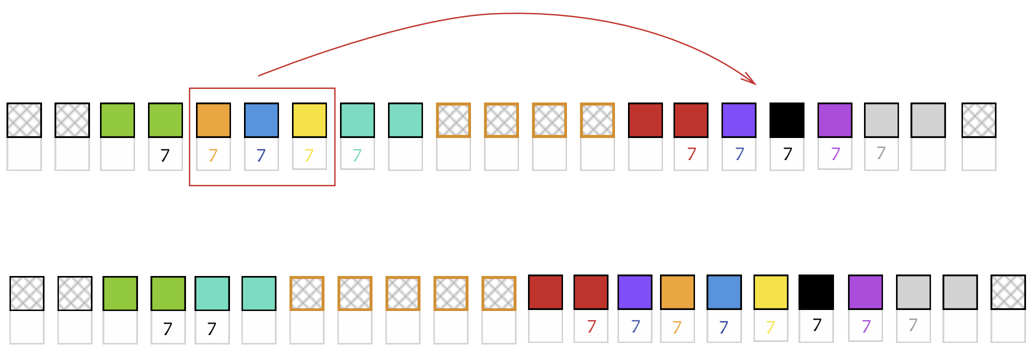

Figure 6: Rearranging a card sequence to merge disjoint bridges.

□

Now, a winning sequence π can always be rearranged to look like a sequence of chunks that are either consecutive, or are chained together by distinct number bridges. Note that two number bridges cannot be consecutive, by definition. We state this explicitly in the following corollary.

Corollary 2. *Let* (Γ, c, b) *be a* YES*-instance of* ALL-OR-NONE UNO*. Then, there exists a winning sequence of the form* $\eta_1, \ldots, \eta_r$*, where every* η_i *is either a* ℓ*-chunk for some* $\ell \in [c]$ *or a* t*-bridge for some* $t \in [b]$*, and no two chunks or bridges are associated with the same color or number, respectively.*

We are now ready to describe the algorithm. The starting point is the fact that every color is going to either appear as a non-trivial chunk or will be spread out over at most $(c + 1)$ bridges. We begin by guessing the colors that will be committed, that is, those that will feature in non-trivial chunks. We denote these colors by $X \subseteq [c]$. For the remaining colors, we consider the numbers that they appear with in the deck, that is, we let

$$Y := \bigcup_{\ell \in [c] \setminus X} \{\mathfrak{I}(\mathfrak{g}) \mid \mathfrak{g} \in \Gamma[\ell]\}.$$

Notice that if $|Y| > |X| + 1$, then we can stop and say NO, by Corollary 2. Otherwise, we have to try and find a winning sequence of cards where all cards that have a color from X appear in exhaustive chunks while all remaining cards appear in bridges. For simplicity, we think of this as simply finding an arrangement of elements in $X \cup Y$ that can be "pulled back" to a winning sequence. Specifically, let us call a sequence over $X \cup Y$ a *motif*. A motif $\sigma = \alpha_1, \ldots, \alpha_{|X|+|Y|}$ is said to be *feasible* if, to begin with, if $\alpha_i \in Y$, then $\alpha_{i+1} \in X$, and further, for every $\alpha_i \in X$ we have cards $\mathfrak{h}_i \neq \mathfrak{g}_i$ such that:

- For all $\alpha_i \in X$, $\aleph(\mathfrak{h}_i) = \aleph(\mathfrak{g}_i) = \alpha_i$.
- If $\alpha_i \in X$ and $\alpha_{i+1} \in X$, $\mathfrak{I}(\mathfrak{g}_i) = \mathfrak{I}(\mathfrak{h}_{i+1})$, and $\aleph(\mathfrak{g}_i) = \alpha_i$ and $\aleph(\mathfrak{h}_{i+1}) = \alpha_{i+1}$.
- If $\alpha_i \in Y$, and $1 < i < r + s$, then $\mathfrak{I}(\mathfrak{g}_{i-1}) = \alpha_i = \mathfrak{I}(\mathfrak{h}_{i+1})$, and $\aleph(\mathfrak{g}_{i-1}) = \alpha_{i-1}$, and $\aleph(\mathfrak{g}_{i+1}) = \alpha_{i+1}$.
- If $\alpha_1 \in Y$, then $\alpha_1 = \mathfrak{I}(\mathfrak{h}_2)$, and $\aleph(\mathfrak{g}_2) = \alpha_2$.
- If $\alpha_{|X|+|Y|} \in Y$, then $\alpha_{|X|+|Y|-1} = \mathfrak{I}(\mathfrak{h}_{|X|+|Y|-1})$, and $\aleph(\mathfrak{g}_{|X|+|Y|-1}) = \alpha_{|X|+|Y|-1}$.

