A Fixed-Parameter Algorithm for the Kneser Problem

Ishay Haviv
School of Computer Science, The Academic College of Tel Aviv-Yaffo, Tel Aviv, Israel

Abstract

The Kneser graph $K(n,k)$ is defined for integers n and k with $n \geq 2k$ as the graph whose vertices are all the k-subsets of $\{1,2,\ldots,n\}$ where two such sets are adjacent if they are disjoint. A classical result of Lovász asserts that the chromatic number of $K(n,k)$ is $n-2k+2$. In the computational Kneser problem, we are given an oracle access to a coloring of the vertices of $K(n,k)$ with $n-2k+1$ colors, and the goal is to find a monochromatic edge. We present a randomized algorithm for the Kneser problem with running time $n^{O(1)} \cdot k^{O(k)}$. This shows that the problem is fixed-parameter tractable with respect to the parameter k. The analysis involves structural results on intersecting families and on induced subgraphs of Kneser graphs.

We also study the Agreeable-Set problem of assigning a small subset of a set of m items to a group of ℓ agents, so that all agents value the subset at least as much as its complement. As an application of our algorithm for the Kneser problem, we obtain a randomized polynomial-time algorithm for the Agreeable-Set problem for instances that satisfy $\ell \geq m - O(\frac{\log m}{\log\log m})$. We further show that the Agreeable-Set problem is at least as hard as a variant of the Kneser problem with an extended access to the input coloring.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability; Mathematics of computing → Graph coloring; Mathematics of computing → Combinatorial algorithms; Mathematics of computing → Probabilistic algorithms

Keywords and phrases Kneser graph, Fixed-parameter tractability, Agreeable Set

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.72

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: http://arxiv.org/abs/2204.06761

Funding *Ishay Haviv*: Research supported in part by the Israel Science Foundation (grant No. 1218/20).

Acknowledgements We are grateful to Andrey Kupavskii for helpful discussions and to the anonymous reviewers for their very useful suggestions.

1 Introduction

The Kneser graph $K(n,k)$ is defined for integers n and k with $n \geq 2k$ as the graph whose vertices are all the k-subsets of $[n] = \{1,2,\ldots,n\}$ where two such sets are adjacent if they are disjoint. In 1955, Kneser [17] observed that there exists a proper coloring of the vertices of $K(n,k)$ with $n-2k+2$ colors and conjectured that fewer colors do not suffice, that is, that its chromatic number satisfies $\chi(K(n,k)) = n-2k+2$. The conjecture was proved more than two decades later by Lovász [19] as a surprising application of the Borsuk-Ulam theorem from algebraic topology [2]. Following this result, topological methods have become a common and powerful tool in combinatorics, discrete geometry, and theoretical computer science (see, e.g., [22]). Several alternative proofs of Kneser's conjecture were provided in the literature over the years (see, e.g., [23]), and despite the combinatorial nature of the conjecture, all of them essentially rely on the topological Borsuk-Ulam theorem. One exception is a proof by Matoušek [21], which is presented in a combinatorial form, but is yet inspired by a discrete variant of the Borsuk-Ulam theorem known as Tucker's lemma.

49th International Colloquium on Automata, Languages, and Programming (ICALP 2022).
Editors: Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff;
Article No. 72; pp. 72:1–72:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In the computational Kneser problem, we are given an access to a coloring of the vertices of $K(n,k)$ with $n-2k+1$ colors, and the goal is to find a monochromatic edge, i.e., two vertices with the same color that correspond to disjoint sets. Since the number of colors used by the input coloring is strictly smaller than the chromatic number of $K(n,k)$ [19], it follows that every instance of the problem has a solution. However, the topological proofs of the lower bound on the chromatic number of $K(n,k)$ are not constructive, in the sense that they do not supply an efficient algorithm for finding a monochromatic edge. By an efficient algorithm we mean that its running time is polynomial in n, whereas the number of vertices $\binom{n}{k}$ might be exponentially larger. Hence, it is natural to assume that the input coloring is given as an access to an oracle that given a vertex of $K(n,k)$ returns its color. Alternatively, the coloring can be given by some succinct representation, e.g., a Boolean circuit that computes the color of any given vertex.

The question of determining the complexity of the Kneser problem was proposed by Deng, Feng, and Kulkarni [6], who asked whether it is complete in the complexity class PPA. This complexity class belongs to a family of classes that were introduced by Papadimitriou [26] in the attempt to characterize the mathematical arguments that lie behind the existence of solutions to search problems of TFNP. The complexity class TFNP, introduced in [25], is the class of all total search problems in NP, namely, the search problems in which a solution is guaranteed to exist and can be verified in polynomial running time. Papadimitriou has introduced in [26] several subclasses of TFNP, each of which consists of the total search problems that can be reduced to a problem that represents some mathematical argument. In particular, the class PPA (Polynomial Parity Argument) corresponds to the simple fact that every (undirected) graph with maximum degree 2 that has a vertex of degree 1 must have another degree 1 vertex. Hence, PPA is defined as the class of all problems in TFNP that can be efficiently reduced to the Leaf problem, in which given a succinct representation of a graph with maximum degree 2 and given a vertex of degree 1 in the graph, the goal is to find another such vertex.

In recent years, it has been shown that the complexity class PPA perfectly captures the complexity of several total search problems for which the existence of the solution relies on the Borsuk-Ulam theorem. Indeed, Filos-Ratsikas and Goldberg [9, 10] proved that the Consensus Halving problem with inverse-polynomial precision parameter is PPA-complete and derived the PPA-completeness of some classical problems, such as the Splitting Necklace problem with two thieves and the Discrete Sandwich problem. The PPA-hardness of the Consensus Halving problem was further extended to a constant precision parameter in a recent work of Deligkas, Fearnley, Hollender, and Melissourgos [4]. Another PPA-complete problem, studied in [15] and closely related to the Kneser problem, is the Schrijver problem which given a coloring of the Schrijver graph $S(n,k)$ with fewer colors than its chromatic number asks to find a monochromatic edge. Note that $S(n,k)$ is the induced subgraph of $K(n,k)$ on the collection of all k-subsets of $[n]$ with no two consecutive elements modulo n, and that its chromatic number is equal to that of $K(n,k)$ [27]. Despite the progress in understanding the complexity of total search problems related to the Borsuk-Ulam theorem, the question of [6] on the complexity of the Kneser problem is still open. The question of determining the complexity of its extension to Kneser hypergraphs was recently raised by Filos-Ratsikas, Hollender, Sotiraki, and Zampetakis [11].

The study of the Kneser problem has been recently motivated by its connection to a problem called Agreeable-Set that was introduced by Manurangsi and Suksompong [20] and further studied by Goldberg, Hollender, Igarashi, Manurangsi, and Suksompong [14]. This problem falls into the category of resource allocation problems, where one assigns items from a given collection $[m]$ to ℓ agents that have different preferences. The preferences are given by monotone utility functions that associate a non-negative value to each subset of $[m]$. In the Agreeable-Set setting, the agents act as a group, and the goal is to collectively allocate a subset of items that is agreeable to all of them, in the sense that every agent likes it at least as much as it likes the complement set. The authors of [20] proved that for every ℓ agents with monotone utility functions defined on the subsets of $[m]$, there exists a subset $S \subseteq [m]$ of size

$$|S| \leq \min\left(\left\lfloor \frac{m+\ell}{2} \right\rfloor, m\right) \tag{1}$$

that is agreeable to all agents, and that this bound is tight in the worst case. They initiated the study of the Agreeable-Set problem that given an oracle access to the utility functions of the ℓ agents, asks to find a subset $S \subseteq [m]$ that satisfies the worst-case bound given in (1) and that is agreeable to all agents. Note that for instances with $\ell \geq m$, the collection $[m]$ forms a proper solution.

Interestingly, the proof of [20] for the existence of a solution to the Agreeable-Set problem relies on the chromatic number of Kneser graphs. In fact, the proof implicitly shows that the Agreeable-Set problem is efficiently reducible to the Kneser problem, hence the Kneser problem is at least as hard as the Agreeable-Set problem. An alternative existence proof, based on the Consensus Halving theorem [28], was given by the authors of [14]. Their approach was applied there to show that the Agreeable-Set problem can be solved in polynomial time when the utility functions of the agents are additive and when the number ℓ of agents is a fixed constant. The complexity of the problem in the general case is still open.

1.1 Our Contribution

Our main result concerns the parameterized complexity of the Kneser problem. We prove that the problem is fixed-parameter tractable with respect to the parameter k, namely, it can be solved on an input coloring of a Kneser graph $K(n,k)$ in running time $n^{O(1)} \cdot f(k)$ for some function f.

