Canonical forms for matrix tuples in polynomial time

Youming Qiao
Centre for Quantum Software and Information
University of Technology Sydney
Sydney, Australia
Youming.Qiao@uts.edu.au

Xiaorui Sun
Computer Science Department
University of Illinois at Chicago
Chicago, USA
xiaorui@uic.edu

2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS) | 979-8-3315-1674-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/FOCS61266.2024.00054

***Abstract*—Left-right and conjugation actions on matrix tuples have received considerable attention in theoretical computer science due to their connections with polynomial identity testing, group isomorphism, and tensor isomorphism. In this paper, we present polynomial-time algorithms for computing canonical forms of matrix tuples over a finite field under these actions. Our algorithm builds upon new structural insights for matrix tuples, which can be viewed as a generalization of Schur's lemma for irreducible representations to general representations.**

***Index Terms*—canonical form, matrix tuples, tensors, group isomorphism, computer algebra**

I. Introduction

Representing objects in a canonical and succinct way that can exhibit the underlying properties and structures of the objects is a fundamental problem in mathematics and computer science.

A classic example is the Jordan normal form for matrices in linear algebra. It not only transforms the matrices into a canonical form under the similarity relation[1], but it also demonstrates important structural information such as characteristic polynomials, algebraic and geometric eigenvalue multiplicities, the structure of generalized eigenvectors, and invariant subspace decompositions. The Jordan normal form is an important archetype in some mathematical areas. For example, it leads to Jordan–Chevalley decompositions, a useful tool in the study of linear algebraic groups [Bor91]. It also implies that the classification problem for matrices under the conjugation action is "tame" which basically means it is classifiable in the representation theory of finite-dimensional algebras [Rin97]. The Jordan normal form is also found useful in spectral graph theory [DGT17], [Her94].

The general canonical form problem aims to transform combinatorial and algebraic objects, such as graphs [Bab19], [BCS+13], [BL83], [SW15], [SW19], [WL68], tensors [NQT24], [Wei84], and groups [BEO02], into a canonical representation such that for equivalent inputs, the output representation is the same. The study of these canonical forms leads to the development of a wide range of structural theories [BCS+13], [SW15], [SW19], [WL68].

In this paper, we study the canonical representations of matrix tuples over finite fields. A matrix tuple is a sequence of matrices of the same size over the same finite field. We consider two actions for matrix tuples: the left-right action and the conjugation action. For a matrix tuple $(A_1, \ldots, A_\ell)$ of size $n \times m$, the *left-right action* by an $n \times n$ invertible matrix L and an $m \times m$ invertible matrix R transforms $(A_1, \ldots, A_\ell)$ into another matrix tuple $(LA_1R^{-1}, \ldots, LA_mR^{-1})$. For the *conjugation action*, it requires the matrix tuple to be square matrices, and the conjugation action by an invertible matrix L sends $(A_1, \ldots, A_\ell)$ to $(LA_1L^{-1}, \ldots, LA_\ell L^{-1})$. Two matrix tuples are *equivalent* if there exists a left-right action that transforms one matrix tuple into another, and two square matrix tuples are *conjugate* if there is a conjugation action that transforms one matrix tuple into another.

A *canonical form* algorithm for the matrix tuples in the equivalence case needs to satisfy the following conditions: given a matrix tuple $\mathbf{A} = (A_1, \ldots, A_\ell)$, the algorithm outputs $\mathbf{A}^* = (A_1^*, \ldots, A_\ell^*)$, such that $\mathbf{A}$ and $\mathbf{A}^*$ are equivalent, and for any matrix tuple $\mathbf{A}' = (A_1', \ldots, A_\ell')$ equivalent to $\mathbf{A}$, the algorithm outputs the same $\mathbf{A}^*$. In other words, $\mathbf{A}^*$ serves as a representative in the set of matrix tuples equivalent to $\mathbf{A}$. A canonical form algorithm in the matrix tuple conjugation case is defined in the same way by replacing "equivalent" with "conjugate" in the above.

The main result of this article is polynomial-time canonical form algorithms for matrix tuples under equiv-

Extended abstract for FOCS 2024. The full version of the paper is available at: https://arxiv.org/abs/2409.12457.

Youming Qiao is partly supported by Australian Research Council DP200100950 and LP220100332.

Xiaorui Sun is supported by the National Science Foundation (NSF) under Grant No. 2240024.

[1]Two $n \times n$ matrices A and B are similar if there exists an invertible matrix T such that $A = TBT^{-1}$. Then A and B are similar if and only if their Jordan normal forms are the same.

2575-8454/24/$31.00 ©2024 IEEE
DOI 10.1109/FOCS61266.2024.00054

alence or conjugation actions over finite fields. In the following, we shall introduce motivations for studying this problem and then describe our results in more detail.

A. Motivations

Matrix tuples, which encode systems of linear transformations or bilinear forms, have been studied in various scenarios. We motivate the study of matrix tuples and their canonical form from both theoretical computer science and mathematics perspectives.

1) Motivations from theoretical computer science:

Orbit closure intersection problems. Matrix tuples under the left-right action have received considerable attention in theoretical computer science [AZGL$^+$18], [DM20], [GGdOW20], [HH21], [IQ23], [IQS17], [IQS18].

One reason for interest in this action is the symbolic determinant identity testing (SDIT) problem. SDIT asks whether, for a given matrix tuple, the linear span of the matrices in this tuple contains a full-rank matrix. Derandomizing SDIT implies circuit lower bounds that seem beyond current techniques [KI04]. As the left-right action preserves matrix ranks, it is desirable to study the equivalence classes of matrix tuples under the left-right action, as pursued in several works mentioned above as well as in [Mul17], [MW21].

In particular, the orbit closure intersection problems for the left-right and conjugation actions have surprising connections to many areas of mathematics [GGdOW20], [IQS18]. By [DM17], they can be formulated as an instance of symbolic determinant identity testing. Recent advances [AZGL$^+$18], [DM20], [IQ23] provide deterministic polynomial-time algorithms for these orbit closure intersection problems.

Matrix tuples from group isomorphism. Testing the isomorphism of (finite) groups has been extensively studied since the 1970s. However, even after more than half a century, the best-known algorithm for group isomorphism remains a quasi-polynomial time algorithm dating back to the 1970s [FN70], [Mil78]. Improving the running time for group isomorphism is of interest in both computer science and mathematics, as evidenced by Gowers' question [Gow11], which led to Wilson's work [Wil19]. On the other hand, due to the recent breakthrough in graph isomorphism by Babai [Bab16], group isomorphism has become a major bottleneck in making any further progress on graph isomorphism [Bab16].

Very recently, Sun proposed an $n^{O((\log n)^{5/6})}$-time algorithm for testing isomorphism of p-groups of class 2 and exponent p [Sun23]. This result removes a major barrier for an $n^{o(\log n)}$-time algorithm for group isomorphism. In Sun's work, the problem is reduced to understanding three matrix tuples under different actions, two of which are left-right actions. Therefore, it is conceivable that understanding the structure of matrix tuples can shed new light on further improvements to group isomorphism.

