Lifting Constructions of PDAs for Coded Caching With Linear Subpacketization

V. R. Aravind®, Pradeep Kiran Sarvepalli®, and Andrew Thangaraj®, Senior Member, IEEE

Abstract—Coded caching is a technique where multicasting and coding opportunities are utilized to achieve better rate-memory tradeoff in cached networks. A crucial parameter in coded caching is subpacketization, which is the number of parts a file is to be split into for coding purposes. The Maddah-Ali-Niesen scheme has order-optimal rate, but the subpacketization is exponential in the number of users for certain memory regimes. In contrast, coded caching schemes designed using placement delivery arrays (PDAs) can have linear subpacketization with a penalty in rate. In this work, we propose several constructions of efficient PDAs through lifting, where a base PDA is expanded by replacing each entry by another PDA. By proposing and using the notion of Blackburn-compatibility of PDAs, we provide multiple lifting constructions with increasing coding gains. We compare the constructed coded caching schemes with other existing schemes for moderately high number of users and show that the proposed constructions are versatile and achieve a good rate-memory tradeoff at low subpacketizations.

Index Terms—Coded caching, placement delivery arrays, lifting construction, Blackburn-compatibility, broadcast channels.

I. INTRODUCTION

COMMUNICATION networks get overburdened with data traffic during peak hours and underutilized in off-peak hours. Caching is a technique to alleviate the high transmission load of a server in a communication network during peak hours, and it involves prefetching popular content and storing it nearer to or at the user’s device during off-peak hours. Depending on the limitations on memory, a part of these files would be prefetched and once the user makes a demand, the rest of the requested file will be transmitted. The fundamental problem in caching is the optimal tradeoff between the cache memory with each user versus the transmission load.

Maddah-Ali and Niesen had shown that coding can achieve significant gain over uncoded caching by making use of multicast opportunities [1]. They showed their scheme to be order optimal with an information-theoretic lower bound on the number of files needed to be transmitted (known as rate). This scheme achieves a coding gain (also known as global caching gain) in addition to the caching gain. Asymptotically, its coding gain is proportional to the number of users and that results in a rate independent of the number of users. A version of Maddah-Ali-Niesen (MN) scheme, optimal for uncoded prefetching, was presented in [2]. Though the exact capacity expression for rate of coded caching is still an open problem, several bounds have been presented in [3], [4], [5], [6], and [7]. The problem has been studied in several settings like decentralized caching [8], non-uniform demands [9], hierarchical caching [10], coded prefetching [11], [12], [13], content security [14], [15], demand privacy [16], [17], [18] to name a few.

An important parameter of interest in coded caching is subpacketization. It is the number of parts a file will be split into, for the purpose of coding. In the standard Maddah-Ali-Niesen scheme [1] for $\frac{1}{K} < \frac{M}{N} < \frac{K-1}{K}$, the subpacketization, denoted $f$, grows exponentially with $K$ and is given by $f = (\frac{K}{KM/N}) \approx 2^{KH(M/N)}$. This limits the utility of the scheme in practical scenarios where there may be a large number of users (large $K$). Hence, reducing subpacketization is important in coded caching schemes. It was shown in [19] that for rate independent of the number of users, the subpacketization should be superlinear in the number of users. A few other bounds relating subpacketization with other parameters were proposed in [20] and [21]. The pursuit towards lower subpacketization had led to formulating coded caching in combinatorial frameworks. Under certain constraints, the coded caching problem is equivalent to the design of placement delivery arrays [22], caching matrices [23], partial Latin rectangles with Blackburn property [19], [24], certain 3-uniform 3-partite hypergraphs [19], resolvable designs from linear block codes [25], induced matchings of a Ruzsa-Szeméredi graph [26], [27], strong edge coloring of the bipartite graph [28] or a clique cover for the complement of the square of the associated line graph [29]. These frameworks require the cache contents to be uncoded and symmetric with respect to all files. When the number of users is very large, rate of $O(K^\delta)$ for small $\delta$ is achievable with linear subpacketization (in $K$) from schemes based on dense Ruzsa-Szemerédi graphs [26], [30]. Some of the constructions for reducing subpacketization in practical scenarios, built based on the combinatorial frameworks are summarised in Table I.

Since the first version of our manuscript, there have been several works on coded caching with linear subpacketization. Notable references include [31], [32], [33], [34], [35], [36]. They utilize concepts like Hamming distance [35], [36].
In this work, we present a few construction schemes for placement delivery arrays or PDAs. PDA developed in [22] captures the placement and delivery schemes as non-integer and integer entries in an array that satisfies some conditions. The number of columns of a PDA indicates the number of users while the number of rows indicates the subpacketization. We focus on PDAs where the number of rows is linear with the number of columns. Drawing inspiration from the lifting constructions for low-density parity-check codes [38], we propose lifting constructions for PDAs, where we use PDAs of small size to obtain larger PDAs. We introduce a new technical notion of Blackburn-compatibility for PDAs that enable these lifting constructions. We propose a variety of constructions for PDAs which satisfy this constraint. This includes algebraic and randomized constructions. With the lifting constructions using these PDAs, we obtain good memory-rate tradeoffs with linear subpacketization. In particular, when the number of users has many divisors or is a power of 2, the memory-rate tradeoffs are close to that of [1]. The random construction ensures PDAs satisfying Blackburn-compatibility with good performance for arbitrary parameters. Our methods perform well both in terms of obtaining good coding gains and versatility. Our specific contributions are as follows:

i) We present a basic lifting construction that takes any two PDAs to construct larger PDAs with higher coding gain.

ii) We propose the notion of Blackburn-compatibility between PDAs and present a general lifting construction that uses a base PDA and a set of Blackburn-compatible PDAs.

iii) We present several constructions of Blackburn-compatible PDAs for the lifting constructions which includes both algebraic and randomized constructions.

iv) Using a combination of the constructions we propose, we demonstrate that significant coding gain and good memory-rate tradeoffs can be achieved with linear subpacketization.

The proposed coded caching schemes target the moderate (non-asymptotic) regime of parameters and are shown to be competitive with other existing schemes, particularly in terms of subpacketization. We recently became aware of some existing works [34], [39] that also combine PDAs to obtain new PDAs. Our constructions are more general and follow a different approach.

The rest of the paper is organized as follows. In Section II, we describe the system setup and the problem statement. In Section III, we propose the basic and general lifting constructions for PDAs. In Section IV, we propose several constructions for Blackburn-compatible PDAs which are the building blocks for the lifting constructions. We present our results and compare them with existing works in Section V and conclude in Section VI. An extended version of this paper is available at [40]. A summary of the constructions for Blackburn-compatible PDAs is provided in Table III of the extended version [40].

II. PRELIMINARIES

A. Coded Caching

Consider a server, holding N files $W_i$, $i \in [N] \triangleq \{0, 1, \ldots, N - 1\}$, of F bits each, connected to K users via a multicast link. User $k$, $k \in [K]$, has a cache $Z_k$ of size $MF$ bits. Coded caching works in two phases. In the first phase, called the placement phase, the cache $Z_k$ of User $k$ is populated with content by the server, while being unaware of the files demanded by the users. In the second phase, called the delivery phase, User $k$ demands file $D_k \in [N]$ from the server. Let $D = (D_0, D_1, \ldots, D_{K-1})$. Based on the demands and stored cache contents, the server multicasts packets of the same size. The entire multicast transmission from the server is denoted $X^D$ for a demand vector $D$, and we suppose that the length of $X^D$ is $RF$ bits. The quantities $M$ and $R$ are measures of cache size and rate of transmission, respectively.
The main requirement in a coded caching scheme is that User $k$ should be able to decode the file $W_{D_k}$ using $Z_k$ and $X^D$. We denote a coded caching scheme with $K$ users, $N$ files, local cache size $M$, and rate $R$ as a $(K, N; M; R)$ coded caching scheme, or as a $(K, N)$ scheme.

### B. Coded Caching Schemes From PDAs

We use the framework of placement delivery arrays for centralized coded caching schemes [22]. For positive integers $K$, $f$, $Z$, and a set of integers $S$, a $(K, f, Z, S)$ placement delivery array is an $f \times K$ matrix $P = [p_{j,k}], j \in [f], k \in [K]$, containing either a “$\star$” or integers from $S$ in each array cell such that they satisfy the following conditions.

**C1.** The symbol $\star$ appears $Z$ times in each column.

**C2.** Each integer $s \in S$ occurs at least once in the array.

**C3.** (Blackburn property) If the entries in two distinct cells $p_{j_1,k_1}$ and $p_{j_2,k_2}$ are the same integer $s \in S$, then $p_{j_1,k_1} = p_{j_2,k_2} = \star$.

If there is no ambiguity, we will use the notation $(K, f, Z, S)$ for the PDA corresponding to a subfile label and if $s \in S$ then $p_{j,k} = s$.

We propose several deterministic and a randomized construction for lifted LDPC codes, we start with a base PDA, and replace each integer or a $\star$ in the base PDA are called constituent PDAs.