▶ **Theorem 1.** *There exists a randomized algorithm that given integers n and k with $n \geq 2k$ and an oracle access to a coloring $c : \binom{[n]}{k} \rightarrow [n-2k+1]$ of the vertices of the Kneser graph $K(n,k)$, runs in time $n^{O(1)} \cdot k^{O(k)}$ and returns a monochromatic edge with probability $1 - 2^{-\Omega(n)}$.*

It should be mentioned that the notion of fixed-parameter tractability is used here in a slightly different manner than in the parameterized complexity literature (see, e.g., [3]). This is because Theorem 1 deals with the parameterized complexity of a total search problem, rather than a decision problem, and because its input is given as an oracle access. In fact, borrowing the terminology of this area, our algorithm for the Kneser problem can be viewed as a randomized polynomial Turing kernelization algorithm for the problem (see, e.g., [12, Chapter 22]). Namely, we show that the problem of finding a monochromatic edge in a Kneser graph $K(n,k)$ can essentially be reduced by a randomized efficient algorithm to finding a monochromatic edge in a Kneser graph $K(n',k)$ for $n' = O(k^4)$ (see Section 3.4 for the precise details).

The analysis of the algorithm given in Theorem 1 relies on properties of induced subgraphs of Kneser graphs (see Section 3.1). The proofs involve a stability result due to Hilton and Milner [16] of the celebrated Erdös-Ko-Rado theorem [8] (see Theorem 9) and an idea recently applied by Frankl and Kupavskii [13] in the study of maximal degrees in induced subgraphs of Kneser graphs. An overview of the proof of Theorem 1 is given in Section 1.2.

Our next result provides a simple deterministic algorithm for the Kneser problem that is particularly useful for Kneser graphs $K(n,k)$ with k close to $n/2$. Its analysis is based on the chromatic number of Schrijver graphs [27].

▶ **Theorem 2.** *There exists an algorithm that given integers n and k with $n \geq 2k$ and an oracle access to a coloring $c : \binom{[n]}{k} \rightarrow [n-2k+1]$ of the vertices of the Kneser graph $K(n,k)$, returns a monochromatic edge in running time polynomial in $\binom{n-k+1}{k} \leq n^{\min(k,n-2k+1)}$.*

We proceed by presenting our results on the Agreeable-Set problem. First, as applications of Theorems 1 and 2, using the relation from [20] between the Agreeable-Set and Kneser problems, we obtain the following algorithmic results.

▶ **Theorem 3.** *There exists a randomized algorithm for the Agreeable-Set problem that given an oracle access to an instance with m items and ℓ agents ($\ell < m$), runs in time $m^{O(1)} \cdot k^{O(k)}$ for $k = \lceil \frac{m-\ell}{2} \rceil$ and returns a proper solution with probability $1 - 2^{-\Omega(m)}$.*

▶ **Theorem 4.** *There exists an algorithm for the Agreeable-Set problem that given an oracle access to an instance with m items and ℓ agents ($\ell < m$), returns a proper solution in running time polynomial in $\binom{m-k+1}{k} \leq m^{\min(k,m-2k+1)}$ for $k = \lceil \frac{m-\ell}{2} \rceil$.*

We apply Theorems 3 and 4 to show that the Agreeable-Set problem can be solved in polynomial time for certain families of instances. By Theorem 3, we obtain a randomized efficient algorithm for instances in which the number of agents ℓ is not much smaller than the number of items m, namely, for $\ell \geq m - O(\frac{\log m}{\log\log m})$. By Theorem 4, we obtain an efficient algorithm for instances with a constant number of agents, providing an alternative proof for a result of [14].

We finally explore the relations between the Agreeable-Set and Kneser problems. As already mentioned, there exists an efficient reduction from the Agreeable-Set problem to the Kneser problem [20]. Here we provide a reduction in the opposite direction. However, for the reduction to be efficient we reduce from a variant of the Kneser problem with an extended type of queries, which we call *subset queries* and define as follows. For an input coloring $c : \binom{[n]}{k} \rightarrow [n-2k+1]$ of a Kneser graph $K(n,k)$, a subset query is a pair (i,B) of a color $i \in [n-2k+1]$ and a set $B \subseteq [n]$, and the answer on the query (i,B) determines whether B contains a vertex colored i, that is, whether there exists a k-subset $A \subseteq B$ satisfying $c(A) = i$. We prove the following result (see Section 2.1 for the computational input model of the problems).

▶ **Theorem 5.** *There exists a polynomial-time reduction from the Kneser problem with subset queries to the Agreeable-Set problem.*

1.2 Proof Overview of Theorem 1

We present here the main ideas of our fixed-parameter algorithm for the Kneser problem. Suppose that we are given, for $n \geq 2k$, an oracle access to a coloring $c : \binom{[n]}{k} \rightarrow [n-2k+1]$ of the vertices of the Kneser graph $K(n,k)$ with $n-2k+1$ colors. As mentioned before, since the chromatic number of $K(n,k)$ is $n-2k+2$ [19], the coloring c must have a monochromatic edge. Such an edge can clearly be found by an algorithm that queries the oracle for the colors of all the vertices. However, the running time of such an algorithm is polynomial in $\binom{n}{k}$, so it is not fixed-parameter with respect to the parameter k.

A natural attempt to improve on this running time is to consider a randomized algorithm that picks uniformly and independently at random polynomially many vertices of $K(n,k)$ and checks if any two of them form a monochromatic edge. However, it is not difficult to see that there exist colorings of $K(n,k)$ with $n-2k+1$ colors for which a small fraction of the vertices are involved in all the monochromatic edges, implying that the success probability of this randomized algorithm is negligible on them. To see this, consider the canonical coloring of $K(n,k)$, in which for every $i \in [n-2k+1]$ the vertices $A \in \binom{[n]}{k}$ with $\min(A) = i$ are colored i, and the remaining vertices, those contained in $[n] \setminus [n-2k+1]$, are colored $n-2k+2$. By recoloring the vertices of the last color class with arbitrary colors from $[n-2k+1]$, we get a coloring of $K(n,k)$ with $n-2k+1$ colors such that every monochromatic edge has an endpoint in a collection of $\binom{2k-1}{k}$ vertices. This implies that the probability that two random vertices chosen uniformly and independently from $\binom{[n]}{k}$ form a monochromatic edge may go to zero faster than any inverse polynomial in n.

While the above coloring shows that all the monochromatic edges can involve vertices from a small set, one may notice that this coloring is very well structured, in the sense that each color class is quite close to a trivial intersecting family, i.e., an intersecting family all of whose members share a common element. This is definitely not a coincidence. It is known that large intersecting families of k-subsets of $[n]$ are "essentially" contained in trivial intersecting families. Indeed, the classical Erdös-Ko-Rado theorem [8] asserts that the largest size of an intersecting family of k-subsets of $[n]$ is $\binom{n-1}{k-1}$, attained by, and only by, the n largest trivial families (for $n > 2k$). Moreover, Hilton and Milner [16] proved a stability result for the Erdös-Ko-Rado theorem, saying that if an intersecting family of sets from $\binom{[n]}{k}$ is not trivial then its size cannot exceed $\binom{n-1}{k-1} - \binom{n-k-1}{k-1} + 1$, which is much smaller than the largest possible size of an intersecting family when n is sufficiently larger than k (see Theorem 9 and Remark 10). More recently, it was shown by Dinur and Friedgut [7] that every intersecting family can be made trivial by removing not more than $\tilde{c} \cdot \binom{n-2}{k-2}$ of its members for a constant $\tilde{c}$ (see [18] for an exact $\binom{n-3}{k-2}$ bound on the number of sets that should be removed, provided that $n \geq \tilde{c} \cdot k$ for a constant $\tilde{c}$). Hence, our strategy for finding a monochromatic edge in $K(n,k)$ is to learn the structure of the large color classes which are close to being intersecting. We use random samples from the vertex set of the graph in order to identify the common elements of the trivial families that "essentially" contain these color classes. Roughly speaking, this allows us to repeatedly reduce the size of the ground set $[n]$ of the given Kneser graph and to obtain a small subgraph, whose size depends only on k, that is expected to contain a monochromatic edge. Then, such an edge can simply be found by querying the oracle for the colors of all the vertices of this subgraph.