Towards understanding tensor isomorphism and canonical form. Matrix tuples under equivalence actions also serve as an intermediate step to generalize our knowledge from matrices to tensors. Tensors have become increasingly important for computer science, not to mention their natural roles in statistics and quantum information.

Equivalence relations of tensors are natural generalizations of equivalence relations of matrices, such as the similarity relation discussed in the context of Jordan normal forms. One natural equivalence between tensors is as follows: Let $(A_1, \ldots, A_n)$ and $(B_1, \ldots, B_n)$ be two tuples of $n \times n$ matrices. We say that they are *isomorphic as tensors* if there exist $n \times n$ invertible matrices $L, R, T = (t_{i,j})$, such that for every $i \in [n]$, $LA_iR^{-1} = \sum_{j\in[n]} t_{i,j}B_j$. The tensor isomorphism problem then asks to decide whether two given matrix tuples are isomorphic as tensors, and the tensor canonical form problem asks to compute a canonical form of the input matrix tuple that is invariant under the isomorphism as tensors.

The tensor isomorphism problem has been studied in a series of works [CGQ$^+$24], [GQ23a], [GQ24], [GQT22], [GQ23b], with applications found in quantum information [CGQ$^+$24], [GQ23a]. Current evidence suggests that tensor isomorphism is a hard problem. For $n \times n \times n$ tensors over a finite field $\mathbb{F}_q$, the best algorithm with worst-case analysis runs in time $q^{\tilde{O}(n^{1.5})}$ [IMQ$^+$24], [Sun23], with average-case analysis in time $q^{O(n)}$ [BLQW20], [LQ17], and with heuristic analysis in time $q^{\frac{1}{2}n} \cdot \mathrm{poly}(n, \log q)$ [NQT24]. Indeed, because of these difficulties, digital signature schemes based on the assumed hardness of tensor isomorphism or equivalent problems have been proposed, including MEDS [CNP$^+$23a], [CNP$^+$23b] and ALTEQ [BDN$^+$23], [TDJ$^+$22], which are in submission to the NIST call for post-quantum digital signature schemes [Nat22]. Furthermore, recent research suggests that the tensor canonical form problem is also important to study in cryptography. For example, in [NQT24], canonical forms for tensors are used as an isomorphism invariant for birthday paradox-based algorithms.

We believe that computing canonical forms for matrix tuples is an important intermediate step for the isomorphism and canonical form problems of tensors because matrix tuple equivalence is a more restricted form of tensor isomorphism. Indeed, the best algorithms for tensor isomorphism [IMQ$^+$24], [NQT24], [Sun23] are obtained by partially fixing the matrices in one

direction of the tensor and then reducing the problem to certain equivalence problems for matrix tuples. Hence, we believe that understanding the structural and canonical form of matrix tuples is an important intermediate step toward understanding the structure and canonical form of tensors.

2) Motivations from mathematics: **Matrix tuples as a wild classification problem.** In the representation theory of associative algebras, classifying representations of quivers is a central topic [Rin97], dating back to the work of Gelfand and Ponomarev [GP69]. Roughly speaking, a classification problem is tame if it is classifiable (such as Jordan normal forms), and wild if it is not classifiable (defined as "containing" the problem of classifying pairs of matrices under simultaneous conjugation) (cf. [Ben98]). The celebrated tame-wild dichotomy was proved by Drozd [Dro79]. Classifying matrix tuples under equivalence (for no less than three matrices) or conjugation relations (for no less than two matrices) are well-known wild classification problems.

Isomorphism and canonical form algorithms. The wildness in classifying matrix tuples does not obstruct solutions to computational problems about them. Indeed, polynomial-time algorithms are known for testing whether two matrix tuples are conjugate or equivalent [BL08], [IKS10], [IQ19]. In particular, testing whether two matrix tuples are conjugate is a central problem in computer algebra [BW15], with practical algorithms implemented in computer algebra systems such as GAP [GAP17] and Magma [BJP97]. Interestingly, to the best of our knowledge, these algorithms have not led to a canonical form algorithm for matrix tuple conjugation or equivalence so far.

For canonical form problems, Belitskii and Sergeichuk presented algorithms for the canonical form problems for these actions over algebraically closed fields [Bel00], [BS03], [Ser00]. However, the complexity of the Belitskii–Sergeichuk algorithm seems missing in the literature[2], and it is unclear to us whether their algorithm extends to the finite field setting. Nevertheless, their algorithms indicate that non-trivial algorithms can be designed for the canonical form problem despite the wildness of the classification problem.

B. Main results

We now state our main result. Let $\mathrm{M}(n \times m, \mathbb{F}_q)$ denote the linear space of $n \times m$ matrices over $\mathbb{F}_q$ and $\mathrm{M}(n \times m, \mathbb{F}_q)^\ell$ denote the linear space of matrix tuples of length ℓ with each matrix in $\mathrm{M}(n \times m, \mathbb{F}_q)$.

[2]For example, there may be issues with the field extension degree required for the resulting canonical forms.

Theorem I.1. *There is a randomized Las-Vegas algorithm to compute a canonical form of a matrix tuple in* $\mathrm{M}(n \times m, \mathbb{F}_q)^\ell$ *under the equivalence relation in* $\mathrm{poly}(n, m, \ell, \log q)$ *time.*

Theorem I.1 relies on computing matrix algebra structures by Friedl, Ivanyos, and Rónyai [FR85], [Iva00], [Rón90]. Since the algorithms for computing matrix algebra structures over finite fields utilize polynomial factoring [Ber67], [CZ81], our algorithm for Theorem I.1 is a Las-Vegas randomized algorithm.

It is well-known that a canonical form algorithm for matrix tuple equivalence implies a canonical form algorithm for matrix tuple conjugation. Therefore, Theorem I.1 also provides a canonical form algorithm for matrix tuple conjugation.

Corollary I.2. *There is a randomized Las-Vegas algorithm to compute a canonical form of a matrix tuple in* $\mathrm{M}(n \times n, \mathbb{F}_q)^\ell$ *under the conjugation relation in* $\mathrm{poly}(n, \ell, \log q)$ *time.*

The key to Theorem I.1 is a structural result for matrix tuples. To state our structural result, we introduce some definitions. Similar to a block-diagonal matrix, a block diagonal matrix tuple is a matrix tuple where a sequence of matrix tuple blocks lies along the diagonal, and all other entries in each matrix are zero. A matrix tuple is decomposable if it is equivalent to a block-diagonal matrix tuple with at least two blocks; otherwise, it is indecomposable.