#### III. LIFTING OR PROTOGRAPH-TYPE CONSTRUCTIONS

Constructions of PDAs with coding gain 2 are generally simple and several constructions exist with low subpacketization. To increase coding gain and obtain $g$-regular $f \times K$ PDAs for $2 < g < K$ without a significant increase in subpacketization, we employ the idea of lifting or protograph construction. Similar to the popular notion of protograph or lifted LDPC codes, we start with a base PDA, and replace each entry with another PDA. The PDAs that replace an integer or a $\star$ in the base PDA are called constituent PDAs.

An important requirement when lifting PDAs is that we have to ensure that the Blackburn property is preserved during the lifting. For this purpose, we define a constraint called Blackburn-compatability which needs to be satisfied by the constituent PDAs for the lifting to be valid.

We propose several deterministic and a randomized construction for lifting of PDAs and compare them with other existing PDAs in terms of their memory-rate tradeoff and subpacketization.

#### A. Notation for PDAs

The following PDAs are used repeatedly in lifting constructions.

1) For an integer $t$, $I_n(t)$ denotes the $(n, n, n - 1, 1)$ PDA with the integer $t$ on the main diagonal and $\star$ in all other cells. $\overline{I}_n(t)$ denotes the $(n, n, n - 1, 1)$ PDA with the integer $t$ on the main anti-diagonal and $\star$ in all other cells.

2) The M-N scheme [1] for $t = \frac{KN}{M} = 1$ results in dense 2-PDAs. For a set $S = \{s_1, \ldots, s_m\}$, $m = n(n - 1)/2$, the following are $2-(n, n, n(n - 1)/2)$ PDAs:

$$
G_n(S) = \begin{bmatrix}
1 & \cdots & \cdots & \star \\
\vdots & \ddots & \ddots & \vdots \\
\star & \cdots & \cdots & \star \\
\star & \cdots & \cdots & \star \\
\star & \cdots & \cdots & \star \\
\end{bmatrix}
$$

$$
H_n(S) = \begin{bmatrix}
1 & \cdots & \cdots & \star \\
\vdots & \ddots & \ddots & \vdots \\
\star & \cdots & \cdots & s_{m-1} \\
\star & \cdots & \cdots & \star \\
\star & \cdots & \cdots & \star \\
\end{bmatrix}
$$

3) For a set $S$ of $n^2$ integers, $J_n(S)$ denotes the $(n, n, 0, S)$ PDA obtained by filling all the cells in the array with distinct integers from $S$ row-wise in the specified order.

#### B. Basic Lifting

In the basic lifting method, we start with a base PDA and replace $\star$’s with all-\$\star$ array, and replace integers with PDAs that contain disjoint sets of integers. A more general case of the following theorem and corollary are proved later in the paper as Theorem 3 and Corollary 2.

**Theorem 2 (Basic lifting):** Let $P_0$ be a $(K, f, Z, S_0)$ PDA. Let $P = \{P_i : i \in S_0\}$, where $P_i$ is an $(m, n, c, S_i)$ PDA and $S_i, S_j$ are disjoint if $i \neq j$. Let an array $B_P(P_0)$ be defined as
follows: (a) Each $\ast$ in $P_0$ is replaced by a $n \times m$ all-$\ast$ array. (b) Each integer $i \in S_0$ is replaced by $P_i \in P$. Then, $B_P(P_0)$ is a $(Km, f_n, Z_n + (f - Z)c, S)$ PDA, where $S = \bigcup_{i \in S_0} S_i$.

Example 1: Let $P_0 = \tilde{I}_2(1)$ with 1 replaced by $G_2(\{0, 1, 2\})$ resulting in a 4-regular $(6, 6, 4, 3)$ PDA \( \prod_{i,j} G_2(\{0, 1, 2\}) \).

The basic lifting construction is simple, and provides PDAs of various sizes with higher coding gains in a direct manner. Note that a lifting construction where each integer in a base PDA was replaced by $I_{s}(t)$ was proposed in [34]. The simplest $g$-regular construction by basic lifting is captured in the following corollary to Theorem 2.

Corollary 1: Let $P_0$ be a $g_0$-regular $(K_0 f_b Z_b, K_0 (f_b - Z_b))$ PDA. Let $P_1$ be a $g_1$-regular $(m, n, e, m(n-e))$ PDA, and $P = \{P_i : i \in \{K_0 (f_b - Z_b)\}\}$, where $P_i$ are copies of $P_1$ with its integers replaced by another disjoint set of integers. Then, $B_P(P_0)$, which is denoted simply as $B_P(P_0)$ in this case, is a $g_0 g_1$-regular $(K_0 m, f_b n, Z_b n + (f_b - Z_b) e, K_0 (f_b - Z_b) m(n-e))$ PDA.

C. General Lifting

In basic lifting, every $\ast$ in the base PDA is replaced with the all-$\ast$ array. For coded caching schemes with low cache memory, since we need the number of $\ast$s in each column in the lifted PDA to be low, replacing $\ast$s in the base PDA with non-trivial PDAs is beneficial. However, to ensure that the Blackburn property for the lifted PDA is not violated, the constituent PDAs that are used to replace the integers and $\ast$s need to satisfy some additional constraints. We introduce the notion of Blackburn-compatibility of PDAs to capture such constraints on the constituent PDAs.

1) Blackburn-Compatibility: Two $n \times n$ PDAs $P_0 = [p_{ij}^{(0)}]$ and $P_1 = [p_{ij}^{(1)}]$ are said to be Blackburn-compatible with respect to (w.r.t.) a third $n \times n$ PDA $P_3 = [p_{ij}^{(3)}]$ if, whenever $p_{ij}^{(0)} = p_{ij}^{(1)} \neq \ast$, we have $p_{ij}^{(3)} = \ast$. In other words, if two entries in $P_0$ and $P_1$ are a common integer $s$, the mirrored locations of $s$ in $P_3$ are $\ast$s. For $g \geq 2$, we say $P_0, \ldots, P_{g-1}$ are Blackburn-compatible w.r.t. $P_3$ when they are pairwise Blackburn-compatible with $P_3$.

To see the connection between Blackburn- and lifting, consider an integer $s$ occurring $g$ times in a base PDA. The rows and columns containing $s$ in the PDA, after permutations, can be rearranged into the PDA $I_g(s)$. So, any valid lifting of the base PDA needs to necessarily include a valid lifting of $I_g(s)$. Validity of a certain lifting of $I_g(s)$ and Blackburn-compatibility are shown to be equivalent in the following lemma.

Lemma 1 (Equivalence between Blackburn-compatibility and lifting): Suppose $P_0, P_1, \ldots, P_{g-1}$ are PDAs of the same size. Let $P_s^{(i,j)}$ for $i, j = 0, 1, 2, \ldots$ be copies of $P_i$ containing integers that are disjoint from each other and from the integers in $P_0, \ldots, P_{g-1}$. Then, the set $\mathcal{P} = \{P_0, \ldots, P_{g-1}\}$ is a set of PDAs Blackburn-compatible w.r.t. $P_3$ if and only if the following lifting of $I_g$ is a valid PDA:

\[
L_{P, s}(I_g) \triangleq \left( \begin{array}{cccc}
P_0 & P_s^{(0,1)} & \cdots & P_s^{(0,g-1)} \\
P_s^{(1,0)} & P_1 & \cdots & P_s^{(1,g-1)} \\
\vdots & \vdots & \ddots & \vdots \\
P_s^{(g-1,0)} & P_s^{(g-1,1)} & \cdots & P_{g-1} \\
\end{array} \right)
\]

Proof: The conditions for validity of the above PDA and the definition of Blackburn-compatibility are readily seen to be equivalent. Because of the disjointness properties of the integers in $P_s^{(i,j)}$, no additional conditions arise. □

The above lemma, beyond establishing the connection between lifting and Blackburn-compatibility, provides a way to visualize the mirrored locations and aids in constructions of Blackburn-compatible PDAs. We see that any two PDAs are Blackburn-compatible w.r.t. the trivial all-$\ast$ PDA. If $P_3$ is not all-$\ast$, the Blackburn-compatibility needs to be established more carefully. Before presenting tests for Blackburn-compatibility and general constructions, we show some illustrative examples.

Example 2: $P_0 = \left( \begin{array}{cc} 0 & 2 \\ 0 & 1 \\ 3 & 1 \\ 3 & 0 \end{array} \right)$, $P_1 = \left( \begin{array}{cc} 1 & 2 & \ast \\ 3 & 2 & \ast \end{array} \right)$ are Blackburn-compatible w.r.t. $I_g(t)$ for $t \notin [4]$. We see that $L_{(p_0, p_1), I_3}(I_2) = \left( \begin{array}{c} I_3(t_0) \\ I_3(t_1) \\ I_3(t_2) \\ I_3(t_3) \end{array} \right)$ is a $(6, 6, 3, 6)$ 3-PDA.