With the above idea in mind, let us consider an algorithm that starts by selecting uniformly and independently at random polynomially many vertices of $K(n,k)$ and queries the oracle for their colors. Among the $n-2k+1$ color classes, there must be a non-negligible one that includes at least $\frac{1}{n-2k+1} \geq \frac{1}{n}$ fraction of the vertices. It can be shown, using the Chernoff-Hoeffding bound, that the samples of the algorithm can be used to learn a color $i \in [n-2k+1]$ of a quite large color class. Moreover, the samples can be used to identify an element $j \in [n]$ that is particularly popular on the vertices colored i (say, that belongs to a constant fraction of them) in case that such an element exists. Now, if the coloring satisfies that (a) there exists an element j that is popular on the vertices colored i and, moreover, (b) this element j belongs to *all* the vertices that are colored i, then it suffices to focus on the subgraph of $K(n,k)$ induced by the vertices of $\binom{[n]\setminus\{j\}}{k}$. Indeed, condition (b) implies that the restriction of the given coloring to this subgraph uses at most $n-2k$ colors. Since this subgraph is isomorphic to $K(n-1,k)$, its chromatic number is $n-2k+1$, hence it has a monochromatic edge. By repeatedly applying this procedure, assuming that the conditions (a) and (b) hold in all iterations, we can eliminate elements from the ground set $[n]$ and obtain smaller and smaller Kneser graphs that still have a monochromatic edge.

When the ground set becomes sufficiently small, one can go over all the remaining vertices and efficiently find the required edge. We now turn to address the case where at least one of the conditions (a) and (b) does not hold.

For condition (a), suppose that in some iteration the algorithm identifies a color i that appears on a significant fraction of vertices, but no element of $[n]$ is popular on these vertices. In this case, one might expect the color class of i to be so far from being intersecting, so that the polynomially many samples would include with high probability a monochromatic edge of vertices colored i. It can be observed that the aforementioned stability results of the Erdös-Ko-Rado theorem yield that a non-negligible fraction of the vertices of a large color class with no popular element lie on monochromatic edges. However, in order to catch a monochromatic edge with high probability we need here the stronger requirement, saying that a non-negligible fraction of the *pairs* of vertices from this color class form monochromatic edges. We prove that this indeed holds for color classes of $K(n,k)$ that include at least, say, $\frac{1}{n}$ fraction of the vertices, provided that $n \geq \tilde{c} \cdot k^4$ for some constant $\tilde{c}$. The proof uses the Hilton-Milner theorem and borrows an idea of [13] (see Lemma 12 and Corollary 13).

For condition (b), suppose that in some iteration the algorithm identifies a color i of a large color class and an element j that is popular in its sets but does not belong to all of them. In this case, we can no longer ensure that the final Kneser graph obtained after all iterations has a monochromatic edge, as some of its vertices might be colored i. To handle this situation, we show that every vertex $A \in \binom{[n]}{k}$ with $j \notin A$ has a lot of neighbors colored i. Hence, if the algorithm finds at its final step, or even earlier, a vertex A colored i satisfying $j \notin A$, where i and j are the color and element chosen by the algorithm in a previous iteration, then it goes back to the ground set of this iteration and finds using additional polynomially many random vertices from it a neighbor of A colored i, and thus a monochromatic edge (see Lemma 14 and Corollary 15).

To summarize, our algorithm for the Kneser problem repeatedly calls an algorithm, which we refer to as the "element elimination" algorithm, that uses polynomially many random vertices to identify a color i of a large color class. If no element of $[n]$ is popular on this color class then the random samples provide a monochromatic edge with high probability and we are done. Otherwise, the algorithm finds such a popular $j \in [n]$ and focuses on the subgraph obtained by eliminating j from the ground set. This is done as long as the size of the ground set is larger than $\tilde{c} \cdot k^4$ for a constant $\tilde{c}$. After all iterations, the remaining vertices induce a Kneser graph $K(n',k)$ for $n' \leq \tilde{c} \cdot k^4$, and the algorithm queries for the colors of all of its vertices in time polynomial in $\binom{n'}{k} \leq k^{O(k)}$. If the colors that were chosen through the iterations of the "element elimination" algorithm do not appear in this subgraph, then it must contain a monochromatic edge which can be found by an exhaustive search. Otherwise, this search gives us a vertex A satisfying $c(A) = i$ and $j \notin A$ for a color i associated with an element j by one of the calls to the "element elimination" algorithm. As explained above, given such an A it is possible to efficiently find with high probability a vertex that forms with A a monochromatic edge. This gives us a randomized algorithm with running time $n^{O(1)} \cdot k^{O(k)}$ that finds a monochromatic edge with high probability. The full description of the algorithm and its analysis are presented in Section 3.

1.3 Outline

The rest of the paper is organized as follows. In Section 2, we gather several definitions and results that will be used throughout the paper. In Section 3, we present and analyze our randomized fixed-parameter algorithm for the Kneser problem and prove Theorem 1. In Section 4, we present a simple deterministic algorithm for the Kneser problem and prove Theorem 2. Due to space limitation, we omit the proofs of our results on the Agreeable-Set problem, which can be found in the full version of the paper.

2 Preliminaries

2.1 Computational Models

In this work we consider total search problems whose inputs involve functions that are defined on domains of size exponential in the parameters of the problems. For example, the input of the KNESER problem is a coloring $c : \binom{[n]}{k} \to [n-2k+1]$ of the vertices of the Kneser graph $K(n,k)$ for $n \geq 2k$. For such problems, one has to specify how the input is given. We consider the following two input models.

- In the *black-box input model*, an input function is given as an oracle access, so that an algorithm can query the oracle for the value of the function on any element of its domain. This input model is used in the current work to present our algorithmic results, reflecting the fact that the algorithms do not rely on the representation of the input functions.
- In the *white-box input model*, an input function is given by a succinct representation that can be used to efficiently determine the values of the function, e.g., a Boolean circuit or an efficient Turing machine. This input model is appropriate to study the computational complexity of problems, and in particular, to show membership and hardness results with respect to the complexity class PPA.

Reductions form a useful tool to show relations between problems. Let P_1 and P_2 be total search problems. We say that P_1 is (polynomial-time) reducible to P_2 if there exist (polynomial-time) computable functions f, g such that f maps any input x of P_1 to an input $f(x)$ of P_2, and g maps any pair (x,y) of an input x of P_1 and a solution y of $f(x)$ with respect to P_2 to a solution of x with respect to P_1. For problems P_1 and P_2 in the black-box input model, one has to use the notion of *black-box reductions.* A (polynomial-time) black-box reduction satisfies that the oracle access needed for the input $f(x)$ of P_2 can be simulated by a (polynomial-time) procedure that has an oracle access to the input x. In addition, the solution $g(x,y)$ of x in P_1 can be computed (in polynomial time) given the solution y of $f(x)$ and the oracle access to the input x. In the current work we will use black-box reductions to obtain algorithmic results for problems in the black-box input model. For more details on these concepts, we refer the reader to [1, Section 2.2] (see also [5, Sections 2 and 4] for related discussions).

2.2 Kneser and Schrijver Graphs

Consider the following definition.

▶ **Definition 6.** *For a family $\mathcal{F}$ of non-empty sets, let $K(\mathcal{F})$ denote the graph on the vertex set $\mathcal{F}$ in which two vertices are adjacent if they represent disjoint sets.*

For a set X and an integer k, let $\binom{X}{k}$ denote the family of all k-subsets of X. Equipped with Definition 6, the *Kneser graph* $K(n,k)$ can be defined for integers n and k with $n \geq 2k$ as the graph $K(\binom{[n]}{k})$. A set $A \subseteq [n]$ is said to be *stable* if it includes no two consecutive elements modulo n, that is, it forms an independent set in the cycle C_n with the numbering from 1 to n along the cycle. For integers n and k with $n \geq 2k$, let $\binom{[n]}{k}_{\text{stab}}$ denote the collection of all stable k-subsets of $[n]$. The *Schrijver graph* $S(n,k)$ is defined as the graph $K(\binom{[n]}{k}_{\text{stab}})$. Equivalently, it is the induced subgraph of $K(n,k)$ on the vertex set $\binom{[n]}{k}_{\text{stab}}$.

The chromatic numbers of the graphs $K(n,k)$ and $S(n,k)$ were determined, respectively, by Lovász [19] and by Schrijver [27] as follows.

▶ **Theorem 7** ([19, 27]). *For all integers $n \geq 2k$, $\chi(K(n,k)) = \chi(S(n,k)) = n - 2k + 2$.*

The computational search problem associated with the Kneser graph is defined as follows.

▶ **Definition 8.** *In the computational* Kneser *problem, the input is a coloring $c : \binom{[n]}{k} \rightarrow [n-2k+1]$ of the vertices of the Kneser graph $K(n,k)$ with $n-2k+1$ colors for integers n and k with $n \geq 2k$, and the goal is to find a monochromatic edge, i.e., $A, B \in \binom{[n]}{k}$ satisfying $A \cap B = \emptyset$ and $c(A) = c(B)$. In the black-box input model, the input coloring is given as an oracle access that for a vertex A returns its color $c(A)$. In the white-box input model, the input coloring is given by a Boolean circuit that for a vertex A computes its color $c(A)$.*

The existence of a solution to every instance of the Kneser problem follows from Theorem 7.