A row-submatrix tuple $\mathbf{B}$ of a matrix tuple $\mathbf{A}$ is a matrix tuple such that there exists a (not necessarily square) matrix L such that $\mathbf{B} = L\mathbf{A}$. A row-submatrix tuple $\mathbf{B}$ of $\mathbf{A}$ is an *indecomposable-block-corresponding row-submatrix tuple*, or *IBC-tuple* for short, if $\mathbf{A}$ is equivalent to a block-diagonal matrix tuple $\mathbf{D}$ such that $\mathbf{B}$ corresponds to an indecomposable block of $\mathbf{D}$. In other words, there are invertible matrices L and R such that $\mathbf{D} = \mathrm{diag}(\mathbf{D}_1, \ldots, \mathbf{D}_d) = L\mathbf{A}R^{-1}$ and $\mathbf{B}R^{-1} = (B_1R^{-1}, \ldots, B_\ell R^{-1})$ equals $\begin{bmatrix} 0 & \mathbf{D}_i & 0 \end{bmatrix}$ as a row-submatrix tuple of $\mathbf{D}$ for some block $\mathbf{D}_i$.

Theorem I.3. *For a matrix tuple* $\mathbf{A}$ *and an IBC-tuple* $\mathbf{B}$ *of* $\mathbf{A}$*, let* $V'_{\mathbf{B}}$ *denote the set of all the IBC-tuples* $\mathbf{B}'$ *right-equivalent to* $\mathbf{B}$*, i.e., there exists an invertible matrix* R *such that* $\mathbf{B}' = \mathbf{B}R^{-1}$*, and let* $V_{\mathbf{B}}$ *be the linear span of* $V'_{\mathbf{B}}$*. Then there exists a subspace* $K_{\mathbf{B}}$ *of* $V_{\mathbf{B}}$*, such that* $V'_{\mathbf{B}} = V_{\mathbf{B}} \setminus K_{\mathbf{B}}$.

To understand why Theorem I.3 is interesting, note that the structure of $V'_{\mathbf{B}}$, the set of IBC-tuples right-equivalent to $\mathbf{B}$, could be highly nonlinear. On the other hand, $V_{\mathbf{B}}$ is a linear space of row-submatrix tuples. Theorem I.3 then suggests that $V'_{\mathbf{B}}$ can be viewed as a quotient space of $V_{\mathbf{B}}$ over the subspace $K_{\mathbf{B}}$, so it is

close to being linear.

Theorem I.3 can be seen as a generalization of the fundamental Schur's lemma in representation theory. Roughly speaking, a matrix tuple $\mathbf{A}$ under the left-right action can be viewed as a representation of the so-called Kronecker quivers [Ben98]. Schur's lemma states that if $\mathbf{A}$ is not just indecomposable but also simple (also known as irreducible, a condition stronger than indecomposable, see [Ben98]), then the endomorphism algebra of $\mathbf{A}$ is a division algebra, which implies Theorem I.3 for indecomposable and simple matrix tuples by specifying $K_{\mathbf{A}}$ as the zero space.

Our result can be viewed as a generalization of Schur's lemma to general representations. We remark that such a generalization is nontrivial, even with the known characterization of endomorphism algebras of indecomposable modules as local algebras [Alp93, Section 2, Theorem 2]. In particular, Theorem I.3 for indecomposable subrepresentations in the general situation is affected by the interactions between non-equivalent indecomposable subrepresentations. Furthermore, unlike Schur's lemma, which is concerned with endomorphism algebras or homomorphisms, the subject in Theorem I.3 is indecomposable subrepresentations up to right equivalence, which seems not studied in the literature. Our main contribution is to discover and utilize this structural result in the process of devising canonical form algorithms.

II. Overview

In this section, we provide a high-level overview of the algorithm to compute canonical forms for matrix tuples under the left-right action in polynomial time.

For convenience, we assume that all the vectors are row vectors throughout the paper. For a matrix tuple $\mathbf{A} = (A_1, \ldots, A_\ell) \in \mathrm{M}(n \times m, \mathbb{F}_q)^\ell$, we define a tuple of row vectors $\mathbf{a} \in (\mathbb{F}_q^m)^\ell$ as a *row tuple* of $\mathbf{A}$ if there exists a row vector $v \in \mathbb{F}_q^n$ such that $\mathbf{a} = v\mathbf{A}$, where $v\mathbf{A} = (vA_1, \ldots, vA_\ell)$. We refer to vA_i as the *i-th coordinate* of $\mathbf{a}$ for every $i \in [\ell]$. For example, consider the matrix tuple $\mathbf{A} = (A_1, A_2) \in \mathrm{M}(3 \times 4, \mathbb{F}_2)^2$:

$$A_1 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Then $\mathbf{a} = ((1,1,0,1),(0,0,1,1))$ is a row tuple of $\mathbf{A}$ because $\mathbf{a} = (1,0,1)\mathbf{A}$, and $(1,1,0,1)$ is the first coordinate of $\mathbf{a}$. On the other hand, $\mathbf{a}' = ((1,1,1,1),(1,1,0,1))$ is not because no row vector $v \in \mathbb{F}_2^3$ satisfies $\mathbf{a}' = v\mathbf{A}$.

In this paper, we investigate IBC-tuples of a matrix tuple. Structurally, we prove Theorem I.3. Algorithmically, we give an algorithm to compute a representative IBC-tuple sequence, denoted as $\mathbf{B}_1, \ldots, \mathbf{B}_k$, for an input matrix tuple $\mathbf{A}$. This sequence consists of IBC-tuples that are mutually non-equivalent (treating IBC-tuples as matrix tuples), and every IBC-tuple of $\mathbf{A}$ is equivalent to one of the IBC-tuples in the sequence. Moreover, the representative IBC-tuple sequence produced by our algorithm is canonical. That is, for two equivalent matrix tuples $\mathbf{A}$ and $\mathbf{A}'$, the representative IBC-tuple sequences $\mathbf{B}_1, \ldots, \mathbf{B}_k$ for $\mathbf{A}$ and $\mathbf{B}'_1, \ldots, \mathbf{B}'_k$ for $\mathbf{A}'$ satisfy that $\mathbf{B}_i$ and $\mathbf{B}'_i$ are right-equivalent for every $i \in [k]$.

Based on such a representative IBC-tuple sequence, we can compute the canonical form of the matrix tuple by selecting IBC-tuples in a certain way from the linear spaces spanned by IBC-tuples right-equivalent to each of $\mathbf{B}_1, \ldots, \mathbf{B}_k$ according to Theorem I.3.

In this overview, we focus on computing a representative IBC-tuple sequence for a matrix tuple canonically. We emphasize that this is a challenging task because an IBC-tuple may have a lot of equivalent IBC-tuples that are not right-equivalent to itself. Selecting a representative IBC-tuple among these equivalents canonically requires an in-depth analysis of the structure of equivalent IBC-tuples in the input matrix tuple.

Our algorithm systematically explores the structure of the matrix tuple by detecting non-trivial characteristic subspaces of row tuples. It continues this process until the information gathered from these characteristic subspaces enables the construction of the desired IBC-tuples in a canonical way. Here, a linear subspace of row tuples (or row vectors) for a matrix tuple is characteristic if, under any automorphism of the matrix tuple by left-right action, the subspace remains invariant.