Example 3: Let

\[
P_3\{t_0, t_1, t_2, t_3\} = \left( \begin{array}{c} J_3(t_0) \\ J_3(t_1) \\ J_3(t_2) \\ J_3(t_3) \end{array} \right)
\]

\[
P_0 = \left( \begin{array}{c} H_{3,0}(9, 13, 17) \\ H_{3,1}(10, 14, 15) \\ H_{3,2}(11, 12, 16) \\ H_{3,3}(2, 3, 7) \end{array} \right), \quad P_1 = \left( \begin{array}{c} H_{3,0}(0, 4, 8) \\ H_{3,1}(1, 5, 6) \\ H_{3,2}(2, 3, 7) \end{array} \right)
\]

$P_0$ and $P_1$ are Blackburn-compatible w.r.t. $P_3$ for $t_0, t_1, t_2, t_3 \notin [18]$ and $L_{(p_0, p_1), I_3}(I_2)$ is a $(6, 24, 11, 20)$ 3-PDA.

The above examples are for two PDAs Blackburn-compatible w.r.t. a third non-trivial PDA. When $P_3$ is all-$\ast$, an arbitrary number of copies of a PDA, $\{P, P, \ldots\}$, are Blackburn-compatible w.r.t. the all-$\ast$ array, and this is used in the basic lifting of Theorem 2. However, for a general $P_3$, we require integer-disjoint copying of PDAs to ensure Blackburn-compatibility. Since this is a repeatedly occurring step in constructions, we record it as a lemma.

Lemma 2 (Replication of Blackburn-compatible PDAs): Given a set $\mathcal{P}$ of $b$ PDAs Blackburn-compatible w.r.t. $P_3$, the set of $mb$ PDAs formed by $m$ integer-disjoint copies of the PDAs in $\mathcal{P}$ is Blackburn-compatible w.r.t. $P_3$, for any integer $m > 0$.

Proof: The proof is immediate by the disjointness of the integers. □

2) General Lifting Theorem: Using Blackburn-compatible PDAs, a generalization of basic lifting is presented in the following theorem.
Theorem 3 (General lifting): Let $P_b$ be a $(K, f, Z_b, S_b)$ PDA. Let $g_s$ be the frequency of integer $s$ in $P_b$. For $s \in S_b$ and $t \in \{1, \ldots, g_s\}$, let $P_{s,t}$ be an $(m, n, Z_s, S_{s,t})$ PDA such that for any $s$, $P_{s,1}, \ldots, P_{s,g_s}$ are Blackburn-compatible w.r.t. an $(m, n, Z, S)$ PDA $P_s$, and, for distinct integers $s, s' \in S_b$, $S_{s,t}$ and $S_{s',t'}$ are disjoint. Let $P_{s,t}, r \in [KZ_b]_g$, be integer-disjoint copies of $P_s$, which are integer-disjoint with $P_{s,t}$ as well. Let an array $P$ be defined as follows:

1) $r$-th $s$ in $P_b$ is replaced by $P_{g,r}$ for $r \in [KZ_b]_g$.
2) $t$-th occurrence of integer $s \in S_b$ in $P_b$ is replaced by $P_{s,t}$ for $t = 1, \ldots, g_s$.

Then, $P$ is a $(Km, fn, Z_bZ_s + (f - Z_b)Z_n, S)$ PDA, where $S = \bigcup_{r \in [KZ_b]} S_{s,r} \bigcup \{\sum_{s \in S_b} \sum_{t \in [g_s]} S_{s,t}\}$, and $S_{s,t}$ is the set of integer-disjoint copies of $P_s$.

Proof: Clearly, $P$ is an $fn \times Km$ array. Each column of $P_b$ has $Z_b$ elements and $f - Z_b$ integers. So, each column of $P$ has $Z_bZ_s + (f - Z_b)Z_n$ elements satisfying $S_b$. Since $P_b$ is a PDA, $P$ has all integers in $S$ occurring at least once, satisfying $S_b$. Finally, we need to verify the Blackburn property for $P$. Let $P_{i,j}$ and $q_{i,j}$ denote the $(i,j)$-th elements of $P_b$ and $P$, respectively. Let $i/n$ denote the quotient when $i$ is divided by $n$, and let $j/m$ be defined similarly. If $q_{i_1,j_1} = q_{i_2,j_2} = s$ is in the lifted PDA, we necessarily have $p_{i_1/n,j_1/m} = p_{i_2/n,j_2/m}$ in the base PDA $P_b$. Because different integers are expanded to PDAs containing disjoint sets of integers, we have $p_{i_1/n,j_1/m} = p_{i_2/n,j_2/m}$ is $\forall s$ by the Blackburn property. Since an $s$ is replaced by $P_{s,t}$ and an entry in $P_{s,t}$ is $\exists s$ if the corresponding entry in $P_b$ is $\exists s$, $q_{i_1,j_1} = q_{i_2,j_2}$ implies $P_b$ is due to the Blackburn-compatibility of the PDAs replacing an integer w.r.t. $P_{g,r}$. Hence, the Blackburn property $C_3$ is satisfied.

Clearly, basic lifting is a special case of general lifting, where $P_b$ is the all-$s$ array and $P_{s,t}$ are arbitrary PDAs. However, if a non-trivial $P_b$ is to be used, then we require as many Blackburn-compatible PDAs w.r.t. $P_b$ as the largest integer frequency of the base PDA. We will see general methods to construct Blackburn-compatible PDAs in the next sections.

The following corollary of the above general lifting theorem presents a sufficient condition for regular lifting using Blackburn-compatibility. We skip the proof as the parameters are easy to verify.

Corollary 2 (General regular lifting): Let $P_b$ be a $(K_b, f_b, Z_b, S_b)$ PDA. Let $P_b$ be an $(m, n, Z_s, m(n - Z_s)/g_r)$ g-PDA. Let $P_b$ be a $(m, n, Z_s, m(n - Z_s)/g_r)$ g-PDA. Let $P = \{P_0, P_1, \ldots, P_{g-1}\}$ be a set of $n \times m$ PDAs satisfying the following conditions:

- the number of $s$'s in every column of every $P_i$ is equal to $e$, an integer occurring in any one $P_i$ occurs a total of $g$ times across all $P_i$'s, $P$ is Blackburn-compatible w.r.t. $P_b$. For $s \in [K (f - Z_b)/g_s]$ let $P_s = \{P_s,0, \ldots, P_{s,g_s-1}\}$ be an integer-disjoint copy of $P_b$. Let the lifting of $P_b$ using Theorem 3 with $P_{s,t}$ as constituent PDAs be denoted $L_{P,P'}(P_b)$.

Then, $L_{P,P'}(P_b)$ is a $g$-regular $(K,m, f_m, Z_bZ_s + (f_b - Z_b)c, Z_bZ_s - (f_b - Z_b)c)$ PDA.

A crucial requirement for lifting with $P_s$ not being all-$s$ is sets of Blackburn-compatible PDAs. Integer-disjoint copying is one simple method for constructing any number of Blackburn-compatible PDAs w.r.t. any $P_s$. For specific choices of $P_s$, other methods of construction could improve upon integer-disjoint copying, and we consider such constructions next.

IV. CONSTRUCTIONS OF BLACKBURN-COMPATIBLE PDAS

We will present some general constructions for Blackburn-compatible PDAs using the following ideas - (1) permutation of integer indices and blocks to ensure mirrored locations are $\forall s$, (2) tiling of identity/regular PDAs, (3) recursive methods, and (4) a randomized construction.

One strategy to construct Blackburn-compatible PDAs is to use existing PDAs from Section III-A for $P_b$ and $P_0$ and obtain the rest of $P_i$'s by transforming $P_0$ using permutations, transpose etc. In Corollary 2, when $P_b$ is $g_r$-regular and each $P_i$ in $P$ is $g_i$-regular with same set of integers, we have $g = g_0g_i$. One natural choice for $P_s$ is $I_g(t)$ since it is a linear PDA with high coding gain. In most cases, we will consider the choice of $P_s$ as $I_g(t)$. The following lemma provides a test for Blackburn-compatibility of PDAs with respect to $I_g(t)$.

Lemma 3 (Test for compatibility w.r.t. $I_g(t)$): Two $g \times g$ PDAs $P_0$ and $P_1$ are Blackburn-compatible with $I_g(t)$ for $t$ not appearing in $P_0$ or $P_1$ iff $p_{1,1}^{(0)} = p_{1,1}^{(1)} \neq \exists s$ implies $i_0 \neq j$ and $i_1 \neq j_0$. In words, mirrored locations of integers should be off-diagonal.

Proof: If $p_{1,1}^{(0)} = p_{1,1}^{(1)} \neq \exists s$ implies $i_0 \neq j$ and $i_1 \neq j_0$, then for no $i \in [g]$ we need $p_{1,i}^{(0)} \neq \exists s$. Hence $P_b$ and $P_1$ are Blackburn-compatible w.r.t. any $g \times g$ PDA which has integers only in its diagonal. Now, let $P_b$ and $P_1$ are Blackburn-compatible w.r.t. $I_g(t)$. Assume that $s \in S_b \cap S_1$ such that $p_{1,i_0}^{(1)} = p_{1,i_j}^{(1)} = s$ and $i_0 = j_1$. This implies that cell $(i_0, i_0)$ of $I_g(t)$ is $\exists s$. But this is a contradiction. Hence $i_0 \neq j_1$. Similarly we can prove that $i_1 \neq j_0$.