The cards $\{\mathfrak{h}_i, \mathfrak{g}_i \mid i \text{ such that } \alpha_i \in X\}$ will be referred to as the *witnesses of feasibility*. We now claim that a winning sequence over Γ suggests a feasible motif over $(X \cup Y)$, and conversely.

Lemma 5. *The instance (Γ, c, b) has a winning sequence π if, and only if, for some $X \subseteq [c]$ there is a feasible motif σ of $(X \cup Y)$ (where Y is defined as before).*

Proof. Indeed, suppose π is a winning sequence where X are the committed colors. The chunks and bridges of π naturally form a word over $(X \cup Y)$. We note that the word is a sequence of elements from $(X \cup Y)$ — certainly, chunks are not repeated in the word because of the rules of ALL-OR-NONE UNO and a bridge of a particular number can be assumed to appear at most once because of Corollary 2.

On the other hand, the fact that a feasible motif σ leads to a winning sequence is quite clear: we lay out the all cards that have colors in X as dictated by the witnesses of feasibility, and call this sequence $\pi^\star$. If a card with a color ℓ in X is missing, and ℓ appears at location i in σ, then we insert the missing card between the witnesses $\mathfrak{h}_i$ and $\mathfrak{g}_i$ (note that these cards are obliged to be of the same color, by the definition of feasibility). Now, any missing card must have a number $t \in Y$, and by definition of feasibility, since t features in σ, we will have two consecutive cards whose number is t in $\pi^\star$. We insert all missing cards whose number is t (that is, all cards in $\Gamma[t]$) at this location, and the lemma follows once we repeat this process for all missing cards. ☐

Note that it now suffices to check if X admits a feasible motif. While this can be done by exploring all arrangements of the motifs, this proposition will lead us to an algorithm whose running time $O(c!)$, which is not single-exponential in c. Therefore, to keep the time in check, we reduce this task to the problem of finding a path in a restricted sense, and it turns out that the latter can be determined by dynamic programming [5].

We now describe the details of this reduction.

Let s and t denote $|X|$ and $|Y|$, respectively. Further, let $X := \{\ell_1, \ldots, \ell_s\}$, and let $Y := \{p_1, \ldots, p_t\}$. We construct the following auxiliary graph $\mathcal{H}$ based on X and Y (see Figure 7).

- The vertex set of $\mathcal{H}$ is a disjoint union of $3|X| + |Y|$ sets, which we denote as follows:

  $$X_1^L, X_1^{\boxtimes}, X_1^R, \ldots, X_s^L, X_t^{\boxtimes}, X_t^R, Y_1, \ldots, Y_t$$

  For all $1 \leqslant i \leqslant s$, the sets X_i^L, $X_i^{\boxtimes}$ and X_i^R each contain one vertex $v_x^i[L]$, $v_x^i[\boxtimes]$ and $v_x^i[R]$ for every card $x \in \Gamma[\ell_i]$. Note that we have at most b cards in these sets since there are no repeated cards. For all $1 \leqslant i \leqslant t$, the set Y_i contains one vertex $y^i[j, j']$ for every (ordered) pair of cards of the form (ℓ_j, p_i), $(\ell_{j'}, p_i)$ in $\Gamma \times \Gamma$ where $1 \leqslant j \neq j' \leqslant s$. In the discussion that follows, for notational clarity, we will drop the i-superscript, although we will always state these vertices in the context of the color class that they are in, which will disambiguate them.

- The adjacencies in $\mathcal{H}$ are as follows.