The characteristic subspace naturally emerges when specific row tuples or row vectors are distinguished from others. For instance, let us consider a matrix tuple where the first matrix is not full row rank. In this case, all the row tuples with zero vector as their first coordinate form a characteristic row tuple subspace within the linear space spanned by all the row tuples of the matrix tuple. As another example, all the row vectors in the first matrix of a matrix tuple form a characteristic row vector subspace. However, in some cases, identifying the characteristic row tuple subspace is not as straightforward as illustrated in previous examples, requiring a careful analysis of the matrix tuple structure.

A. Matrix tuple without non-trivial characteristic row tuple subspace

The first question is under which circumstances the matrix tuple does not have a non-trivial characteristic row tuple subspace and how to compute a representative IBC-tuple sequence canonically in such cases. To address this question, let us consider the case in which the matrix tuple is square. Without loss of generality, we assume that every matrix in the matrix tuple is full rank. Other-

wise, it would be possible to obtain some characteristic row tuple subspace using the aforementioned approach.

For a length-ℓ matrix tuple $\mathbf{A} = (A_1, \ldots, A_\ell)$, we investigate the induced matrix tuple $\mathbf{G} = (A_1^{-1}A_2, \ldots, A_1^{-1}A_\ell)$ of length $\ell - 1$ under the conjugation action. We show that $\mathbf{A}$ has no nontrivial characteristic row tuple subspace if only if the matrix algebra generated by $\mathbf{G}$ is isomorphic to a finite field. Furthermore, our algorithm verifies whether the matrix algebra generated by $\mathbf{G}$ is a finite field. If it is not, our algorithm proceeds to produce a characteristic row tuples subspace of $\mathbf{A}$ based on the classification of matrix algebra by the Artin–Wedderburn–Mal'tsev theory. If the matrix algebra generated by $\mathbf{G}$ is isomorphic to a finite field, we show that all the IBC-tuples of the matrix tuple are equivalent, and thus, any representative IBC-tuple sequence contains a single IBC-tuple. In addition, we give an algorithm to select a representative IBC-tuple canonically. Specifically, the IBC-tuples selected for the sequence among equivalent input matrix tuples are right-equivalent.

Let us consider an example with $\mathbf{A} = (A_1, A_2, A_3) \in \mathrm{M}(4 \times 4, \mathbb{F}_2)^3$ being the following matrix tuple

$$A_1 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

It can be verified that all the A_i matrices are invertible. Let $\mathbf{G} = (A_1^{-1}A_2, A_1^{-1}A_3)$. The matrix algebra generated by $\mathbf{G}$ is isomorphic to $\mathbb{F}_4$.

For the case that the matrix algebra generated by $\mathbf{G}$ is isomorphic to a finite field, we observe that if an IBC-tuple $\mathbf{B}$ contains a row tuple $\mathbf{a}$ (i.e., there is a row vector v such that $\mathbf{a} = v\mathbf{B}$), then all the row tuples containing a vector that is a linear combination of the coordinates of $\mathbf{a}$ must also be in $\mathbf{B}$. For example, start with the row tuple $\mathbf{a}_1 = e_1\mathbf{A} = ((1,0,1,0),(0,0,0,1),(1,0,1,1))$, the row tuple $\mathbf{a}_2 = (1,1,0,1)\mathbf{A} = ((0,0,0,1),(1,0,1,1),(1,0,1,0))$ is in any IBC-tuple containing $\mathbf{a}_1$ because the first coordinate of $\mathbf{a}_2$ is the same as the second coordinate of $\mathbf{a}_1$.

On the other hand, we show that in this case, any row-submatrix tuple $\mathbf{C}$ can be decomposed into one or a few IBC-tuples if and only if any row tuple containing a coordinate being a row vector in $\mathbf{C}$ is contained in $\mathbf{C}$. As any row tuple containing a row vector that is a linear combination of coordinates of $\mathbf{a}_1$ and $\mathbf{a}_2$ is also a linear combination of $\mathbf{a}_1$ and $\mathbf{a}_2$, the row-submatrix tuple $\mathbf{B}$ of two rows with $\mathbf{a}_1$ as the first row and $\mathbf{a}_2$ as the second row is an IBC-tuple of $\mathbf{A}$.

Furthermore, let $\mathbf{a}'_1$ be an arbitrary row tuple of $\mathbf{A}$ and $\mathbf{a}'_2$ be the row tuple of $\mathbf{A}$ whose first coordinate equals the second coordinate of $\mathbf{a}'_1$. Then the row-submatrix tuple $\mathbf{B}'$ of two rows with $\mathbf{a}'_1$ as the first row and $\mathbf{a}'_2$ as the second row is also an IBC-tuple of $\mathbf{A}$, right isomorphic $\mathbf{B}$. Hence, a representative IBC-tuple can be obtained by identifying row tuples that have to belong to the same IBC-tuple (as $\mathbf{a}_1$ and $\mathbf{a}_2$ in this example) and defining the relations of these row tuples canonically (as the second coordinate of the first row of the IBC-tuple equals the first coordinate of the second row in this example).

As another example, let $\mathbf{A}' = (A'_1, A'_2, A'_3)$ be the matrix tuple with

$$A'_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, A'_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix},$$

$$A'_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}.$$

Because the matrix algebra generated by $((A'_1)^{-1}A'_2, (A'_1)^{-1}A'_3)$ has a nontrivial radical (i.e., the largest nilpotent ideal of the matrix algebra) generated by the following matrices

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

the linear span of the row vectors in the matrices of the radical, i.e., $\langle\{(0,0,0,1),(0,0,1,0)\}\rangle$, is a characteristic row vector subspace of $\mathbf{A}'$. Consequently, $\langle\{e_2\mathbf{A}', e_3\mathbf{A}'\}\rangle$ is a characteristic row tuple subspace because the row tuples in this subspace have all coordinates in the characteristic row vector subspace. We will address this situation in Section II-C.

Next, we turn to the scenario where the matrix tuple is a rectangle with no obvious characteristic row tuple subspace. In this case, the rectangle must have more columns than rows because otherwise, there are characteristic row tuple subspaces, like the linear space spanned by row tuples with the i-th coordinate being zero.

On the other hand, although there is no characteristic row tuple subspace for such rectangle matrix tuples,

there must be nontrivial characteristic row vector subspaces because there are more columns than rows. Let U_i be the linear space spanned by row vectors of the i-th matrix in the matrix tuple. U_i is a nontrivial characteristic row vector subspace for each i if the i-th matrix is nonzero.

In our algorithm, we utilize the correspondence between these characteristic row vector subspaces to either identify some characteristic row tuple subspaces or reduce the rectangle matrix tuple problem to the square matrix tuple problem. Specifically, if U_i and U_j have some nontrivial intersection, then such an intersection allows us to find some nontrivial characteristic row tuple subspace. Otherwise, the entire row vector space corresponds to a direct sum of some of the U_i row vector subspaces. In this case, one can canonically define the correspondence between row vectors from different U_i subspaces and then create a square matrix tuple corresponding to the rectangle matrix tuple in terms of the IBC-tuple structure. Hence, computing a representative IBC-tuple sequence for the rectangle matrix tuple is reduced to computing a representative IBC-tuple sequence for the constructed square matrix tuple.