A. Permutation Constructions

Now we introduce two permutation operations to obtain $P_i$'s for $i > 0$ from $P_0$ when $P_0$ is either a 1-PDA or a 2-PDA as defined in Section III-A. These $P_i$'s will be Blackburn-compatible w.r.t. $I_g(t)$. The first permutation construction uses cyclic rotation of diagonal or anti-diagonal elements, for which, we need the following notation. Given an $n \times n$ PDA $P = [p_{ij}]$, where $i$ and $j$ take values from 0 to $n - 1$, a PDA $\pi_{D,1}(P)$ is defined as

$$\pi_{D,1}(P) = [p_{i+1,j}] = \begin{cases} p_{i,j}, & i = j, i = 1 \mod n \newline p_{i,j - 1}, & i \neq j \end{cases} .$$

Basically, $\pi_{D,1}(P)$ is identical to $P$ except for the diagonal entries, which are cyclically shifted down by one position. For an integer $l$, $\pi_{D,1}^{l}(P)$ denotes the PDA obtained by $l$ applications of $\pi_{D,1}$ on $P$. For negative $l$, the diagonal entries are shifted up $l$ times. A similar cyclic rotation of anti-diagonal elements in $P$ is denoted by $\pi_{D,1}^{l}(P)$, where $(i,j)$ on the
anti-diagonal goes to \((i - 1, j + 1)\) mod \(n\) and all other locations are retained.

Given a \(2n \times 2n\) PDA \(P = \left( P_{11}, P_{12}, P_{21}, P_{22} \right)\), where \(P_{ij}\) are \(n \times n\) blocks, a PDA \(\pi_{D,2}(P)\) is defined as

\[
\pi_{D,2}(P) = \left( \pi_{D,1}(P_{11}), \pi_{D,1}(P_{12}), \pi_{D,1}(P_{21}), \pi_{D,1}(P_{22}) \right).
\]

\(\pi_{D,2}(P)\) is identical to \(P\) except for the \(2n\) diagonal entries - the first \(n\) are circularly shifted down by one position, and the second \(n\) are circularly shifted up by one position. A similar cyclic rotation of anti-diagonal elements in \(P\) is denoted by \(\pi_{AD,2}(P)\).

**Lemma 4 (Cyclic rotation):** 1) (Construction C1)

-a) Given a 1-PDA \(P\), \(\{P, \pi_{D,1}(P)\}\) is Blackburn-compatible w.r.t. \(I_g(t)\), and \(\{P, \pi_{AD,1}(P)\}\) w.r.t. \(I_g(t)\).

-b) Letting \(P = J_g(g^2)\), \(\mathcal{P}_D = \{P, \pi_{D,1}(P), \ldots, \pi_{D,1}^{-1}(P)\}\) is a set of \(g\) 1-PDAs Blackburn-compatible w.r.t. \(I_g(t)\), and \(\mathcal{P}_AD = \{P, \pi_{AD,1}(P), \ldots, \pi_{AD,1}^{-1}(P)\}\) w.r.t. \(I_g(t)\).

-c) \(L_{\mathcal{P}_D} I_g(t)\) and \(L_{\mathcal{P}_AD} I_g(t)\) are regular \(g\)-regular \((g^2, g^2, g^2 - 2g + 1, g(2g - 1))\) PDAs.

2) (Construction C2)

-a) Letting \(Q_0 = G_{2g}(g(2g - 1))\) and \(Q_1 = H_{2g}(g(2g - 1))\), \(\{Q_0, \pi_{D,2}(Q_0)\}\) is Blackburn-compatible w.r.t. \(I_g(t)\), and \(\{Q_1, \pi_{AD,2}(Q)\}\) w.r.t. \(I_g(t)\).

-b) Then \(\mathcal{Q}_D = \{Q_0, \pi_{D,2}(Q_0), \ldots, \pi_{D,2}^{-1}(Q_0)\}\) is a set of \(g\) 2-PDAs Blackburn-compatible w.r.t. \(I_g(t)\), and \(\mathcal{Q}_AD = \{Q_1, \pi_{AD,2}(Q_1), \ldots, \pi_{AD,2}^{-1}(Q_1)\}\) w.r.t. \(I_g(t)\).

-c) \(L_{\mathcal{Q}_D} I_g(t)\) and \(L_{\mathcal{Q}_AD} I_g(t)\) are regular \(g\)-regular \((g^2, 2g^2, 2g^2 - 3g + 2, g(3g - 2))\) PDAs.

The integer \(t\) is chosen to be disjoint from the integers in \(P\), \(Q_0\), or \(Q_1\).

**Proof:** We prove Part 1(a) as follows. Suppose \([P]_{ij} = \pi_{D,1}(P)_{ij} \neq \ast\) for \(i \neq j\) (off-diagonal). Then, the corresponding mirrored \((i, j)\)-th entry of \(I_g(t)\) for \(i \neq j\) is \(\ast\). Suppose \([P]_{ii} = \pi_{D,1}(P)_{ii} \neq \ast\) (diagonal). Then, \(i \neq j\) because of the rotation, and the mirrored \((i, j)\)-th entry of \(I_g(t)\) for \(i \neq j\) is \(\ast\). The claim for \(P\) and \(\pi_{AD,1}(P)\) can be proved in a similar fashion. Part 1(b) uses Part 1(a) with \(P = J_g(g^2)\). Part 1(c) can be verified by a straightforward calculation. C2 can be proved in a similar way and we skip the details.

**Example 4:** For \(g = 3\), a set of 3 PDAs \(\mathcal{P}_3 = \{P_0, P_1, P_2\}\) that are pairwise Blackburn-compatible w.r.t. \(P_\ast = I_3(t)\) constructed using C1 is shown below.

\[
P_\ast = I_3 = \begin{pmatrix} t & \ast & \ast \\ \ast & t & \ast \\ \ast & \ast & t \end{pmatrix}, \quad P_0 = J_3 = \begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{pmatrix},
\]

\[
P_1 = \begin{pmatrix} 8 & 1 & 2 \\ 3 & 0 & 5 \\ 6 & 7 & 4 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 4 & 1 & 2 \\ 3 & 8 & 5 \\ 6 & 7 & 0 \end{pmatrix}.
\]
Construct $P_i$, $i = 1, \ldots, d-1$, as a block-wise concatenation of $F_i^{(j,k)}$ defined as follows.

$$P_i^{(j,k)} = \begin{cases} 
\pi_{AD,2}(P_0^{(j,j)}), & j = k, \\
\pi_{AD,1}(P_0^{(j,k)}), & j > k, \\
(P_i^{(k,j)})^T, & j < k.
\end{cases}$$

We claim that $\mathcal{P} = \{P_0, \ldots, P_{d-1}\}$ is the desired set. Within $P_i$, Blackburn-compatibility is satisfied by Construction T1. Between $P_i$ and $P_j$, we consider the diagonal case and two off-diagonal cases separately: (1) Two diagonal blocks $P_i^{(j,j)}$ and $P_j^{(j,j)}$ share the same set of integers only when $j \neq j'$, which means that the mirrored locations fall in an off-diagonal block $I_{2d}$ of $P_i$. So, by Construction C1, Blackburn-compatibility is satisfied for diagonal blocks. (2) Two off-diagonal blocks $P_i^{(j,k)}$ and $P_j^{(k,j)}$, $j \neq k$, share the same set of integers and have mirrored locations falling on diagonal blocks of $P_i$, which have $\mathcal{V}$s on all even diagonals. So, by Construction T2, Blackburn-compatibility is satisfied. (3) Two off-diagonal blocks $P_i^{(j,k)}$ and $P_j^{(k,j)}$, $j \neq k$, share the same set of integers, and the mirrored locations fall in an off-diagonal block $I_{2d}$ of $P_i$. So, by Construction C1, Blackburn-compatibility is satisfied.

Example 6: Let $g = 4$ and $d = 2$. $P_x$ is as defined in Lemma 7.

$$P_0 = \begin{pmatrix}
\star & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
0 & \star & 7 & 8 & 9 & 10 & 11 & 12 \\
2 & 8 & \star & 13 & 18 & 19 & 20 & 21 \\
3 & 9 & 14 & 18 & \star & 22 & 23 & 24 \\
4 & 10 & 15 & 19 & 22 & \star & 25 & 26 \\
5 & 11 & 16 & 20 & 23 & 25 & \star & 27 \\
6 & 12 & 17 & 21 & 24 & 26 & 27 & \star
\end{pmatrix}$$

$$P_1 = \begin{pmatrix}
\star & 22 & 23 & 25 & 3 & 4 & 5 & 15 \\
22 & \star & 24 & 26 & 9 & 10 & 18 & 12 \\
23 & 24 & \star & 27 & 14 & 6 & 16 & 17 \\
25 & 26 & 27 & \star & 11 & 19 & 20 & 21 \\
3 & 9 & 14 & 11 & \star & 0 & 1 & 7 \\
4 & 10 & 6 & 19 & 0 & \star & 2 & 8 \\
5 & 18 & 16 & 20 & 1 & 2 & \star & 13 \\
15 & 12 & 17 & 21 & 7 & 8 & 13 & \star
\end{pmatrix}$$

For the next blockwise construction, we consider $P_{x} = \mathcal{B}_{H_{g/d}^{-1}}(I_{d})$ to be a $g \times g$ PDA constructed by basic lifting of $I_{d}$ using $H_{g/d}((g/2(1-1)))$, for integers $g$, $d$ such that $d|g$. Therefore, it has $H_{g/d}$ repeated along the diagonal blocks and all-$\mathcal{V}$ blocks appearing in the off-diagonal blocks.