2.3 Intersecting Families

For integers n and k with $n \geq 2k$, let $\mathcal{F} \subseteq \binom{[n]}{k}$ be a family of k-subsets of $[n]$. We call $\mathcal{F}$ *intersecting* if for every two sets $F_1, F_2 \in \mathcal{F}$ it holds that $F_1 \cap F_2 \neq \emptyset$. The Erdös-Ko-Rado theorem [8] asserts that every intersecting family $\mathcal{F} \subseteq \binom{[n]}{k}$ satisfies $|\mathcal{F}| \leq \binom{n-1}{k-1}$. This bound is tight and is attained, for each $i \in [n]$, by the family $\{A \in \binom{[n]}{k} \mid i \in A\}$. An intersecting family of sets is said to be *trivial* if its members share a common element. Hilton and Milner [16] proved the following stability result for the Erdös-Ko-Rado theorem, providing an upper bound on the size of any non-trivial intersecting family.

▶ **Theorem 9** (Hilton-Milner Theorem [16])**.** *For all integers $k \geq 2$ and $n \geq 2k$, every non-trivial intersecting family of k-subsets of $[n]$ has size at most $\binom{n-1}{k-1} - \binom{n-k-1}{k-1} + 1$.*

▶ Remark 10. The bound given in Theorem 9 is tight. To see this, for an arbitrary k-subset F of $[n]$ and for an arbitrary element $i \notin F$, consider the family $\mathcal{F} = \{A \in \binom{[n]}{k} \mid A \cap F \neq \emptyset,\ i \in A\} \cup \{F\}$. The family $\mathcal{F}$ is intersecting and non-trivial, and its size coincides with the bound given in Theorem 9. Note that $|\mathcal{F}| \leq k \cdot \binom{n-2}{k-2}$, provided that $k \geq 3$.

2.4 Chernoff-Hoeffding Bound

We need the following concentration result (see, e.g., [24, Theorem 2.1]).

▶ **Theorem 11** (Chernoff-Hoeffding Bound)**.** *Let $0 < p < 1$, let $X_1, \ldots, X_m$ be m independent binary random variables satisfying $\Pr[X_i = 1] = p$ and $\Pr[X_i = 0] = 1 - p$ for all i, and put $\overline{X} = \frac{1}{m} \cdot \sum_{i=1}^{m} X_i$. Then, for any $\mu \geq 0$,*

$$\Pr\left[|\overline{X} - p| \geq \mu\right] \leq 2 \cdot e^{-2m\mu^2}.$$

3 A Fixed-Parameter Algorithm for the Kneser Problem

In this section we present and analyze our randomized fixed-parameter algorithm for the Kneser problem. We start with a couple of lemmas on induced subgraphs of Kneser graphs that will play a central role in the analysis of the algorithm. We then describe an algorithm, called "element elimination", which forms a main ingredient in our algorithm for the Kneser problem, and then use it to present the final algorithm and to prove Theorem 1.

3.1 Induced Subgraphs of Kneser Graphs

The following lemma shows that in a large induced subgraph of $K(n,k)$ whose vertices do not have a popular element, a random pair of vertices forms an edge with a non-negligible probability. Its proof uses the Hilton-Milner theorem (Theorem 9) and can be found in the full version of the paper. Recall that for a family $\mathcal{F} \subseteq \binom{[n]}{k}$, we let $K(\mathcal{F})$ stand for the subgraph of $K(n,k)$ induced by $\mathcal{F}$ (see Definition 6).

▶ **Lemma 12.** *For integers $k \geq 3$ and $n \geq 2k$, let $\mathcal{F}$ be a family of k-subsets of $[n]$ of size $|\mathcal{F}| \geq k^2 \cdot \binom{n-2}{k-2}$ and let $\gamma \in (0,1]$. Suppose that every element of $[n]$ belongs to at most γ fraction of the sets of $\mathcal{F}$. Then, the probability that two random sets chosen uniformly and independently from $\mathcal{F}$ are adjacent in $K(\mathcal{F})$ is at least*

$$\frac{1}{2} \cdot \left(1 - \gamma - \frac{k}{|\mathcal{F}|} \cdot \binom{n-2}{k-2}\right) \cdot \left(1 - \frac{k^2}{|\mathcal{F}|} \cdot \binom{n-2}{k-2}\right).$$

As a corollary of Lemma 12, we obtain the following.

▶ **Corollary 13.** *For integers $k \geq 3$ and $n \geq 8k^4$, let $\mathcal{F}$ be a family of k-subsets of $[n]$ of size $|\mathcal{F}| \geq \frac{1}{2n} \cdot \binom{n}{k}$ and let $\gamma \in (0,1]$. Suppose that every element of $[n]$ belongs to at most γ fraction of the sets of $\mathcal{F}$. Then, the probability that two random sets chosen uniformly and independently from $\mathcal{F}$ are adjacent in $K(\mathcal{F})$ is at least $\frac{3}{8} \cdot (\frac{3}{4} - \gamma)$.*

Proof. Observe that the assumptions $|\mathcal{F}| \geq \frac{1}{2n} \cdot \binom{n}{k}$ and $n \geq 8k^4$ imply that

$$\frac{k^2}{|\mathcal{F}|} \cdot \binom{n-2}{k-2} \leq \frac{2nk^2}{\binom{n}{k}} \cdot \binom{n-2}{k-2} = \frac{2k^3(k-1)}{n-1} \leq \frac{1}{4}.$$

Applying Lemma 12, we obtain that the probability that two random sets chosen uniformly and independently from $\mathcal{F}$ are adjacent in $K(\mathcal{F})$ is at least $\frac{1}{2} \cdot (1 - \gamma - \frac{1}{4}) \cdot (1 - \frac{1}{4}) = \frac{3}{8} \cdot (\frac{3}{4} - \gamma)$. ◀

The following lemma shows that if a large collection of vertices of $K(n,k)$ has a quite popular element, then every vertex that does not include this element is adjacent to many of the vertices in the collection.

▶ **Lemma 14.** *For integers $k \geq 2$ and $n \geq 2k$, let $X \subseteq [n]$ be a set, let $\mathcal{F}$ be a family of k-subsets of X, and let $\gamma \in (0,1]$. Let $j \in X$ be an element that belongs to at least γ fraction of the sets of $\mathcal{F}$, and suppose that $A \in \binom{[n]}{k}$ is a set satisfying $j \notin A$. Then, the probability that a random set chosen uniformly from $\mathcal{F}$ is disjoint from A is at least*

$$\gamma - \frac{k}{|\mathcal{F}|} \cdot \binom{|X|-2}{k-2}.$$

Proof. Let $\mathcal{F} \subseteq \binom{X}{k}$ be a family as in the lemma, and put $\mathcal{F}' = \{F \in \mathcal{F} \mid j \in F\}$. By assumption, it holds that $|\mathcal{F}'| \geq \gamma \cdot |\mathcal{F}|$. Suppose that $A \in \binom{[n]}{k}$ is a set satisfying $j \notin A$. Observe that the number of sets $B \in \binom{X}{k}$ with $j \in B$ that intersect A does not exceed $|A \cap X| \cdot \binom{|X|-2}{k-2} \leq k \cdot \binom{|X|-2}{k-2}$. It thus follows that the number of sets of $\mathcal{F}$ that are disjoint from A is at least

$$|\mathcal{F}'| - k \cdot \binom{|X|-2}{k-2} \geq \gamma \cdot |\mathcal{F}| - k \cdot \binom{|X|-2}{k-2}.$$

Hence, a random set chosen uniformly from $\mathcal{F}$ is disjoint from A with the desired probability. ◀

As a corollary of Lemma 14, we obtain the following.

▶ **Corollary 15.** *For integers $k \geq 2$ and n, let $X \subseteq [n]$ be a set of size $|X| \geq 8k^3$, let $\mathcal{F}$ be a family of k-subsets of X of size $|\mathcal{F}| \geq \frac{1}{2|X|} \cdot \binom{|X|}{k}$, and let $\gamma \in (0,1]$. Let $j \in X$ be an element that belongs to at least γ fraction of the sets of $\mathcal{F}$, and suppose that $A \in \binom{[n]}{k}$ is a set satisfying $j \notin A$. Then, the probability that a random set chosen uniformly from $\mathcal{F}$ is disjoint from A is at least $\gamma - \frac{1}{4}$.*

Proof. Observe that the assumptions $|\mathcal{F}| \geq \frac{1}{2|X|} \cdot \binom{|X|}{k}$ and $|X| \geq 8k^3$ imply that

$$\frac{k}{|\mathcal{F}|} \cdot \binom{|X|-2}{k-2} \leq \frac{2|X|k}{\binom{|X|}{k}} \cdot \binom{|X|-2}{k-2} = \frac{2k^2(k-1)}{|X|-1} \leq \frac{1}{4}.$$

Applying Lemma 14, we obtain that the probability that a random set chosen uniformly from $\mathcal{F}$ is disjoint from A is at least $\gamma - \frac{1}{4}$. ◀

3.2 The Element Elimination Algorithm

A main ingredient in our fixed-parameter algorithm for the Kneser problem is the "element elimination" algorithm given by the following theorem. It will be used to repeatedly reduce the size of the ground set of a Kneser graph while looking for a monochromatic edge.