B. Matrix tuple with direct sum row tuple decomposition

In Section II-A, we either obtain some characteristic row tuple subspace or have a complete characterization of the IBC-tuples and have an algorithm to construct a representative IBC-tuple sequence for such a matrix tuple. The next major question is how to canonically compute a representative IBC-tuple sequence given some nontrivial characteristic row tuple subspaces. In this section, we consider the base case that the entire row tuple space is the direct sum of some characteristic row tuple subspaces. We will study the general case in Section II-C.

The solution for the direct sum of characteristic row tuple subspaces is to identify the correspondence between row tuples from different characteristic row tuple subspaces and construct a new matrix tuple without nontrivial characteristic row tuple subspaces by merging corresponding row tuples from different characteristic row tuple subspaces into a row tuple in the new matrix tuple. For example, consider the following matrix tuple $\mathbf{A} = (A_1, \ldots, A_4) \in \mathrm{M}(4 \times 4, \mathbb{F}_2)^4$.

$$A_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, A_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Naturally, $\mathbf{A}$ has two characteristic row tuple subspaces: T_1, which contains all the row tuples of $\mathbf{A}$ whose first coordinates are zero, i.e., $T_1 = \langle\{e_1\mathbf{A}, e_2\mathbf{A}\}\rangle$, and T_2 which contains all the row tuples of $\mathbf{A}$ whose second coordinates are zero, i.e., $T_2 = \langle\{e_3\mathbf{A}, e_4\mathbf{A}\}\rangle$. Consequently, let U_1 be the linear space spanned by the second coordinates of the row tuples in T_1 (i.e., $U_1 = \langle\{e_1, e_2\}\rangle$) and U_2 be the linear space spanned by the first coordinates of the row tuples in T_2 (i.e., $U_2 = \langle\{e_3, e_4\}\rangle$). U_1 and U_2 are characteristic row vector subspaces, and the entire row vector space of $\mathbf{A}$ is the direct sum of U_1 and U_2.

Since both the second coordinates of T_1 and the fourth coordinates of T_2 are row vectors in U_2, we can define an isomorphism $f : T_1 \to T_2$ such that $f(\mathbf{a}) = \mathbf{b}$ for any $\mathbf{a} \in T_1, \mathbf{b} \in T_2$ if and only if the second coordinate of $\mathbf{a}$ is equal to the fourth coordinate of $\mathbf{b}$. Then, by choosing an arbitrary linear basis of the row tuples in T_1, we can construct a matrix tuple $\mathbf{C} = (C_1, \ldots, C_8) = \mathrm{M}(2 \times 4, \mathbb{F}_2)^8$ such that for any $\mathbf{c}$ as a row tuple of $\mathbf{C}$, the first four coordinates of $\mathbf{c}$ corresponds to a row tuple $\mathbf{a}$ of T_1, and the last four coordinates of $\mathbf{c}$ correspond to the $f(\mathbf{a})$ of T_2. One example of $\mathbf{C}$ is

$$C_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, C_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

$$C_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, C_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C_5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, C_6 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$C_7 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, C_8 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

By the correspondence between $\mathbf{C}$ and $\mathbf{A}$, we show that the IBC-tuples of $\mathbf{C}$ and $\mathbf{A}$ have a one-to-one correspondence. Therefore, once we obtain a canonical representative IBC-tuple sequence of $\mathbf{C}$ according to Section II-A, we can compute a representative IBC-tuple sequence of $\mathbf{A}$ canonically.

C. Matrix tuple with hierarchical row tuple decomposition

We discuss the algorithm for constructing the representative IBC-tuple sequence of a matrix tuple for general characteristic row tuple subspaces. The challenge arises from the potential impossibility of decomposing the entire row tuple space of the matrix tuple into a direct sum of a few characteristic row tuple subspaces, as demonstrated by $\mathbf{A}'$ defined in Section II-A.

1) Hierarchical row tuple decomposition and quotient matrix tuple: We organize the characteristic row tuple subspaces hierarchically to unveil the direct sum property for characteristic row tuple subspaces level by level via maintaining a *hierarchical row tuple decomposition*. For simplicity, in this overview, we assume the hierarchical row tuple decomposition contains only two levels.

A two-level hierarchical row tuple decomposition consists of a sequence of characteristic row tuple subspaces $T_1, \ldots, T_\zeta$ and a parameter $1 < h \leq \zeta$. The decomposition is hierarchical in the following sense: Let W be the linear space spanned by the row vectors in the row tuples of $T_h, \ldots, T_\zeta$. Then, the following two conditions hold:

1) Let S be the linear space spanned by row tuples with all coordinates in W. Then $T_h, \ldots, T_\zeta$ is a direct sum decomposition of S.
2) All the row tuples of the matrix tuple become the direct sum of $T_1, \ldots, T_{h-1}$ after shrinking all the row vectors in W to zero for each row tuple of the matrix tuple. (The row tuples in $T_h, \ldots, T_\zeta$ and the row tuples in $T_1, \ldots, T_{h-1}$ that are also in S shrink to zero row tuples.)

Based on a two-level hierarchical row tuple decomposition of a matrix tuple, we can construct a *quotient matrix tuple* for the matrix tuple by shrinking all vectors in W to zero. We ensure that the resulting matrix tuple has all the rows linearly independent by arbitrarily choosing a linear basis of the row tuples after shrinking. For example, let $\mathbf{A} = (A_1, A_2) \in \mathrm{M}(8 \times 8, \mathbb{F}_2)^2$ with

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

$$A_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Let T_1 be the linear space spanned by all the row tuples of $\mathbf{A}$. By Section II-A, using the radical of the matrix algebra generated by $A_1^{-1}A_2$, $\mathbf{A}$ has a characteristic row tuple subspace $T_2 = \{e_3\mathbf{A}, e_4\mathbf{A}, e_7\mathbf{A}, e_8\mathbf{A}\}$. T_1 and T_2 with $h = 1$ form a two-level hierarchical row tuple decomposition of $\mathbf{A}$. Consequently, $W = \langle\{e_3, e_4, e_7, e_8\}\rangle$, and the matrix tuple $\mathbf{Q} = (Q_1, Q_2)$ with

$$Q_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, Q_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (2)$$

is a quotient matrix tuple of $\mathbf{A}$ such that e_1, e_2, e_3 and e_4 of $\mathbf{Q}$ correspond to $e_1 + W, e_2 + W, e_5 + W$ and $e_6 + W$, where $v + W$ denotes $\{v + w : w \in W\}$.

2) IBC-tuple construction via quotient matrix tuple: To obtain the IBC-tuples for an input matrix tuple, we investigate the relations between the IBC-tuples of the quotient matrix tuple and the input matrix tuple.