Lemma 8 (Construction BW3): For positive integers $g$ and $d$ such that $d|g$, there exist a set $\mathcal{P}$ of $d$-regular $(g, g, 1, g(2g-1))$ PDAs that are Blackburn-compatible w.r.t $P_{x}^{(i, j, k)} = L_{P_{x}, P_{x}}(I_{d})$, such that $L_{P_{x}, P_{x}}(I_{d})$ is a $2d$-regular $(gd, gd, gd, g + d - 2g + g/d, g(2g - d - g/d)/2)$ PDA.

Proof: Consider the $2g \times 2g$ $2$-PDA $P_0$ defined in the proof of Lemma 6 and its $2g^2 \times 2g^2$ blocks $P_0^{(j,k)}$, $j, k \in [d]$. Construct $P_i$, $i \in [d] \setminus \{0\}$ with blocks $P_i^{(j,k)}$ defined as follows.

$$P_i^{(j,k)} = \begin{cases} 
\pi_{AD,2}(P_0^{(j,j)}), & j = k, \\
\pi_{AD,1}(P_0^{(j,k)}), & j > k, \\
(P_i^{(k,j)})^T, & j < k.
\end{cases}$$

The main difference, when compared to Construction BW1, is the rotation of the diagonal blocks.

We claim that $\mathcal{P} = \{P_0, \ldots, P_{d-1}\}$ is the desired set, and skip the proof details, which are largely similar to the previous proof of Construction BW2.
Example 7: Let $g = 6$ and $d = 2$.

$$P_e = \begin{pmatrix} H_b & H_e \\ H_e & H_b \end{pmatrix}, \quad P_0 = \begin{pmatrix} \ast & 0 & 1 & 2 & 3 & 4 \\ 0 & \ast & 5 & 6 & 7 & 8 \\ 1 & 5 & \ast & 9 & 10 & 11 \\ 2 & 6 & 9 & \ast & 12 & 13 \\ 3 & 7 & 10 & 12 & \ast & 14 \\ 4 & 8 & 11 & 13 & 14 & \ast \end{pmatrix}.$$ 

Finally, consider a $P$, whose diagonal blocks contain one $\ast$ per column and off-diagonal blocks are copies of $I_g$. Let $T = H_{2n}([n(2n-1)])$ be the $2n \times 2n$ 2-PDA as in (2). Then we set $P_s^{(n)} = L_{\pi_{AD,2}(H_{2n})} :I_{2n}$ (1)). All the diagonal blocks of $P_s^{(n)}$ have $\ast$s on the diagonals, all off-diagonal blocks have $\ast$s on all even diagonals, and every column has $1 + (n-1)$ $(2n - 1) = 2n^2 + 3n + 2$ $\ast$s.

**Lemma 9**: (Construction BW4) For an integer $n$, there exist a set $\mathcal{P}$ of $2n$ $(2n^2, 2n^2, 0, 4n^3)$ 1-PDAs that are Blackburn-compatible w.r.t. $P_s^{(n)}$ (defined above) such that $L_{\mathcal{P}, P_s^{(n)}}(I_{2n})$ is a 2$n$-regular $(4n^3, 4n^3, 4n^3 - 8n^2 + 7n - 2, 2n^2(8n^2 - 7n + 2))$ PDA.

**Proof**: Consider the $2n^2 \times 2n^2$ 1-PDA $P_0$ defined as a blockwise concatenation of $2n \times 2n$ blocks. The $(j, k)$-th block, $j, k \in [n]$, $P_0^{(j,k)} = J_{2n}(S_{j,k})$, where $S_{j,k} = \{4n^2(nj + k), 4n^2(nj + k - 1)\}$ is a partition of the set $[4n^2]$ consisting of disjoint subsets of $4n^2$ consecutive integers.

Construct $P_i$ and $\tilde{P}_i$, $i = [n]$, as a blockwise concatenations of $2n \times 2n$ PDAs $P_i^{(j,k)}$, $j, k \in [n]$, defined as follows:

$$P_i^{(j,k)} = \begin{cases} P_{AD,1}(P_i^{(j-1,k)} \mod n, j-1 \mod n), & j = k, \\ P_{AD,1}(P_i^{(j,k)}), & j \neq k, \end{cases}$$

$$\tilde{P}_i^{(j,k)} = \begin{cases} (P_i^{(j,j)})^T, & j = k, \\ P_{AD,1}(P_i^{(j,k)}), & j \neq k. \end{cases}$$

When going from $P_{i-1}$ to $P_i$, a diagonal block is cyclically shifted down, and its anti-diagonal is cyclically shifted twice. An off-diagonal block has its anti-diagonal cyclically shifted twice. When going from $P_i$ to $\tilde{P}_i$, diagonal blocks are transposed. An off-diagonal block has its anti-diagonal cyclically shifted once.

We claim that $\mathcal{P} = \{P_0, \ldots, P_{n-1}, \tilde{P}_0, \ldots, \tilde{P}_{n-1}\}$ is Blackburn-compatible w.r.t. $P_s^{(n)}$. To prove the claim, we will consider multiple cases where two blocks of the PDAs in $\mathcal{P}$ share integers. (1) $P_i^{(j,k)}$ and $\tilde{P}_i^{(j',k')}$, $i \neq i'$, share integers if $j \neq k$, $P_i^{(j,k)}$ and $\tilde{P}_i^{(j',k')}$, $j \neq j'$, share integers only if $j \neq j'$. In both cases, the mirrored locations are an off-diagonal block of $P_e$, which is $I_{2n}$. By C1, Blackburn-compatibility follows. The same argument holds when $P$ is replaced with $\tilde{P}$ in this case. (2) $P_i^{(j,k)}$ and $\tilde{P}_i^{(j',k')}$ share integers. Mirrored locations are on diagonal blocks of $P_e$, which have $\ast$s on the diagonal. Blackburn-compatibility follows by T1. (3) $P_i^{(j,k)}$ and $\tilde{P}_i^{(j,k)}$ share integers if $j \neq k$, $P_i^{(j,j)}$ and $\tilde{P}_i^{(j,j)}$, $i \neq i'$, share integers only if $j \neq j'$. Mirrored locations are an off-diagonal block of $P_s$, which is $\tilde{I}_{2n}$. By C1, Blackburn-compatibility follows.

**Example 8**: For $n = 2$, the following set of $8 \times 8$ PDAs can be constructed using Lemma 9.

$$P_s = \begin{pmatrix} H_4(2:7) \\ I_4(1) \end{pmatrix},$$

$$P_0 = \begin{pmatrix} P_0^{(0,0)} & P_0^{(0,1)} \\ P_0^{(1,0)} & P_0^{(1,1)} \end{pmatrix},$$

$$\tilde{P}_0 = \begin{pmatrix} (P_0^{(0,0)})^T & P_{AD,1}(P_0^{(0,1)}) \\ P_{AD,1}(P_0^{(1,0)}) & (P_0^{(1,1)})^T \end{pmatrix},$$

$$P_1 = \begin{pmatrix} P_{AD,1}(P_0^{(1,1)}) & P_1^{(0,1)} \\ P_{AD,1}(P_1^{(0,1)}) & (P_1^{(1,1)})^T \end{pmatrix}.$$ 

**B. Tiling Construction**

The next lemma provides a tiling construction for regular Blackburn-compatible PDAs, and characterizes a tradeoff between the number of $\ast$s per column and the number of integers.

**Lemma 10** (Tiling): For a positive integer $g$ and a divisor $d$ of $g$, there exists a set $\mathcal{P}$ of $(d, g, g - d, d^2)$ $(g/d)$-PDAs that are Blackburn-compatible w.r.t. $I_g(t)$ (when $t$ does not appear in any PDA in $\mathcal{P}$). $L_{\mathcal{P}, I_g}(I_d)$ is a $g$-regular $(dg, dg, dg - 2d + 1, d^2(2d - 1))$ PDA.