▶ **Theorem 16.** *There exists a randomized algorithm that given integers n and k, a set $X \subseteq [n]$ of size $|X| \geq 8k^4$, a set of colors $C \subseteq [n-2k+1]$ of size $|C| = |X| - 2k + 1$, and an oracle access to a coloring $c : \binom{X}{k} \to [n-2k+1]$ of the vertices of $K(\binom{X}{k})$, runs in time polynomial in n and returns, with probability $1 - 2^{-\Omega(n)}$,*

(a) *a monochromatic edge of $K(\binom{X}{k})$, or*

(b) *a vertex $A \in \binom{X}{k}$ satisfying $c(A) \notin C$, or*

(c) *a color $i \in C$ and an element $j \in X$ such that for every $A \in \binom{[n]}{k}$ with $j \notin A$, a random vertex B chosen uniformly from $\binom{X}{k}$ satisfies $c(B) = i$ and $A \cap B = \emptyset$ with probability at least $\frac{1}{16n}$.*

Proof. For integers n and k, let $X \subseteq [n]$, $C \subseteq [n-2k+1]$, and $c : \binom{X}{k} \to [n-2k+1]$ be an input satisfying $|X| \geq 8k^4$ and $|C| = |X| - 2k + 1$ as in the statement of the theorem. It can be assumed that $k \geq 3$. Indeed, Theorem 7 guarantees that $K(\binom{X}{k})$ has either a monochromatic edge or a vertex whose color does not belong to C, hence for $k \leq 2$, an output of type (a) or (b) can be found by querying the oracle for the colors of all the vertices in time polynomial in n. For $k \geq 3$, consider the algorithm that given an input as above acts as follows (see Algorithm 1).

The algorithm first selects uniformly and independently m random sets $A_1, \ldots, A_m \in \binom{X}{k}$ for $m = n^3$ (see lines 1–2) and queries the oracle for their colors. If the sampled sets include two vertices that form a monochromatic edge, then the algorithm returns such an edge (output of type (a); see line 5). If they include a vertex whose color does not belong to C, then the algorithm returns it (output of type (b); see line 10). Otherwise, the algorithm defines $i^* \in C$ as a color that appears on a largest number of sampled sets A_t (see lines 13–16). It further defines $j^* \in X$ as an element that belongs to a largest number of sampled sets A_t with $c(A_t) = i^*$ (see lines 17–20). Then, the algorithm returns the pair (i^*, j^*) (output of type (c); see line 21).

It is clear that the algorithm runs in time polynomial in n. We turn to prove that for every input, the algorithm returns a valid output, of type (a), (b), or (c), with probability $1 - 2^{-\Omega(n)}$. We start with the following lemma that shows that if the input coloring has a large color class with no popular element, then with high probability the algorithm returns a valid output of type (a).

▶ **Lemma 17.** *Suppose that the input coloring c has a color class $\mathcal{F} \subseteq \binom{X}{k}$ of size $|\mathcal{F}| \geq \frac{1}{2|X|} \cdot \binom{|X|}{k}$ such that every element of X belongs to at most half of the sets of $\mathcal{F}$. Then, Algorithm 1 returns a monochromatic edge with probability $1 - 2^{-\Omega(n)}$.*

Algorithm 1 Element Elimination Algorithm (Theorem 16).

Input: Integers n and $k \geq 3$, a set $X \subseteq [n]$ of size $|X| \geq 8k^4$, a set of colors $C \subseteq [n-2k+1]$ of size $|C| = |X|-2k+1$, and an oracle access to a coloring $c : \binom{X}{k} \to [n-2k+1]$ of $K(\binom{X}{k})$.
Output: (a) A monochromatic edge of $K(\binom{X}{k})$, or (b) a vertex $A \in \binom{X}{k}$ satisfying $c(A) \notin C$, or (c) a color $i \in C$ and an element $j \in X$ such that for every $A \in \binom{[n]}{k}$ with $j \notin A$, a random vertex B chosen uniformly from $\binom{X}{k}$ satisfies $c(B) = i$ and $A \cap B = \emptyset$ with probability at least $\frac{1}{16n}$.

```
 1: $m \leftarrow n^3$
 2: pick uniformly and independently at random sets $A_1, \ldots, A_m \in \binom{X}{k}$
 3: for all $t, t' \in [m]$ do
 4:     if $c(A_t) = c(A_{t'})$ and $A_t \cap A_{t'} = \emptyset$ then
 5:         return $\{A_t, A_{t'}\}$                          ▷ output of type (a)
 6:     end if
 7: end for
 8: for all $t \in [m]$ do
 9:     if $c(A_t) \notin C$ then
10:         return $A_t$                                      ▷ output of type (b)
11:     end if
12: end for
13: for all $i \in C$ do
14:     $\widetilde{\alpha}_i \leftarrow \frac{1}{m} \cdot |\{t \in [m] \mid c(A_t) = i\}|$
15: end for
16: $i^* \leftarrow$ an $i \in C$ with largest value of $\widetilde{\alpha}_i$
17: for all $j \in X$ do
18:     $\widetilde{\gamma}_{i^*,j} \leftarrow \frac{1}{m} \cdot |\{t \in [m] \mid c(A_t) = i^*$ and $j \in A_t\}|$
19: end for
20: $j^* \leftarrow$ a $j \in X$ with largest value of $\widetilde{\gamma}_{i^*,j}$
21: return $(i^*, j^*)$                                      ▷ output of type (c)
```

Proof. Let $\mathcal{F}$ be as in the lemma. Applying Corollary 13 with $\gamma = \frac{1}{2}$, using the assumptions $k \geq 3$ and $|X| \geq 8k^4$, we obtain that two random sets chosen uniformly and independently from $\mathcal{F}$ are adjacent in $K(\mathcal{F})$ with probability at least $\frac{3}{8} \cdot (\frac{3}{4} - \gamma) = \frac{3}{32}$. Further, the fact that $|\mathcal{F}| \geq \frac{1}{2|X|} \cdot \binom{|X|}{k}$ implies that a random vertex chosen uniformly from $\binom{X}{k}$ belongs to $\mathcal{F}$ with probability at least $\frac{1}{2|X|}$. Hence, for two random vertices chosen uniformly and independently from $\binom{X}{k}$, the probability that they both belong to $\mathcal{F}$ is at least $(\frac{1}{2|X|})^2$, and conditioned on this event, their probability to form an edge in $K(\mathcal{F})$ is at least $\frac{3}{32}$. This implies that the probability that two random vertices chosen uniformly and independently from $\binom{X}{k}$ form a monochromatic edge in $K(\binom{X}{k})$ is at least $(\frac{1}{2|X|})^2 \cdot \frac{3}{32} = \frac{3}{128|X|^2}$.

Now, by considering $\lfloor m/2 \rfloor$ pairs of the random sets chosen by Algorithm 1 (line 2), it follows that the probability that no pair forms a monochromatic edge does not exceed

$$\left(1 - \frac{3}{128|X|^2}\right)^{\lfloor m/2 \rfloor} \leq e^{-3 \cdot \lfloor m/2 \rfloor/(128|X|^2)} \leq 2^{-\Omega(n)},$$

where the last inequality follows by $|X| \leq n$ and $m = n^3$. It thus follows that with probability $1 - 2^{-\Omega(n)}$, the algorithm returns a monochromatic edge, as required. ◀

We next handle the case in which every large color class of the input coloring has a popular element. To do so, we first show that the samples of the algorithm provide a good estimation for the fraction of vertices in each color class as well as for the fraction of vertices that share any given element in each color class. For every color $i \in C$, let α_i denote the fraction of vertices of $K(\binom{X}{k})$ colored i, that is,

$$\alpha_i = \frac{|\{A \in \binom{X}{k} \mid c(A) = i\}|}{\binom{|X|}{k}},$$

and let $\widetilde{\alpha}_i$ denote the fraction of the vertices sampled by the algorithm that are colored i (see line 14). Similarly, for every $i \in C$ and $j \in X$, let $\gamma_{i,j}$ denote the fraction of vertices of $K(\binom{X}{k})$ colored i that include j, that is,

$$\gamma_{i,j} = \frac{|\{A \in \binom{X}{k} \mid c(A) = i \text{ and } j \in A\}|}{\binom{|X|}{k}},$$

and let $\widetilde{\gamma}_{i,j}$ denote the fraction of the vertices sampled by the algorithm that are colored i and include j. Let E denote the event that

$$|\alpha_i - \widetilde{\alpha}_i| \leq \frac{1}{2|X|} \quad \text{and} \quad |\gamma_{i,j} - \widetilde{\gamma}_{i,j}| \leq \frac{1}{2|X|} \quad \text{for all } i \in C,\ j \in X. \tag{2}$$

By a standard concentration argument, we obtain the following lemma.