As the base case, by the definition of a two-level hierarchical row tuple decomposition, any quotient matrix tuple is associated with a direct sum row tuple decomposition induced by $T_1, \ldots, T_{h-1}$. Hence, we can use the approach in Section II-B to obtain a representative IBC-tuple sequence for an arbitrary quotient matrix tuple. In the rest of this section, we give an overview of our approach to computing a representative IBC-tuple sequence for a matrix tuple $\mathbf{A}$ based on a representative IBC-tuple sequence for a quotient matrix tuple $\mathbf{Q}$ of $\mathbf{A}$.

First, consider an arbitrary block-diagonal matrix tuple $\mathbf{D}$ equivalent to $\mathbf{A}$ such that all the blocks of $\mathbf{D}$ are indecomposable. We observe that the row vectors from different blocks of $\mathbf{D}$ that correspond to row vectors in W (the row vectors shrunk to zero when computing the quotient matrix tuple) of $\mathbf{A}$ via a left-right action span a row vector subspace of $\mathbf{D}$ corresponding to W of $\mathbf{A}$. If we shrink the row vectors in this row vector subspace to zero for $\mathbf{D}$, similar to constructing a quotient matrix tuple for $\mathbf{A}$, then we get a block-diagonal matrix tuple equivalent to $\mathbf{Q}$. We show that this block-diagonal matrix tuple has every block corresponding to a few IBC-tuples of $\mathbf{Q}$. So, we want to understand the following question: given an IBC-tuple $\mathbf{B}$ for $\mathbf{Q}$, if $\mathbf{A}$ has an IBC-tuple $\mathbf{C}$ containing a row-submatrix tuple corresponding to $\mathbf{B}$ via the correspondence between $\mathbf{A}$ and $\mathbf{Q}$, what is such a row-submatrix tuple?

To answer this question, we extend the IBC-tuples for $\mathbf{Q}$ to row-submatrix tuples of $\mathbf{A}$. We say a row-submatrix tuple $\mathbf{C}$ of $\mathbf{A}$ is an *extension* of an IBC-tuple $\mathbf{B}$ of $\mathbf{Q}$ in $\mathbf{A}$ if $\mathbf{C}$ corresponds to $\mathbf{B}$ via the correspondence between $\mathbf{A}$ and $\mathbf{Q}$. For example, consider the matrix tuple $\mathbf{A}$ defined by Equation (1) and quotient matrix tuple $\mathbf{Q}$ defined by Equation (2). Using the result from Section II-A, The following $\mathbf{B}$ is an IBC-tuple of $\mathbf{Q}$,

$$\mathbf{B} = \left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \right) \quad (3)$$

and the following $\mathbf{C}$ is an extension of $\mathbf{B}$ in $\mathbf{A}$

$$\mathbf{C} = \left(\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \right). \tag{4}$$

There are many feasible extensions for an IBC-tuple of $\mathbf{Q}$ in $\mathbf{A}$, but not all of them can be contained by an IBC-tuple of $\mathbf{A}$. By investigating the relations between extensions and row vectors in W, we observe that for an extension to be contained in an IBC-tuple of $\mathbf{A}$, the linear span of row vectors in the extension must contain only the necessary row vectors from W. We refer to such extensions as "essential extensions".

For example, the $\mathbf{C}$ defined by Equation (4) is not essential for $\mathbf{B}$ as defined by Equation (3) because there is an IBC-tuple of $\mathbf{A}$ containing an extension of $\mathbf{B}$ but not containing the row vectors e_7 and e_8, which are vectors in W that are also in the linear space spanned by row vectors of $\mathbf{C}$. The row-submatrix tuple $\mathbf{C}'$ defined below is an essential extension of $\mathbf{B}$ because all the IBC-tuples of $\mathbf{A}$ containing an extension of $\mathbf{B}$ always contain the row vector e_3, which is the only vector in W and also in the linear space spanned by the row vectors of $\mathbf{C}'$.

$$\mathbf{C}' = \left(\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \right)$$

We show that such essential extensions exist, and if an IBC-tuple of $\mathbf{A}$ contains a row-submatrix tuple corresponding to $\mathbf{B}$, then the IBC-tuple of $\mathbf{A}$ must contain an essential extension of $\mathbf{B}$. We also give an algorithm to compute the canonical essential extensions. Furthermore, we show that if the IBC-tuples right-isomorphic to a given IBC-tuple of $\mathbf{Q}$ satisfy Theorem I.3, then the essential extensions obtained for these IBC-tuples by our algorithm also have the linearity similar to Theorem I.3.

Second, we explore the connection between essential extensions of IBC-tuples of $\mathbf{Q}$ and row tuples in the linear span of $T_h, \ldots, T_\zeta$ by constructing a new matrix tuple, called the *compression matrix tuple*. Roughly speaking, to construct the compression matrix tuple, for each essential extension obtained, we use a new row tuple to canonically encode the intersection of W and the linear space spanned by row vectors of the essential extension. By the linearity of essential extensions for right-equivalent IBC-tuples of $\mathbf{Q}$, the new row tuples constructed for extensions of right-equivalent IBC-tuples of $\mathbf{Q}$ also span a linear space of row tuples.

The compression matrix tuple $\mathbf{E}$ consists of row tuples with each coordinate in W. It contains two parts: one part corresponds to the row tuples of $\mathbf{A}$ with all row vectors in W (i.e., row tuples in $T_h, \ldots, T_\zeta$), and another part corresponds to the newly constructed row tuples from essential extensions.

The construction of the compression matrix tuple naturally results in a direct sum decomposition of the characteristic row tuple subspaces. Therefore, we apply the algorithm described in Section II-B to find the IBC-tuples of the compression matrix tuple. If the algorithm in Section II-B returns a new characteristic row tuple subspace of $\mathbf{E}$, then we can use this characteristic row tuple subspace to further refine the hierarchical row tuple decomposition we have for $\mathbf{A}$. We then restart the entire process with the refined hierarchical row tuple decomposition.

Finally, we study the consequence of the algorithm in Section II-B returning a representative IBC-tuple sequence for the compression matrix tuple $\mathbf{E}$. We show that in this case, for any block-diagonal matrix tuple $\mathbf{D_A}$ equivalent to $\mathbf{A}$, there exists a block-diagonal matrix tuple $\mathbf{D_E}$ equivalent to $\mathbf{E}$, such that there is a one-to-one correspondence between the blocks of $\mathbf{D_A}$ and blocks of $\mathbf{D_E}$, thereby implying a correspondence between IBC-tuples of $\mathbf{A}$ and IBC-tuples of $\mathbf{E}$. By carefully analyzing this correspondence, we give an algorithm for constructing a representative IBC-tuple sequence for $\mathbf{A}$ canonically guided by the IBC-tuples of $\mathbf{E}$.

REFERENCES

[Alp93] Jonathan L. Alperin. *Local representation theory: Modular representations as an introduction to the local representation theory of finite groups*. Cambridge University Press, 1993. doi:10.1017/CBO9780511623592.

[AZGL+18] Zeyuan Allen-Zhu, Ankit Garg, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. Operator scaling via geodesically convex optimization, invariant theory and polynomial identity testing. In *Proceedings of the 50th Annual Symposium on Theory of Computing, STOC*, pages 172–181, 2018. doi:10.1145/3188745.3188942.