**Proof**: Let $P_0, \ldots, P_{d-1}$ be $(d, d, 0, d^2)$ 1-PDAs Blackburn-compatible w.r.t. $I_g(t)$ obtained using the first part of Lemma 4. Each of these PDAs has $d$ integers per column. Let $g = d/d$. Replace an integer $s$ in $P_i$ with $I_0(s)$ to obtain a $(g, g, g - d, d^2)$ PDA, which we denote as $P_i$. It is easy to see that $\mathcal{P} = \{P_0, \ldots, P_{d-1}\}$ is a set of PDAs Blackburn-compatible w.r.t. $I_g(t)$. $L_{\mathcal{P}, I_g}(I_d)$ has $P_i$, $i \in [d]$, on the diagonal and $P_s$ is $I_g$ as off-diagonal blocks. Each integer in $[d^2]$ occurs $q$ times in every $P_i$, which adds up to a total of $qg = q$ times in $L$. $I_g$, by definition, contains one integer appearing $q$. Each column of $L_i$ has $P_i$ with $d$ integers on the diagonal and $I_g$ with one integer on $d - 1$ off-diagonal positions. So, each column has $d + d - 1 = 2d - 1$ integers, or $gd - (2d - 1)$ $\ast$s. So the parameters of $L$ are as claimed.

Using integer-diisjoint copying, the $d$ PDAs obtained from Lemma 10 can be replicated to obtain a set of $mg(g,g,g-d,d^2)(g/d)$-PDAs. This idea is captured in the following corollary.

**Corollary 3** (Tiling): For positive integers $g, b$ and $d = \gcd(g, b)$, there exists a set $\mathcal{P}$ of $(b, g, g - d, d^2)$ $(g/d)$-PDAs that are Blackburn-compatible w.r.t. $I_g(t)$ (when $t$ does not appear in any PDA in $\mathcal{P}$). Then $L_{\mathcal{P}, I_g}(I_b)$ is a $g$-regular $(bg, bg, bg - b - d + 1, b(b + d - 1))$ PDA.

**Proof**: Obtain $d$ PDAs using Lemma 10. Since $d/b$, the $d$ PDAs can be replicated with disjoint sets of integers to obtain $b$
PDAs with the same parameters. The parameters of $L_{P', I_6}(I_6)$ are easy to verify.

Example 9: For $n = 6$, $b = 4$, we have $d = 2$, $q = 3$ and obtain $Q = \{Q_0, Q_1, Q_2, Q_3\}$ shown below. Then note that $L(Q, I_6)$ is a 6-regular $(24, 24, 19, 20)$ PDA.

$$Q_0 = \left(I_3(0) I_3(1)\right), \quad Q_1 = \left(I_3(3) I_3(1)\right), \quad Q_2 = \left(I_3(4) I_3(5)\right), \quad Q_3 = \left(I_3(6) I_3(4)\right).$$

C. Recursive Construction

Repeated application of lifting can result in larger Blackburn-compatible PDAs starting from small-sized base PDAs. The following lemma is an important step in such recursive constructions.

Lemma 11: Let $P = \{P_0, \ldots, P_{g-1}\}$ be a set of $g \times n \times n$ PDAs Blackburn-compatible w.r.t. a PDA $P_*$ with $S$ as its set of integers. For sets of disjoint integers $S_j, j \in [g(n-1)] \cup \{t\}$, with $|S_j| = |S|$, let $P_*(S_j)$ indicate the PDA constructed by replacing integers in $P_*$ with those in $S_j$. Let $\pi(l) = (l + 1) \mod g$ be the cyclic shift permutation on $[g]$. For $i \in [g]$, let

$$P'_i = P_*(S_0) \cdots P_*(S_{g-2}) \quad (6)$$

Then, $P' = \{P'_0, \ldots, P'_{g-1}\}$ is a set of $g \times n \times n$ PDAs Blackburn-compatible w.r.t. $P'_*$. If $L_{P, P_0}(I_6)$ is a $g_0$-regular $ng \times ng$ PDA, then $L_{P', P'_*}(I_6)$ is a $gg_0$-regular $ng^2 \times ng^2$ PDA.

Proof: A $P_*(S_i)$ block appears off-diagonal at the same location in all $P'_i$. This does not violate Blackburn-compatibility w.r.t. $P'_*$. If $P_*(S_i)$ has all-$\pi$ arrays as off-diagonal blocks, a block that appears in the diagonal of $P'_i$ at its $k$-th position will be $P_{\pi'(k)}$ and that of $P'_j$ will be $P_{\pi'(k)}$. Since $\pi'(k) \neq \pi'(k)$ for $i \neq j$, these blocks do not violate Blackburn-compatibility with $P'_*$. Finally, we prove the regularity claim. If $L_{P_0}(I_6(s))$ is $g_0$-regular, then $P_*(S_i)$ is $g_0$-regular, and so is $P_*(S_i)$ for every $i$. So, $P'_i$ is $gg_0$-regular, which implies that $L_{P_0}(I_6(s))$ is $gg_0$-regular in the off-diagonal blocks. Since every $P'_i$ is a diagonal-block-permuted version of $L_{P_0}(I_6(s))$, $P'_i$ (which is a diagonal block in $L_{P_0}(I_6(s))$) is $g_0$-regular as well. Since the same set of integers appear in every $P'_i$, the diagonal blocks of $L_{P_0}(I_6(s))$ together are $gg_0$-regular.

We illustrate how to apply the above lemma repeatedly to construct larger PDAs. Consider the two 1-PDAs, $A_2(x)$ and $A_2'(x)$, which are Blackburn-compatible w.r.t. $I_2(t)$ for $t \notin \{x, x+1, x+2, x+3\}$ and obtained using C1 for $g = 2$. They are given by

$$A_2(x) = \begin{pmatrix} x & x+1 \\ x+2 & x+3 \end{pmatrix}, \quad A_2'(x) = \begin{pmatrix} x+3 & x+1 \\ x+2 & x \end{pmatrix}. \quad (7)$$

Lemma 12: Consider the following $2^r \times 2^r$ PDAs for $r > 1$.

$$A_2(x) = \begin{pmatrix} A_{2r-1}(x+2) & A_{2r-1}(x) \\ A_{2r-1}(x+1) & A_{2r-1}(x+2) \end{pmatrix}, \quad \text{and } A_2'(x) = \begin{pmatrix} A_{2r-1}(x+2) & A_{2r-1}(x) \\ A_{2r-1}(x+1) & A_{2r-1}(x+2) \end{pmatrix}, \quad (8)$$

where $A_2(x)$ and $A_2'(x)$ are as defined in (7). Let $S$ be the set of integers in $A_2(x)$ and $A_2'(x)$. For $r \geq 1$ and $t \notin S$, $A = \{A_2(x), A_2'(x)\}$ is a set of two Blackburn-compatible PDAs w.r.t. $I_2(t)$, and $L_A(I_2(s))$ is a $2^r$-regular $(2^r+1, 2^r+1, 2^r+1 - r - 2, 2r + 4)$ PDA.

Proof: Let $P_1 = A_2, P_2 = A_2'(x)$ and apply Lemma 11 $(r - 1)$ times recursively.

D. Randomized Construction

For positive integers $b, r, e, \alpha, \eta$, we propose a randomized algorithm $\text{RandBC}_{b, e, \alpha, \eta}$ that, when successful, will construct $\pi \times \alpha \times \eta$ PDAs $P_i, i \in [b]$, satisfying the following conditions:

- every $P_i$ contains $\pi \times \alpha \times \eta$ per column, $L_{(P_0, \ldots, P_{b-1})}(I_6)$ is $r$-regular,
- the set $\{P_0, \ldots, P_{b-1}\}$ is Blackburn-compatible w.r.t. $I_{r}(t, 1, \alpha)$,

$$P_1 = \begin{pmatrix} I_{r}(t, 1, \alpha) & \cdots & I_{r}(t, 1, \alpha) \\ \vdots & \ddots & \vdots \\ I_{r}(t, \eta, 1) & \cdots & I_{r}(t, \eta, 1) \end{pmatrix},$$

where $t_{j,k}$ are integers not occurring in the $P_i$’s.

A pseudocode for the random construction is given in Algorithm 1. The algorithm cycles through the $P_i$, starting with $P_0$, and adds integer $v$, one at a time, starting with $v = 1$. Each integer is added $r$ times. Free locations for adding $v$ in $P_i$ are identified, and a penalty term is calculated for each free location. The penalty tends to favour locations that minimize “wasting” of cells in other $P_j, j \neq i$, and those that improve column and row spread of the integers in $P_i$. If no free locations are found at any point, the algorithm fails. Since all conditions are maintained throughout, the algorithm outputs the required PDAs, if successful. Results of successful runs of $\text{RandBC}_{b, e, \alpha, \eta}$ are given in Table II. From the table, we observe that the randomized algorithm succeeds for a wide range of parameters of interest.