▶ **Lemma 18.** *The probability of the event E is $1 - 2^{-\Omega(n)}$.*

Proof. By the Chernoff-Hoeffding bound (Theorem 11) applied with $\mu = \frac{1}{2|X|}$, the probability that an inequality from (2) does not hold is at most

$$2 \cdot e^{-2m/(4|X|^2)} \leq 2 \cdot e^{-n/2},$$

where the inequality follows by $|X| \leq n$ and $m = n^3$. By the union bound over all the colors $i \in C$ and all the pairs $(i, j) \in C \times X$, that is, over $|C| \cdot (1 + |X|) \leq n^2$ events, we get that all the inequalities in (2) hold with probability at least $1 - 2n^2 \cdot e^{-n/2} = 1 - 2^{-\Omega(n)}$, as required. ◀

We now show that if every large color class of the input coloring has a popular element and the event E occurs, then the algorithm returns a valid output.

▶ **Lemma 19.** *Suppose that the coloring c satisfies that for every color class $\mathcal{F} \subseteq \binom{X}{k}$ of size $|\mathcal{F}| \geq \frac{1}{2|X|} \cdot \binom{|X|}{k}$ there exists an element of X that belongs to more than half of the sets of $\mathcal{F}$. Then, if the event E occurs, Algorithm 1 returns a valid output.*

Proof. Assume that the event E occurs. If Algorithm 1 returns an output of type (a) or (b), i.e., a monochromatic edge or a vertex whose color does not belong to C, then the output is verified before it is returned and is thus valid. So suppose that the algorithm returns a pair $(i^*, j^*) \in C \times X$. Recall that the color i^* is defined by Algorithm 1 as an $i \in C$ with largest value of $\widetilde{\alpha}_i$ (see line 16). Since the colors of all the sampled sets belong to C, it follows that $\sum_{i \in C} \widetilde{\alpha}_i = 1$, and thus

$$\widetilde{\alpha}_{i^*} \geq \frac{1}{|C|} \geq \frac{1}{|X|}, \tag{3}$$

where the last inequality follows by $|C| = |X| - 2k + 1 \leq |X|$.

Let $\mathcal{F}$ be the family of vertices of $K(\binom{X}{k})$ colored i^*, i.e., $\mathcal{F} = \{A \in \binom{X}{k} \mid c(A) = i^*\}$. Since the event E occurs (see (2)), it follows from (3) that

$$|\mathcal{F}| = \alpha_{i^*} \cdot \binom{|X|}{k} \geq \Big(\widetilde{\alpha}_{i^*} - \frac{1}{2|X|}\Big) \cdot \binom{|X|}{k} \geq \frac{1}{2|X|} \cdot \binom{|X|}{k}.$$

Hence, by the assumption of the lemma, there exists an element $j \in X$ that belongs to more than half of the sets of $\mathcal{F}$, that is, $\gamma_{i^*,j} > \frac{1}{2}$. Since the event E occurs, it follows that this j satisfies $\widetilde{\gamma}_{i^*,j} > \frac{1}{2} - \frac{1}{2|X|}$. Recalling that the element j^* is defined by Algorithm 1 as a $j \in X$ with largest value of $\widetilde{\gamma}_{i^*,j}$ (see line 20), it must satisfy $\widetilde{\gamma}_{i^*,j^*} > \frac{1}{2} - \frac{1}{2|X|}$, and using again the fact that the event E occurs, we derive that $\gamma_{i^*,j^*} \geq \widetilde{\gamma}_{i^*,j^*} - \frac{1}{2|X|} > \frac{1}{2} - \frac{1}{|X|} \geq \frac{3}{8}$ (using $|X| \geq 8$).

By $k \geq 3$ and $|X| \geq 8k^4$, we can apply Corollary 15 with $\mathcal{F}$, j^*, and $\gamma = \frac{3}{8}$ to obtain that for every set $A \in \binom{[n]}{k}$ with $j^* \notin A$, the probability that a random set chosen uniformly from $\mathcal{F}$ is disjoint from A is at least $\gamma - \frac{1}{4} = \frac{1}{8}$. Since the probability that a random set chosen uniformly from $\binom{X}{k}$ belongs to $\mathcal{F}$ is at least $\frac{1}{2|X|}$, it follows that the probability that a random set B chosen uniformly from $\binom{X}{k}$ satisfies $c(B) = i^*$ and $A \cap B = \emptyset$ is at least $\frac{1}{2|X|} \cdot \frac{1}{8} = \frac{1}{16|X|} \geq \frac{1}{16n}$. This implies that (i^*, j^*) is a valid output of type (c), as required. ◀

Equipped with Lemmas 17, 18, and 19, we are ready to derive the correctness of Algorithm 1 and to complete the proof of Theorem 16. If the input coloring c has a color class $\mathcal{F} \subseteq \binom{X}{k}$ of size $|\mathcal{F}| \geq \frac{1}{2|X|} \cdot \binom{|X|}{k}$ such that every element of X belongs to at most half of the sets of $\mathcal{F}$, then, by Lemma 17, the algorithm returns with probability $1 - 2^{-\Omega(n)}$ a monochromatic edge, i.e., a valid output of type (a). Otherwise, the input coloring c satisfies that for every color class $\mathcal{F} \subseteq \binom{X}{k}$ of size $|\mathcal{F}| \geq \frac{1}{2|X|} \cdot \binom{|X|}{k}$ there exists an element of X that belongs to more than half of the sets of $\mathcal{F}$. By Lemma 18, the event E occurs with probability $1 - 2^{-\Omega(n)}$, implying by Lemma 19 that with such probability, the algorithm returns a valid output. It thus follows that for every input coloring the algorithm returns a valid output with probability $1 - 2^{-\Omega(n)}$, and we are done. ◀

3.3 The Fixed-Parameter Algorithm for the Kneser Problem

We turn to present our fixed-parameter algorithm for the Kneser problem and to complete the proof of Theorem 1.

Proof of Theorem 1. Suppose that we are given, for integers $n \geq 2k$, an oracle access to a coloring $c : \binom{[n]}{k} \to [n-2k+1]$ of the vertices of the Kneser graph $K(n,k)$. Our algorithm has two phases, as described below (see Algorithm 2).

In the first phase, the algorithm repeatedly applies the "element elimination" algorithm given in Theorem 16 (Algorithm 1). Initially, we define

$$s = \max(n - 8k^4, 0), \quad X_0 = [n], \quad \text{and} \quad C_0 = [n-2k+1].$$

In the lth iteration, $0 \leq l < s$, we call Algorithm 1 with n, k, X_l, C_l, and with the restriction of the given coloring c to the vertices of $\binom{X_l}{k}$ to obtain with probability $1 - 2^{-\Omega(n)}$,

(a) a monochromatic edge $\{A, B\}$ of $K(\binom{X_l}{k})$, or

(b) a vertex $A \in \binom{X_l}{k}$ satisfying $c(A) \notin C_l$, or

(c) a color $i_l \in C_l$ and an element $j_l \in X_l$ such that for every $A \in \binom{[n]}{k}$ with $j_l \notin A$, a random vertex B chosen uniformly from $\binom{X_l}{k}$ satisfies $c(B) = i_l$ and $A \cap B = \emptyset$ with probability at least $\frac{1}{16n}$.

Algorithm 2 The Algorithm for the Kneser Problem (Theorem 1).