[Bab16] László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the 48th Annual Symposium on Theory of Computing, STOC 2016*, pages 684–697, 2016. doi:10.1145/2897518.2897542.

[Bab19] László Babai. Canonical form for graphs in quasipolynomial time: preliminary report. In *Proceedings of the 51st Annual Symposium on Theory of Computing, STOC*, pages 1237–1246, 2019. doi:10.1145/3313276.3316356.

[BCS+13] László Babai, Xi Chen, Xiaorui Sun, Shang-Hua Teng, and John Wilmes. Faster canonical forms for strongly regular graphs. In *2013 54th Annual Symposium on Foundations of Computer Science, FOCS*, pages 157–166, 2013. doi:10.1109/FOCS.2013.25.

[BDN+23] Markus Bläser, Dung Hoang Duong, Anand Kumar Narayanan, Thomas Plantard, Youming Qiao, Arnaud Sipasseuth, and Gang Tang. The alteq signature scheme: Algorithm specifications and supporting documentation. *NIST PQC Submission*, 2023. URL: https://pqcalteq.github.io/.

[Bel00] G Belitskii. Normal forms in matrix spaces. *Integral Equations and Operator Theory*, 38(3):251–283, 2000. doi:10.1007/BF01291714.

[Ben98] David J Benson. *Representations and cohomology: Volume 1, basic representation theory of finite groups and associative algebras*, volume 1. Cambridge university press, 1998. doi:10.1017/S0013091500005253.

[BEO02] Hans Ulrich Besche, Bettina Eick, and E.A. O'Brien. A millennium project: Constructing small groups. *Intern. J. Alg. and Comput*, 12:623–644, 2002. doi:10.1142/S0218196702001115.

[Ber67] Elwyn R Berlekamp. Factoring polynomials over finite fields. *Bell System Technical Journal*, 46(8):1853–1859, 1967. doi:10.1002/j.1538-7305.1967.tb03174.x.

[BJP97] W. Bosma, J. J. Cannon, and C. Playoust. The Magma algebra system I: the user language. *J. Symb. Comput.*, pages 235–265, 1997. doi:10.1006/jsco.1996.0125.

[BL83] László Babai and Eugene M Luks. Canonical labeling of graphs. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing, STOC*, pages 171–183, 1983. doi:10.1145/800061.808746.

[BL08] Peter A. Brooksbank and Eugene M. Luks. Testing isomorphism of modules. *J. Algebra*, 320(11):4020 – 4029, 2008. doi:10.1016/j.jalgebra.2008.07.014.

[BLQW20] Peter A. Brooksbank, Yinan Li, Youming Qiao, and James B. Wilson. Improved algorithms for alternating matrix space isometry: From theory to practice. In *28th Annual European Symposium on Algorithms, ESA*, pages 26:1–26:15, 2020. doi:10.4230/LIPIcs.ESA.2020.26.

[Bor91] Armand Borel. *Linear algebraic groups*, volume 126 of *Graduate Texts in Mathematics*. Springer Science & Business Media, 1991. doi:10.1007/978-1-4612-0941-6.

[BS03] Genrich R. Belitskii and Vladimir V. Sergeichuk. Complexity of matrix problems. *Linear Algebra Appl.*, 361:203–222, 2003. Ninth Conference of the International Linear Algebra Society (Haifa, 2001). doi:10.1016/S0024-3795(02)00391-9.

[BW15] Peter A Brooksbank and James B Wilson. The module isomorphism problem reconsidered. *Journal of Algebra*, 421:541–559, 2015. doi:10.1016/j.jalgebra.2014.09.004.

[CGQ+24] Zhili Chen, Joshua A. Grochow, Youming Qiao, Gang Tang, and Chuanqi Zhang. On the complexity of isomorphism problems for tensors, groups, and polynomials III: actions by classical groups. In *15th Innovations in Theoretical Computer Science Conference, ITCS*, pages 31:1–31:23, 2024. doi:10.4230/LIPICS.ITCS.2024.31.

[CNP+23a] Tung Chou, Ruben Niederhagen, Edoardo Persichetti, Lars Ran, Tovohery Hajatiana Randrianarisoa, Krijn Reijnders, Simona Samardjiska, and Monika Trimoska. Matrix equivalence digital signature. *NIST PQC Submission*, 2023. URL: https://www.meds-pqc.org/.

[CNP+23b] Tung Chou, Ruben Niederhagen, Edoardo Persichetti, Tovohery Hajatiana Randrianarisoa, Krijn Reijnders, Simona Samardjiska, and Monika Trimoska. Take your meds: digital signatures from matrix code equivalence. In *International Conference on Cryptology in Africa, AFRICACRYPT*, pages 28–52. Springer, 2023. doi:10.1007/978-3-031-37679-5_2.

[CZ81] David G Cantor and Hans Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, 36(154):587–592, 1981. doi:10.1007/978-1-4757-2226-0_2.

[DGT17] Dean Doron, François Le Gall, and Amnon Ta-Shma. Probabilistic logarithmic-space algorithms for laplacian solvers. In *APPROX/RANDOM*, pages 41:1–41:20, 2017. doi:10.4230/LIPICS.APPROX-RANDOM.2017.41.

[DM17] Harm Derksen and Visu Makam. Polynomial degree bounds for matrix semi-invariants. *Advances in Mathematics*, 310:44–63, 2017. doi:10.1016/j.aim.2017.01.018.

[DM20] Harm Derksen and Visu Makam. Algorithms for orbit closure separation for invariants and semi-invariants of matrices. *Algebra & Number Theory*, 14(10):2791–2813, 2020. doi:10.2140/ant.2020.14.2791.

[Dro79] Ju A Drozd. Tame and wild matrix problems. In *Representation Theory II: Proceedings of the Second International Conference on Representations of Algebras*, pages 242–258. Springer, 1979. doi:10.1007/BFb0088467.

[FN70] V. Felsch and J. Neubüser. On a programme for the determination of the automorphism group of a finite group. In *Computational Problems in Abstract Algebra*, pages 59–60, 1970. doi:10.1016/B978-0-08-012975-4.50011-4.

[FR85] Katalin Friedl and Lajos Rónyai. Polynomial time solutions of some problems in computational algebra. In Robert Sedgewick, editor, *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 153–162. ACM, 1985. doi:10.1145/22145.22162.

[GAP17] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.8.8*, 2017. URL: https://www.gap-system.org/.

[GGdOW20] Ankit Garg, Leonid Gurvits, Rafael Mendes de Oliveira, and Avi Wigderson. Operator scaling: Theory and applications. *Found. Comput. Math.*, 20(2):223–290, 2020. doi:10.1007/s10208-019-09417-z.

[Gow11] W. Timothy Gowers. Comment on Richard Lipton's blog: The group isomorphism problem: Apossible polymath problem?, 2011. URL: https://rjlipton.wordpress.com/2011/11/07/the-group-isomorphism-problem-a-possiblepolymath-problem/.