V. Results

We present lifted PDAs and corresponding coded caching schemes using the Blackburn-compatible PDAs constructed in the previous sections. One of the advantages of our work compared to previous works is the range of parameters supported by our constructions. To bring out the versatility of the lifting procedure, we present lifted PDAs constructed for a given number of users $K$ and a wide range of memory sizes and rates.
Algorithm 1 RandBC$^{h,r}_{\alpha,\eta}$

1. function CHECK($[P_i]_{x,y} \leftarrow v$): return TRUE, if setting 
   $(x, y)$-th position of $P_i$ as $v$ does not violate PDA, number 
   of $\ast$'s per column and Blackburn-compatibility conditions. 
   else return FALSE
2. define FREE($P_i, v$) = \{(x, y) : $[P_i]_{x,y} = \ast$; CHECK($[P_i]_{x,y} \leftarrow v$) = TRUE\}
3. $P_i$ $\leftarrow$ all-$\ast$ for all $i$, $s$ $\leftarrow$ $b\alpha(\eta r - e)$ (s: number of integers 
   occurring in all $P_i$)
4. $i = 0$ (start with $P_0$)
5. for $v \in \{1, \ldots, s\}$ do
6.  loop $r$ times
7.  if FREE($P_i, v$) is empty then declare FAILURE and 
   exit
8.  For $(x, y) \in$ FREE($P_i, v$), PENALTY($x, y$) = $N_r + N_c + w_r + w_c$, where (1) $N_r$ and $N_c$ are the number 
   of rows and columns of all other $P_j$, $j \neq i$, invalidated under 
   Blackburn-compatibility by setting $(x, y)$-th position of $P_i$ 
   as $v$, respectively. (2) $w_r$ and $w_c$ are the number of integers 
   in $x$-th row and $y$-th column of $P_i$, respectively.
9. $(x^*, y^*) =$ arg min$_{(x,y)\in\text{FREE}(P,v)}$ PENALTY($x, y$) 
   (if multiple, pick one at random)
10. Set $[P_i]_{x^*, y^*} \leftarrow v$, Move to next $i = i + 1 \ mod b$

TABLE II

EXAMPLES OF SUCCESSFUL RANDOM GENERATION OF $b \eta r \times \alpha r$ PDAS 
WITH $e$ $\ast$S PER COLUMN IN EACH PDA AND EACH INTEGER 
OCCURRING $r$ TIMES ACROSS ALL PDAS

| $\eta$ | $\alpha$ | $(r, e)$ | $b$ |
|---|---|---|---|
| 1 | 1 | (3,2), (4,1), (5,3), (6,4), (8,5), (10,8), (12,9), (15,14), (16,14), (20,18), (25,24), (30,28), (32,30), (50,48), (125,225) | 2 |
| 1 | 1 | (3,0), (4,0), (5,0), (6,3), (10,6), (12,9), (25,22) | 5 |
| 1 | 1 | (3,2), (5,1) | 20 |
| 2 | 1 | (5,6), (10,16), (25,47), (125,248) | 2 |
| 2 | 1 | (5,1), (10,11), (25,44), (50,98) | 5 |
| 2 | 1 | (5,1) | 50 |
| 4 | 1 | (5,12), (10,32), (25,96), (50,196), (125,496) | 2 |
| 4 | 1 | (5,2), (10,21), (25,89), (50,194) | 5 |
| 4 | 1 | (5,2) | 50 |

A. $K$ Is a Small Multiple of a Power of 2

Starting with 2-PDAs, we consider lifting to obtain PDAs 
with a coding gain of $2^r$, $r = 2, 3, \ldots$

Theorem 4 ($2^r$-lifting):

1. Given a 2-regular $(K_b, f_b, Z_b, S_b)$ PDA $P_b$, there exists 
a $2^r$-regular $(2^r K_b, f_b 2^r, (2^r - r - 1) f_b + r Z_b, (2 + 2^r) S_b + K_{Z_b})$ PDA.
2. For coded caching with $K$ users, if $2^r | K$ for an 
integer $r$, the memory-rate pair $(2^r K(1 - 2^{-r}(r + 1)) + r, 2^{-2r} K(r + 1) - 2^{-r} r)$ is achievable with linear 
subpacketization.

Proof: For the first part, use Corollary 2 to lift the given 
2-regular base PDA $P_0$ using the Blackburn-compatible PDAs 
obtained from Lemma 12 as constituent PDAs. Since $I_{2^r}(t)$ 
has $(2^r - 1) \ast$'s per column and $A_{2^r}(x_1)$, $A_{2^r}(x_2)$ have 
$(2^r - r - 1) \ast$'s per column, the number of $\ast$ in each column 
of the lifted PDA is $(Z)(2^r - 1) + (f_b - Z_b)(2^r - r - 1) = 
(2^r - r - 1) f_b + r Z$. The other parameter values are easy 
to establish.

For the second part, let $q = \frac{K}{2^r}$. Construct a $(q, q, 1, q(q-1)/2)$ $2^r$-PDA using the construction in (2). Lift this 
PDA using the above first part of the theorem to obtain a 
$(K, K(1 - (r + 1) 2^{-r}) + r, 2^{-2r} K(r + 1) - 2^{-r} K(r + 1) - 2^{-r} r)$ 
$2^r$-PDA. This results in the claimed memory-rate pair. □

The memory-rate and memory-subpacketization tradeoff for 
$K = 64$ using the lifting schemes BW2, C2 and $2^r$-lifting is 
compared with others in Figs. 1a and 1b. When $\frac{K}{2^r} = \frac{M}{N}$ 
or $\frac{K}{2^r} = \frac{M}{N}$, the PDAs from MN scheme provide the best 
coding gain with linear subpacketization. For our schemes, 
minimum values of $Z$ obtained for each coding gain $g$ are 
highlighted in red and are labelled using $(Z, g)$. For the points 
highlighted in solid red, details of lifting construction are 
provided in Table III. We see that the lifting scheme has better 
rate than the grouping scheme with $c = 8$ for all memory. 
For memory ratios $0.71 < M/N < 1$, lifting has better rate than 
grouping with $c = 4$. In Fig. 1b, we see that for most values of
TABLE III
LIFTING CONSTRUCTIONS OF SOME OF THE PDAs FROM FIG. 1 AND 2

| $K$   | Example 1                                                                 | Example 2                                                                 |
|-------|---------------------------------------------------------------------------|---------------------------------------------------------------------------|
| 84    | $\frac{(6,6)^2_{13}}{BW3}$                                               | $\frac{(3,3)^2_{13}}{C_2}$                                               |
| 64    | $\frac{(8,8)^4_{12}}{BW2}$                                               | $\frac{(2,2)^2_{C_2}}{(8,8)^4_{C_2}}$                                    |
|       | $\frac{(4,4)^2_{12}}{(16,10)^{11,15}}{(64,64)^{14}}$                     | $\frac{(4,4)^2_{C_2}}{(2,2)^2_{C_2}}{(64,64)^{12}}$                      |
| 240   | $\frac{(10,10)^2_{12}}{(240,240)^{16}_{22}}{BW5}$                        | $\frac{(4,4)^2_{C_2}}{(2,2)^2_{C_2}}{(64,64)^{20}}$                      |
|       | $\frac{(5,5)^2_{12}}{(50,50)^{20}_{30}}{BW3}$                           | $\frac{(5,5)^2_{C_2}}{(10,20)^2_{13}}{(250,250)^{22}_{452}}$            |

Fig. 2. Memory-rate tradeoff for $K = 84, 240, 250$.

cache memory ratio, the subpacketization of lifting schemes is noticeably better than other comparable schemes, except for the grouping scheme with $c = 8$ and the scheme from [35]. Also, our schemes provide a wide range of intermediate points without an increase in subpacketization.

B. $K$ With Many Divisors

When $K$ has many small divisors, lifting can be applied in multiple ways. A good approach is to consider as many possibilities of lifting as possible and find constructions that achieve the best tradeoff between cache memory and rate. In this section, we present illustrative memory-rate tradeoffs of various lifting constructions for $K = 84, 240, 250$ in Fig. 2. Memory-rate tradeoff of the coded caching schemes obtained from our constructions is compared with uncoded and Maddah-Ali-Niesen (MN) schemes. For $K = 84$, our schemes are also compared with existing schemes [19], [28], [31], [36] with linear subpacketization, i.e., $f = \Theta(K)$. We have included schemes with $f < 5K$ for the plots in Fig. 2. All existing schemes with linear subpacketization have restrictions on the values that $K$ can take. However, we noticed that for $K = 84$ some of the previous works have achievable schemes and hence chose this value for $K$ to compare our schemes with them. In these figures, $(M,R)$-pairs obtained by our constructions are shown in light blue and minimum values of $Z$ obtained for each coding gain $g$ are highlighted in red. It can be seen from Fig. 2a that our schemes exist in the range $\frac{M}{N} < 0.33$ and compare favourably to other schemes for $\frac{M}{N} \geq 0.33$. In Fig. 2b and Fig. 2c, the highlighted points are labeled with $(Z,g)$ when $f = K$ and $(Z, g)$ when $f \neq K$. Our PDA-based schemes are close to the MN scheme at different parts of the rate versus memory tradeoff curve. For selected points, description of the lifting construction is given in Table III. The notation $(K_b,f_b)^g_b{(m,n)}_{Z_b}^X$ denotes the lifting of a $g_b$-regular $(K_b,f_b,Z_b)^g_b$ PDA to a $g$-regular $(K,f,Z)^g_X$ PDA using a set of $m \times n$ Blackburn-compatible PDAs having $Z_c$'s per column and $P_c$'s per column. Here, $X$ denotes the construction method. Table III provides a sample of how a multitude of lifting sequences are possible when $K$ has many small divisors. For instance, in Example 1 for $K = 240$, a 2-regular $5 \times 5$ base PDA is first lifted to a 4-regular $30 \times 30$ PDA, which is in turn lifted to an 8-regular $240 \times 240$ PDA. The two liftings use Blackburn-compatible PDAs from Constructions BW3 and C2, respectively.
TABLE IV
RANDOMIZED CONSTRUCTION