Input: Integers n, k with $n \geq 2k$ and an oracle access to a coloring $c : \binom{[n]}{k} \to [n-2k+1]$ of $K(n,k)$.
Output: A monochromatic edge of $K(n,k)$.

```
1:  $s \leftarrow \max(n - 8k^4, 0)$,  $X_0 \leftarrow [n]$,  $C_0 \leftarrow [n-2k+1]$        ▷ $|C_0| = |X_0| - 2k + 1$
2:  for all $l = 0, 1, \dots, s-1$ do                                             ▷ first phase
3:      call Algorithm 1 with $n$, $k$, $X_l$, $C_l$, and with the restriction of $c$ to $\binom{X_l}{k}$
4:      if Algorithm 1 returns an edge $\{A, B\}$ with $c(A) = c(B)$ then   ▷ output of type (a)
5:          return $\{A, B\}$
6:      end if
7:      if Algorithm 1 returns a vertex $A \in \binom{X_l}{k}$ with $c(A) = i_r \notin C_l$ then   ▷ output of type (b)
8:          for all $t \in [n^2]$ do
9:              pick uniformly at random a set $B_t \in \binom{X_r}{k}$
10:             if $c(B_t) = i_r$ and $A \cap B_t = \emptyset$ then
11:                 return $\{A, B_t\}$
12:             end if
13:         end for
14:         return “failure”
15:     end if
16:     if Algorithm 1 returns a pair $(i_l, j_l) \in C_l \times X_l$ then            ▷ output of type (c)
17:         $X_{l+1} \leftarrow X_l \setminus \{j_l\}$,  $C_{l+1} \leftarrow C_l \setminus \{i_l\}$          ▷ $|C_{l+1}| = |X_{l+1}| - 2k + 1$
18:     end if
19: end for
20: query the oracle for the colors of all the vertices of $K(\binom{X_s}{k})$             ▷ second phase
21: if there exists a vertex $A \in \binom{X_s}{k}$ of color $c(A) = i_r \notin C_s$ then
22:     for all $t \in [n^2]$ do
23:         pick uniformly at random a set $B_t \in \binom{X_r}{k}$
24:         if $c(B_t) = i_r$ and $A \cap B_t = \emptyset$ then
25:             return $\{A, B_t\}$
26:         end if
27:     end for
28:     return “failure”
29: else
30:     find $A, B \in \binom{X_s}{k}$ satisfying $c(A) = c(B)$ and $A \cap B = \emptyset$     ▷ exist by Theorem 7 [19]
31:     return $\{A, B\}$
32: end if
```

As will be explained shortly, if the output of Algorithm 1 is of type (a) or (b) then we either return a monochromatic edge or declare “failure”, and if the output is a pair (i_l, j_l) of type (c) then we define $X_{l+1} = X_l \setminus \{j_l\}$ and $C_{l+1} = C_l \setminus \{i_l\}$ and, as long as $l < s$, proceed to the next call of Algorithm 1. Note that the sizes of the sets X_l and C_l are reduced by 1 in every iteration, hence we maintain the equality $|C_l| = |X_l| - 2k + 1$ for all l. We now describe how the algorithm acts in the lth iteration for each type of output returned by Algorithm 1.

If the output is of type (a), then the returned monochromatic edge of $K(\binom{X_l}{k})$ is also a monochromatic edge of $K(n,k)$, so we return it (see lines 4–6).

If the output is of type (b), then we are given a vertex $A \in \binom{X_l}{k}$ satisfying $c(A) = i_r \notin C_l$ for some $r < l$. Since $i_r \notin C_l$, it follows that $j_r \notin X_l$, and thus $j_r \notin A$. In this case, we pick uniformly and independently n^2 random sets from $\binom{X_r}{k}$ and query the oracle for their colors. If we find a vertex B that forms together with A a monochromatic edge in $K(n, k)$, we return the monochromatic edge $\{A, B\}$, and otherwise we declare "failure" (see lines 7–15).

If the output of Algorithm 1 is a pair (i_l, j_l) of type (c), then we define, as mentioned above, the sets $X_{l+1} = X_l \setminus \{j_l\}$ and $C_{l+1} = C_l \setminus \{i_l\}$ (see lines 16–18). Observe that for $0 \leq l < s$, it holds that $|X_l| = n - l > n - s = 8k^4$, allowing us, by Theorem 16, to call Algorithm 1 in the lth iteration.

In case that all the s calls to Algorithm 1 return an output of type (c), we arrive to the second phase of the algorithm. Here, we are given the sets X_s and C_s that satisfy $|X_s| = n - s \leq 8k^4$ and $|C_s| = |X_s| - 2k + 1$, and we query the oracle for the colors of each and every vertex of the graph $K(\binom{X_s}{k})$. If we find a vertex $A \in \binom{X_s}{k}$ satisfying $c(A) = i_r \notin C_s$ for some $r < s$, then, as before, we pick uniformly and independently n^2 random sets from $\binom{X_r}{k}$ and query the oracle for their colors. If we find a vertex B that forms together with A a monochromatic edge in $K(n, k)$, we return the monochromatic edge $\{A, B\}$, and otherwise we declare "failure" (see lines 21–28). Otherwise, all the vertices of $K(\binom{X_s}{k})$ are colored by colors from C_s. By Theorem 7, the chromatic number of $K(\binom{X_s}{k})$ is $|X_s| - 2k + 2 > |C_s|$. Hence, there must exist a monochromatic edge in $K(\binom{X_s}{k})$, and by checking all the pairs of its vertices we find such an edge and return it (see lines 30–31).

We turn to analyze the probability that Algorithm 2 returns a monochromatic edge. Note that whenever the algorithm returns an edge, it checks that it is monochromatic and thus ensures that it forms a valid solution. Hence, it suffices to show that the algorithm declares "failure" with probability $2^{-\Omega(n)}$. To see this, recall that the algorithm calls Algorithm 1 at most $s < n$ times, and that by Theorem 16 the probability that its output is not valid is $2^{-\Omega(n)}$. By the union bound, the probability that any of the calls to Algorithm 1 returns an invalid output is $2^{-\Omega(n)}$ too. The only situation in which Algorithm 2 declares "failure" is when it finds, for some $r < s$, a vertex $A \in \binom{[n]}{k}$ with $c(A) = i_r$ and $j_r \notin A$, and none of the n^2 sampled sets $B \in \binom{X_r}{k}$ satisfies $c(B) = i_r$ and $A \cap B = \emptyset$ (see lines 7–15, 21–28). However, assuming that all the calls to Algorithm 1 return valid outputs, the rth run guarantees, by Theorem 16, that a random vertex B uniformly chosen from $\binom{X_r}{k}$ satisfies $c(B) = i_r$ and $A \cap B = \emptyset$ for the given A with probability at least $\frac{1}{16n}$. Hence, the probability that the algorithm declares "failure" does not exceed $(1 - \frac{1}{16n})^{n^2} \leq e^{-n/16} = 2^{-\Omega(n)}$. Using again the union bound, it follows that the probability that Algorithm 2 either gets an invalid output from Algorithm 1 or fails to find a vertex that forms a monochromatic edge with a set A as above is $2^{-\Omega(n)}$. Therefore, the probability that Algorithm 2 successfully finds a monochromatic edge is $1 - 2^{-\Omega(n)}$, as desired.

We finally analyze the running time of Algorithm 2. In its first phase, the algorithm calls Algorithm 1 at most $s < n$ times, where the running time needed for each call is, by Theorem 16, polynomial in n. It is clear that the other operations made throughout this phase can also be implemented in time polynomial in n. In its second phase, the algorithm enumerates all the vertices of $K(\binom{X_s}{k})$. This phase can be implemented in running time polynomial in n and in the number of vertices of this graph. The latter is $\binom{|X_s|}{k} \leq |X_s|^k \leq (8k^4)^k = k^{O(k)}$. It thus follows that the total running time of Algorithm 2 is $n^{O(1)} \cdot k^{O(k)}$, completing the proof. ◀

3.4 Turing Kernelization for the Kneser Problem

As mentioned in the introduction, our algorithm for the KNESER problem can be viewed as a randomized polynomial Turing kernelization algorithm for the problem. In what follows we extend on this aspect of the algorithm.

We start with the definition of a Turing kernelization algorithm as it is used in the standard context of parameterized complexity (see, e.g., [12, Chapter 22]). A decision parameterized problem P is a language of pairs (x,k) where k is an integer referred to as the parameter of P. For a decision parameterized problem P and a computable function f, a Turing kernelization algorithm of size f for P is an algorithm that decides whether an input (x,k) belongs to P in polynomial time given an oracle access that decides membership in P for instances (x',k') with $|x'| \leq f(k)$ and $k' \leq f(k)$. The kernelization algorithm is said to be polynomial if f is a polynomial function.

The Kneser problem, however, is a total search problem whose input is given as an oracle access. Hence, for an instance of the Kneser problem associated with integers n and k, we require a Turing kernelization algorithm to find a solution using an oracle that finds solutions for instances associated with integers n' and k' that are bounded by a function of k. We further require the algorithm to be able to simulate the queries of the produced instances using queries to the oracle associated with the original input. Note that we have here two different types of oracles: the oracle that solves the Kneser problem on graphs $K(n',k')$ with bounded n',k' and the oracle that supplies an access to the instance of the Kneser problem.