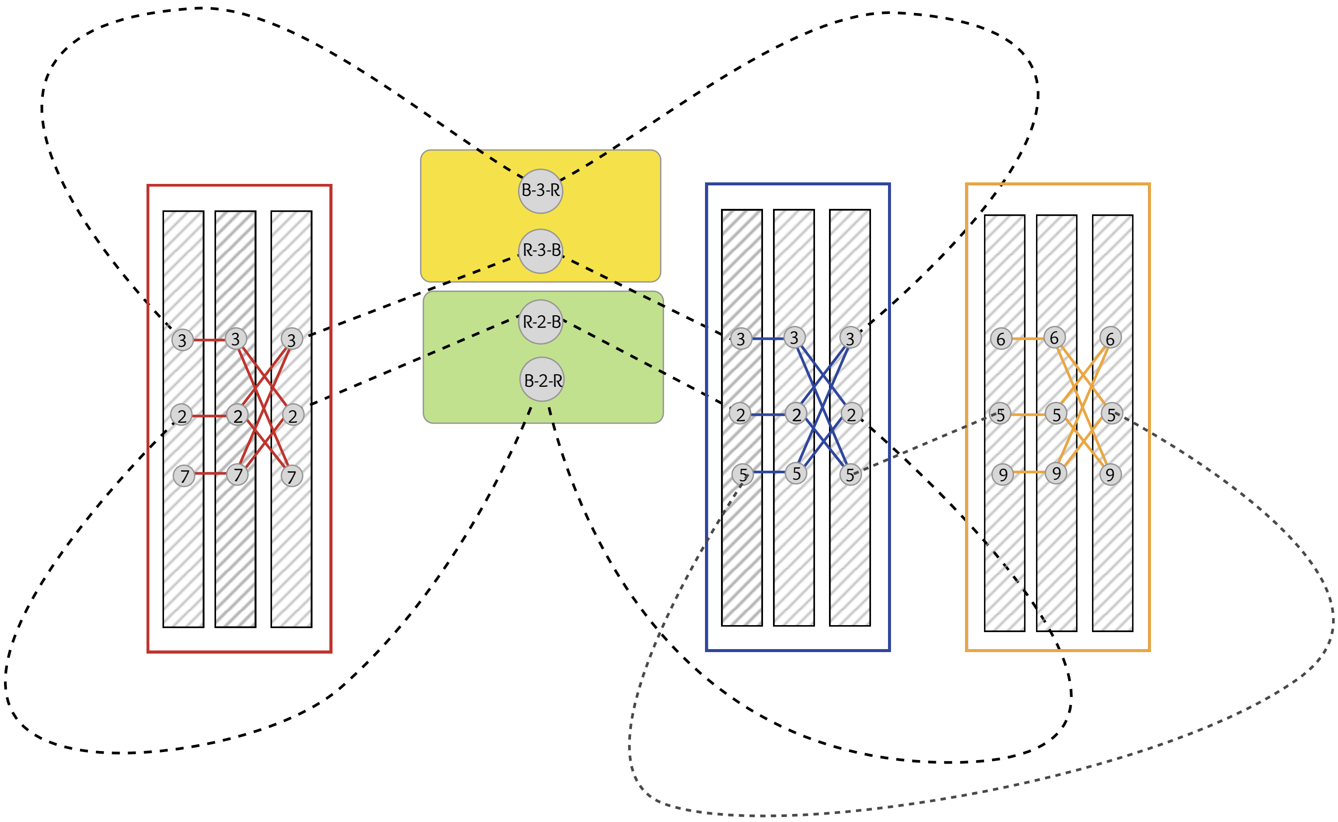

Figure 7: Reducing ALL-OR-NONE UNO to an instance of COLORFUL PATH.

- The vertices within a set have no edges between them.
- For all $i \in [s]$, and for every $x \in \Gamma[\ell_i]$, add an edge between $v_x \in X_i^L$, and $v_x \in X_i^{\boxtimes}$. Further, add edges between $v_x \in X_i^{\boxtimes}$ and $v_y \in X_i^R$ if, and only if, $x, y \in \Gamma[\ell_i]$ and $x \neq y$.
- For every $1 \leqslant i \neq j \leqslant s$, add an edge between $v_x \in X_i^R$ and $v_y \in X_j^L$ if, and only if, $\mathfrak{I}(v_x) = \mathfrak{I}(v_y)$.
- For every $1 \leqslant i \leqslant t$, and for every (ordered) pair of cards of the form (ℓ_j, p_i), $(\ell_{j'}, p_i)$ where $1 \leqslant j \neq j' \leqslant s$, we let x denote the card (ℓ_j, p_i) and y denote the card $(\ell_{j'}, p_i)$. Now we make $y[j, j']$ adjacent to both $v_x \in X_j^R$ and $v_y \in X_{j'}^L$.

A *colorful path* in a graph with a given partition of the vertex set is a path that visits every part exactly once. Our main claim here, roughly speaking, is that a feasible motif implies the existence of a colorful path[III] in $\mathcal{H}$. More formally, we have the following lemma.

Lemma 6. *Let $X, Y, \mathcal{H}$ be as defined above. Then there is a feasible motif of $(X \cup Y)$ if and only if the graph $\mathcal{H}$ has a colorful path with respect to the partition $(X_1^L, X_1^{\boxtimes}, X_1^R, \ldots, X_s^L, X_t^{\boxtimes}, X_t^R, Y_1, \ldots, Y_t)$.*

Proof. In the forward direction, suppose there is a feasible motif over $(X \cup Y)$. As before, we let $s := |X|$ and $t := |Y|$. Let the motif be $\alpha_1, \alpha_2, \ldots, \alpha_{s+t}$.

For every $\alpha_i \in X$ such that $\alpha_{i+1} \in X$, let $x := \mathfrak{h}_i$ and let $y := \mathfrak{g}_i$. Choose the vertex v_x from X_i^L and v_x from $X_i^{\boxtimes}$, and also choose the vertex v_y from X_i^R. For every $\alpha_i \in Y$, let r denote α_i, and let ℓ_j, ℓ_j' denote α_{i-1} and α_{i+1}, respectively. Choose the vertex $y[j, j']$ from Y_r. Arrange the vertices chosen for each α_i in that order according to the ordering specified by σ. It follows from the definition of "witnesses of feasibility" that this suggested ordering is in fact a colorful path.