[GP69] I. M. Gelfand and V. A. Ponomarev. Remarks on the classification of a pair of commuting linear transformations in a finite-dimensional space. *Functional Anal. Appl.*, 3:325–326, 1969. doi:10.1007/BF01076321.

[GQ23a] Joshua A. Grochow and Youming Qiao. On the complexity of isomorphism problems for tensors, groups, and polynomials I: tensor isomorphism-completeness. *SIAM J. Comput.*, 52(2):568–617, 2023. doi:10.1137/21M1441110.

[GQ23b] Joshua A. Grochow and Youming Qiao. On the complexity of isomorphism problems for tensors, groups, and polynomials IV: linear-length reductions and their applications. *CoRR*, abs/2306.16317, 2023. doi:10.48550/ARXIV.2306.16317.

[GQ24] Joshua A. Grochow and Youming Qiao. On p-group isomorphism: Search-to-decision, counting-to-decision, and nilpotency class reductions via tensors. *ACM Trans. Comput. Theory*, 16(1):2:1–2:39, 2024. doi:10.1145/3625308.

[GQT22] Joshua A Grochow, Youming Qiao, and Gang Tang. Average-case algorithms for testing isomorphism of polynomials, algebras, and multilinear forms. *Journal of Groups, Complexity, Cryptology*, 14, 2022. doi:10.46298/jgcc.2022.14.1.9431.

[Her94] Daniel Hershkowitz. Paths in directed graphs and spectral properties of matrices. *Linear algebra and its Applications*, 212:309–337, 1994. doi:10.1016/0024-3795(94)90408-1.

[HH21] Masaki Hamada and Hiroshi Hirai. Computing the nc-rank via discrete convex optimization on cat (0) spaces. *SIAM Journal on Applied Algebra and Geometry*, 5(3):455–478, 2021. doi:10.1137/20M138836X.

[IKS10] Gábor Ivanyos, Marek Karpinski, and Nitin Saxena. Deterministic polynomial time algorithms for matrix

completion problems. *SIAM J. Comput.*, 39(8):3736–3751, 2010. doi:10.1137/090781231.

[IMQ+24] Gábor Ivanyos, Euan Mendoza, Youming Qiao, Xiaorui Sun, and Chuanqi Zhang. Faster isomorphism for p-groups of frattini class 2, 2024. In preparation.

[IQ19] Gábor Ivanyos and Youming Qiao. Algorithms based on *-algebras, and their applications to isomorphism of polynomials with one secret, group isomorphism, and polynomial identity testing. *SIAM J. Comput.*, 48(3):926–963, 2019. doi:10.1137/18M1165682.

[IQ23] Gábor Ivanyos and Youming Qiao. On the orbit closure intersection problems for matrix tuples under conjugation and left-right actions. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms*, pages 4115–4126. SIAM, 2023. doi:10.1137/1.9781611977554.CH158.

[IQS17] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Non-commutative Edmonds' problem and matrix semi-invariants. *computational complexity*, 26(3):717–763, Sep 2017. doi:10.1007/s00037-016-0143-x.

[IQS18] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Constructive non-commutative rank computation is in deterministic polynomial time. *Comput. Complex.*, 27(4):561–593, 2018. doi:10.1007/S00037-018-0165-7.

[Iva00] Gábor Ivanyos. Fast randomized algorithms for the structure of matrix algebras over finite fields. In *Proceedings of the 2000 international symposium on Symbolic and algebraic computation*, pages 175–183. ACM, 2000. doi:10.1145/345542.345620.

[KI04] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1/2):1–46, 2004. doi:10.1007/s00037-004-0182-6.

[LQ17] Yinan Li and Youming Qiao. Linear algebraic analogues of the graph isomorphism problem and the Erdős–Rényi model. In *58th Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 463–474, 2017. doi:10.1109/FOCS.2017.49.

[Mil78] Gary L. Miller. On the $n^{\log n}$ isomorphism technique (a preliminary report). In *STOC*, pages 51–58. ACM, 1978. doi:10.1145/800133.804331.

[Mul17] Ketan Mulmuley. Geometric complexity theory v: Efficient algorithms for noether normalization. *Journal of the American Mathematical Society*, 30(1):225–309, 2017. doi:10.1090/jams/864.

[MW21] Visu Makam and Avi Wigderson. Singular tuples of matrices is not a null cone (and the symmetries of algebraic varieties). *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 2021(780):79–131, 2021. doi:10.1515/crelle-2021-0044.

[Nat22] National Institute of Standards and Technology. Call for additional digital signature schemes for the post-quantum cryptography standardization process, October 2022. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf.

[NQT24] Anand Kumar Narayanan, Youming Qiao, and Gang Tang. Algorithms for matrix code and alternating trilinear form equivalences via new isomorphism invariants. *IACR Cryptol. ePrint Arch.*, page 368, 2024. To appear in Eurocrypt'24. URL: https://eprint.iacr.org/2024/368.

[Rin97] Claus Michael Ringel. The development of the representation theory of finite dimensional algebras 1968-1975. *London Mathematical Society Lecture Note Series*, pages 89–116, 1997.

[Rón90] Lajos Rónyai. Computing the structure of finite algebras. *J. Symb. Comput.*, 9(3):355–373, 1990. doi:10.1016/S0747-7171(08)80017-X.

[Ser00] Vladimir V. Sergeichuk. Canonical matrices for linear matrix problems. *Linear Algebra Appl.*, 317(1-3):53–102, 2000. doi:10.1016/S0024-3795(00)00150-6.

[Sun23] Xiaorui Sun. Faster isomorphism for p-groups of class 2 and exponent p. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 433–440. ACM, 2023. doi:10.1145/3564246.3585250.

[SW15] Xiaorui Sun and John Wilmes. Faster canonical forms for primitive coherent configurations. In *Proceedings of the forty-seventh Annual Symposium on Theory of Computing, STOC*, pages 693–702, 2015. doi:10.1145/2746539.2746617.

[SW19] Pascal Schweitzer and Daniel Wiebking. A unifying method for the design of algorithms canonizing combinatorial objects. In *Proceedings of the 51st Annual Symposium on Theory of Computing, STOC*, pages 1247–1258, 2019. doi:10.1145/3313276.3316338.

[TDJ+22] Gang Tang, Dung Hoang Duong, Antoine Joux, Thomas Plantard, Youming Qiao, and Willy Susilo. Practical post-quantum signature schemes from isomorphism problems of trilinear forms. In *Advances in Cryptology - EUROCRYPT 2022*, pages 582–612, 2022. doi:10.1007/978-3-031-07082-2_21.

[Wei84] David A Weinberg. Canonical forms for symmetric tensors. *Linear algebra and its applications*, 57:271–282, 1984.

[Wil19] James B. Wilson. The threshold for subgroup profiles to agree is logarithmic. *Theory Comput.*, 15:1–25, 2019. doi:10.4086/TOC.2019.V015A019.

[WL68] Boris Weisfeiler and Andrei Leman. The reduction of a graph to canonical form and the algebra which appears therein. 1968.