We claim that the proof of Theorem 1 shows that the Kneser problem admits a randomized polynomial Turing kernelization algorithm in the following manner. Given an instance of the Kneser problem, a coloring $c : \binom{[n]}{k} \rightarrow [n-2k+1]$ of the vertices of a Kneser graph $K(n,k)$ for integers n and k with $n \geq 2k$, the first phase of Algorithm 2 runs in time polynomial in n and either finds a monochromatic edge or produces a ground set $X_s \subseteq [n]$ and a set of colors $C_s \subseteq [n-2k+1]$ satisfying $|C_s| = |X_s| - 2k + 1$ and $|X_s| = O(k^4)$ (see lines 2–19). In the latter case, the restriction of the input coloring c to the vertices of the graph $K(\binom{X_s}{k})$ is not guaranteed to use only colors from C_s, as required by the definition of the Kneser problem. Yet, by applying the oracle to this restriction of c, simulating its queries using the access to the coloring c of $K(n,k)$, we either get a monochromatic edge in $K(\binom{X_s}{k})$ or find a vertex whose color does not belong to C_s. In both cases, a solution to the original instance can be efficiently found with high probability. Indeed, if a monochromatic edge is returned then it forms a monochromatic edge of $K(n,k)$ as well. Otherwise, as shown in the analysis of Algorithm 2, a vertex $A \in \binom{X_s}{k}$ whose color does not belong to C_s can be used to efficiently find with high probability a vertex $B \in \binom{[n]}{k}$ such that $\{A,B\}$ is a monochromatic edge (see lines 21–28).

4 An Algorithm for the Kneser Problem Based on Schrijver Graphs

In this section we present the simple deterministic algorithm for the Kneser problem given in Theorem 2. Let us first state a simple fact on the number of vertices in Schrijver graphs (see Section 2.2). We include a quick proof for completeness.

▶ **Fact 20.** *For integers $k \geq 2$ and $n \geq 2k$, the number of vertices in $S(n,k)$ is*

$$\binom{n-k+1}{k} - \binom{n-k-1}{k-2}.$$

Proof. Recall that the vertex set $\binom{[n]}{k}_{\text{stab}}$ of $S(n,k)$ is the collection of all k-subsets of $[n]$ with no two consecutive elements modulo n. We first observe that the number of k-subsets of $[n]$ with no two consecutive elements, allowing both 1 and n to be in the subsets, is $\binom{n-k+1}{k}$. To see this, identify the subsets of $[n]$ with their characteristic vectors in $\{0,1\}^n$, and in every such vector, interpret the zeros as balls and the ones as separations between bins. It follows that every such set corresponds to a partition of $n-k$ identical balls into $k+1$ bins, where no bin but the first and last is empty. The number of those partitions is equal to the number of partitions of $n-2k+1$ identical balls into $k+1$ bins, which is $\binom{n-k+1}{k}$. Finally, to obtain the number of vertices of $S(n,k)$, one has to subtract the number of k-subsets of $[n]$ with no two consecutive elements that include both 1 and n. The latter is equal to the number of $(k-2)$-subsets of $[n-4]$ with no two consecutive elements, which by the above argument equals $\binom{n-k-1}{k-2}$, so we are done. ◀

Equipped with Fact 20, we are ready to prove Theorem 2.

Proof of Theorem 2. Consider the algorithm that given integers n and k with $n \geq 2k$ and an oracle access to a coloring $c : \binom{[n]}{k} \to [n-2k+1]$ of the vertices of the Kneser graph $K(n,k)$, enumerates all vertices of the Schrijver graph $S(n,k)$, i.e., the sets of $\binom{[n]}{k}_{\text{stab}}$, and queries the oracle for their colors. Then, the algorithm goes over all pairs of those vertices and returns a pair that forms a monochromatic edge. The existence of such an edge follows from Theorem 7, which asserts that $\chi(S(n,k)) = n-2k+2$. The running time of the algorithm is polynomial in the number of vertices of $S(n,k)$, which by Fact 20 does not exceed $\binom{n-k+1}{k} = \binom{n-k+1}{n-2k+1} \leq n^{\min(k,n-2k+1)}$. ◀

References

1 Paul Beame, Stephen A. Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. *J. Comput. Syst. Sci.*, 57(1):3–19, 1998. Preliminary version in STOC'95.

2 Karol Borsuk. Drei Sätze über die n-dimensionale euklidische Sphäre. *Fundamenta Mathematicae*, 20(1):177–190, 1933.

3 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

4 Argyrios Deligkas, John Fearnley, Alexandros Hollender, and Themistoklis Melissourgos. Constant inapproximability for PPA. In *Proc. of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC'22)*, 2022.

5 Argyrios Deligkas, Aris Filos-Ratsikas, and Alexandros Hollender. Two's company, three's a crowd: Consensus-halving for a constant number of agents. In *Proc. of the 22nd ACM Conference on Economics and Computation (EC'21)*, pages 347–368, 2021.

6 Xiaotie Deng, Zhe Feng, and Rucha Kulkarni. Octahedral Tucker is PPA-complete. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:118, 2017.

7 Irit Dinur and Ehud Friedgut. Intersecting families are essentially contained in juntas. *Comb. Probab. Comput.*, 18(1–2):107–122, 2009.

8 Paul Erdös, Chao Ko, and Richard Rado. Intersection theorems for systems of finite sets. *Quart. J. Math.*, 12(1):313–320, 1961.

9 Aris Filos-Ratsikas and Paul W. Goldberg. Consensus halving is PPA-complete. In *Proc. of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC'18)*, pages 51–64, 2018.

10 Aris Filos-Ratsikas and Paul W. Goldberg. The complexity of splitting necklaces and bisecting ham sandwiches. In *Proc. of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC'19)*, pages 638–649, 2019.

11 Aris Filos-Ratsikas, Alexandros Hollender, Katerina Sotiraki, and Manolis Zampetakis. A topological characterization of modulo-p arguments and implications for necklace splitting. In *Proc. of the 32nd ACM-SIAM Symposium on Discrete Algorithms (SODA'21)*, pages 2615–2634, 2021.

12 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press, 2019.

13 Peter Frankl and Andrey Kupavskii. Maximal degrees in subgraphs of Kneser graphs. *arXiv*, abs/2004.08718, 2020. arXiv:2004.08718.

14 Paul W. Goldberg, Alexandros Hollender, Ayumi Igarashi, Pasin Manurangsi, and Warut Suksompong. Consensus halving for sets of items. In *Proc. 16th Web and Internet Economics International Conference (WINE'20)*, pages 384–397, 2020.

15 Ishay Haviv. The complexity of finding fair independent sets in cycles. In *12th Innovations in Theoretical Computer Science Conference (ITCS'21)*, pages 4:1–4:14, 2021.

16 Anthony J. W. Hilton and Eric Charles Milner. Some intersection theorems for systems of finite sets. *Quart. J. Math.*, 18(1):369–384, 1967.

17 Martin Kneser. Aufgabe 360. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 58(2):27, 1955.

18 Andrey Kupavskii. Diversity of uniform intersecting families. *Eur. J. Comb.*, 74:39–47, 2018.

19 László Lovász. Kneser's conjecture, chromatic number, and homotopy. *J. Comb. Theory, Ser. A*, 25(3):319–324, 1978.

20 Pasin Manurangsi and Warut Suksompong. Computing a small agreeable set of indivisible items. *Artif. Intell.*, 268:96–114, 2019. Preliminary versions in IJCAI'16 and IJCAI'17.

21 Jiří Matoušek. A combinatorial proof of Kneser's conjecture. *Combinatorica*, 24(1):163–170, 2004.

22 Jiří Matoušek. *Using the Borsuk-Ulam Theorem: Lectures on Topological Methods in Combinatorics and Geometry.* Springer Publishing Company, Incorporated, 2007.

23 Jiří Matoušek and Günter M. Ziegler. Topological lower bounds for the chromatic number: A hierarchy. *Jahresbericht der DMV*, 106(2):71–90, 2004.

24 Colin McDiarmid. Concentration. In *Probabilistic methods for algorithmic discrete mathematics*, volume 16 of *Algorithms Combin.*, pages 195–248. Springer, Berlin, 1998.

25 Nimrod Megiddo and Christos H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theor. Comput. Sci.*, 81(2):317–324, 1991.

26 Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 48(3):498–532, 1994.

27 Alexander Schrijver. Vertex-critical subgraphs of Kneser graphs. *Nieuw Arch. Wiskd.*, 26(3):454–461, 1978.

28 Forest W. Simmons and Francis Edward Su. Consensus-halving via theorems of Borsuk-Ulam and Tucker. *Math. Soc. Sci.*, 45(1):15–25, 2003.