[III]The path is expected to be colorful with respect to the canonical partitions.

In the reverse direction, consider a colorful path in $\mathcal{H}$ and observe that vertices in $X_i^{\boxtimes}$ are necessarily preceded and followed by vertices in X_i^L and X_i^R. It is also easy to see that a vertex of Y_r is sandwiched between vertices from X_i^R and X_j^L for some (i, j). Consider an ordering of X obtained according to the order of the vertices $X_i^{\boxtimes}$ in the colorful path, and insert p_r between (ℓ_i, ℓ_j) if Y_r is preceded and succeeded by vertices from X_i^R and X_j^L respectively. It is easy to check from the construction that this motif is feasible. □

We can now establish the following.

Theorem 2. *There is an algorithm that decides* ALL-OR-NONE UNO *in time* $O^\star(17^c)$.

Proof. By Lemma 5, the input instance has a winning sequence π if, and only if, for some $X \subseteq [c]$ there is a feasible motif σ of $(X \cup Y)$ (where Y is defined as before). For every $|X| \subseteq [c]$, we have to find a colorful path of length $3|X| + |Y| \leqslant 3|X| + |X| + 1$ (recall that $|Y| \leqslant |X| + 1$). By [5, Lemma 3.1], this can be done in time $O((4|X| + 1) \cdot 2^{(4|X|+1)} \cdot |E(\mathcal{H})|$. The running time, therefore, can be given by:

$$\sum_{X \subseteq [c]} 2^{(4i+1)} \cdot (4i+1) \cdot |E(\mathcal{H})| \leqslant \sum_{i=0}^{c} \binom{c}{i} \cdot 2^{(4i+1)} \cdot (4i+1) \cdot |E(\mathcal{H})| \leqslant 2(4c+1) \cdot |E(\mathcal{H})| \sum_{i=0}^{c} \binom{c}{i} \cdot 2^{(4i)} = O^\star(17^c).$$

□

4.3 A Cubic Kernel

Despite the new rules in place (or thanks to them), ALL-OR-NONE UNO admits a cubic kernel. Recall, from the previous section (Corollary 2) that a winning sequence π can always be rearranged to look like a sequence of chunks that are either consecutive, or are chained together by distinct number bridges. Further, two number bridges cannot be consecutive, by definition. Therefore, if the degree of a color ℓ in Γ is more than $c + 1$, then we may assume that cards of this color appear in a chunk. Let us declare these colors as *pre-committed*, and let $\wp$ denote the set of pre-committed colors.

Note that the number of cards involving colors that are not pre-committed is at most $c(c + 1)$, since there are at most c colors to chose from, and since we are choosing among the colors that are not pre-committed, they can appear with at most $(c + 1)$ numbers (by definition).

Now, let $X \subseteq [b]$ be the set of numbers that appear on the cards that are not pre-committed, that is, let:

$$X := \bigcup_{\ell \notin \wp} \mathfrak{I}(\Gamma[\ell]).$$

It is easy to see that if we have a pre-committed card whose number is not in X, then it safe to delete such a card. Therefore, each pre-committed card can appear with at most $|X| \leqslant c^2$ numbers. Thus, in a reduced instance, $b \leqslant c^2$, leading us to an instance on at most $O(c^3)$ cards. Hence, we have the following.

Theorem 3. ALL-OR-NONE UNO *has a kernel on* $O(c^3)$ *cards.*

5 Conclusions

We showed that deciding the single version of the UNO game is fixed-parameter tractable by showing a cubic kernel for the problem. A natural question is to improve the running time of the FPT algorithm from $2^{O(c^2 \log c)} \cdot n^{O(1)}$, and the size of the cubic kernel from $O(c^3)$.

It is also interesting to see if the multi-player versions are FPT when parameterized by the number of numbers. Also, it is natural to check if there are other parameters that are much smaller than either b or c but that allow for fixed-parameter tractable algorithms. In this context, exploring structural parameters on the natural graph associated with an UNO game would offer a possible direction for future work.

References

[1] Erik D. Demaine et al. “UNO Is Hard, Even for a Single Player”. In: *FUN*. 2010, pp. 133–144.

[2] Reinhard Diestel. *Graph Theory*. Third. Springer-Verlag, Heidelberg, 2005.

[3] Rolf Niedermeier. *Invitation to Fixed Parameter Algorithms (Oxford Lecture Series in Mathematics and Its Applications)*. Oxford University Press, USA, 2006.

[4] J. Flum and M. Grohe. *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag New York, Inc., 2006.

[5] Alon, Yuster, and Zwick. “Color-Coding”. In: *JACM: Journal of the ACM* 42 (1995).