| K   | f       | Gain | Construction                                      | M/N | R     |
|-----|---------|------|--------------------------------------------------|-----|-------|
| 250 | 250     | 125  | (2, 2)\(125,125,124,128\) (250, 250)\(125,250,248\) | 0.992 | 0.016 |
| 250 | 250     | 10   | (5, 5)\(5,5,4\) (25, 25)\(5,16\) (10, 10) | 0.792 | 5.2   |
| 256 | 256     | 4    | (64, 64)\(4,4,1,3\) (256, 256)\(4,26\) | 0.2578 | 47.5  |
| 250 | 500     | 125  | (2, 4)\(2,2\) (125,125,124,128) (250, 500)\(125,250,500,246\) | 0.992 | 0.016 |
| 250 | 500     | 10   | (5, 5)\(5,5,4\) (25, 25)\(5,16\) (10, 20)11,18 | 0.774 | 5.65  |
| 256 | 512     | 4    | (64, 128)\(4,4,1,3\) (256, 512)\(4,32\) | 0.2578 | 47.5  |
| 250 | 1000    | 125  | (2, 8)\(2,8\) (125,125,124,128) (250, 1000)\(125,250,1000,246\) | 0.992 | 0.016 |
| 250 | 1000    | 10   | (5, 5)\(5,5,4\) (25, 25)\(5,16\) (10, 40)25,36 | 0.765 | 5.875 |
| 256 | 1024    | 4    | (64, 256)\(4,4,1,3\) (256, 1024)\(4,64\) | 0.2578 | 47.5  |

Fig. 3. Memory-rate tradeoff for coded caching schemes from PDAs lifted using randomly constructed Blackburn-compatible PDAs. The red curve added for comparison in the first plot is the MR tradeoff obtained using \(2^r\)-lifting for \(K = f = 256\).

C. Randomized Construction: \(f = K, f = 2K, f = 4K\)

To obtain subpacketization as a small multiple of the number of users, randomized construction of Blackburn-compatible PDAs shown in Algorithm 1 can be used. For \(K = 250\) and \(K = 256\), Table IV shows some of the resulting lifting constructions. The memory-rate tradeoffs are shown in Fig. 3. We observe that a wide variety of memory vs rate tradeoffs are obtained by lifting with Blackburn-compatible PDAs obtained using the randomized algorithm. For \(K = f = 250\), we have added the memory-rate tradeoff obtained by deterministic \(2^r\)-lifting for \(K = f = 256\) (red line) for comparison. We see that the randomized method provides tradeoffs that are comparable with the deterministic one.

To summarize, we have demonstrated the versatility of the lifting construction. For a given number of users, lifting can provide multiple lifting constructions for PDAs offering a range of tradeoffs between cache memory size and rate at very low subpacketization.

VI. CONCLUSION

We propose several constructions for coded caching schemes with subpacketization linear with the number of users using the framework of placement delivery arrays. We introduced the notion of Blackburn-compatibility of PDAs and used this concept for a several lifting constructions of PDAs for a wide range of coding gains. We showed that new Blackburn-compatible PDAs can be built from existing sets of Blackburn-compatible PDAs through our blockwise and recursive constructions. We also proposed an algorithm to randomly construct Blackburn-compatible PDAs for any arbitrary setting. In many regimes, our lifting constructions are shown to perform better compared to other existing schemes for lower subpacketization.

REFERENCES

[1] M. A. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” IEEE Trans. Inf. Theory, vol. 60, no. 5, pp. 2856–2867, May 2014.
[2] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, “The exact rate-memory tradeoff for caching with uncoded prefetching,” IEEE Trans. Inf. Theory, vol. 64, no. 2, pp. 1281–1296, Feb. 2018.
[3] A. Sengupta, R. Tandon, and T. C. Clancy, “Improved approximation of storage-rate tradeoff for caching via new outer bounds,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jun. 2015, pp. 1691–1695.
[4] H. Ghasemi and A. Ramamoorthy, “Improved lower bounds for coded caching,” IEEE Trans. Inf. Theory, vol. 63, no. 7, pp. 4388–4413, Jul. 2017.
[5] C.-Y. Wang, S. H. Lim, and M. Gastpar, “A new converse bound for coded caching,” in Proc. Inf. Theory Appl. Workshop (ITA), Jan. 2016, pp. 1–6.
[6] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, “Characterizing the rate-memory tradeoff in cache networks within a factor of 2,” IEEE Trans. Inf. Theory, vol. 65, no. 1, pp. 647–663, Jan. 2019.
[7] K. Wan, D. Tuninetti, and P. Piantanida, “On the optimality of uncoded cache placement,” in Proc. IEEE Inf. Theory Workshop (ITW), Sep. 2016, pp. 161–165.
[8] M. A. Maddah-Ali and U. Niesen, “Decentralized coded caching attains order-optimal memory-rate tradeoff,” IEEE/ACM Trans. Netw., vol. 23, no. 4, pp. 1029–1040, Aug. 2015.
[9] U. Niesen and M. A. Maddah-Ali, “Coded caching with nonuniform demands,” *IEEE Trans. Inf. Theory*, vol. 63, no. 2, pp. 1146–1158, Feb. 2017.

[10] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, “Hierarchical coded caching,” *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3212–3229, Jun. 2016.

[11] L. Chen, P. Fan, and K. B. Letaief, “Fundamental limits of caching: Improved bounds for users with small buffers,” *IET Commun.*, vol. 10, no. 17, pp. 2315–2318, Nov. 2016.

[12] C. Tian and J. Chen, “Caching and delivery via interference elimination,” *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1548–1560, Mar. 2018.

[13] J. Gómez-Vilardebó, “Fundamental limits of caching: Improved rate-memory tradeoff with coded prefetching,” *IEEE Trans. Commun.*, vol. 66, no. 10, pp. 4488–4497, Oct. 2018.

[14] A. Sengupta, R. Tandon, and T. C. Clancy, “Fundamental limits of caching with secure delivery,” *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 355–370, Feb. 2015.

[15] V. Ravindrakumar, P. Panda, N. Karamchandani, and V. Prabhakaran, “Fundamental limits of secretive coded caching,” in *Proc. IEEE Intl. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 425–429.

[16] K. Wan and G. Caire, “On coded caching with private demands,” *IEEE Trans. Inf. Theory*, vol. 67, no. 1, pp. 358–372, Jan. 2021.

[17] V. R. Aravind, P. K. Sarvepalli, and A. Thangaraj, “Subpacketization in coded caching with demand privacy,” in *Proc. Nat. Conf. Commun. (NCC)*, Feb. 2020, pp. 1–6.

[18] S. Kamath, J. Ravi, and B. K. Dey, “Demand-privacy coded caching and the exact trade-off for \( N = K = 2 \),” in *Proc. Nat. Conf. Commun. (NCC)*, Feb. 2020, pp. 1–6.

[19] C. Shangguan, Y. Zhang, and G. Ge, “Centralized coded caching schemes: A hypergraph theoretical approach,” *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5755–5766, Aug. 2018.

[20] M. Cheng, Q. Yan, X. Tang, and J. Jiang, “Coded caching schemes with low rate and subpacketizations,” 2017, arXiv:1703.01548.

[21] H. S. Chittoor, P. Krishnan, K. V. S. Sree, and B. Mamillapalli, “Subexponential and linear subpacketization coded caching via projective geometry,” *IEEE Trans. Inf. Theory*, vol. 67, no. 9, pp. 6193–6222, Sep. 2021.

[22] Q. Yan, M. Cheng, X. Tang, and Q. Chen, “On the placement delivery array design for centralized coded caching scheme,” *IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5821–5833, Sep. 2017.

[23] S. Agrawal, K. V. Sushena Sree, and P. Krishnan, “Coded caching based on combinatorial designs,” in *Proc. IEEE Intl. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 1227–1231.

[24] I. M. Wanless, “A partial Latin squares problem posed by Blackburn,” Bull. Inst. Combinat. Appl., vol. 42, pp. 76–80, 2004.

[25] L. Tang and A. Ramamoorthy, “Coded caching schemes with reduced subpacketization from linear block codes,” *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 3099–3120, Apr. 2018.

[26] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, “Coded caching with linear subpacketization is possible using Ruzsa-Szemerédi graphs,” *IEEE Trans. Inf. Theory (ISIT)*, Jun. 2017, pp. 1237–1241.

[27] K. Shanmugam, A. G. Dimakis, J. Llorca, and A. M. Tulino, “A unified Ruzsa-Szemerédi framework for finite-length coded caching,” in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*.

[28] J. Michel and Q. Wang, “Placement delivery arrays from combinations of strong edge colorings,” in *IPN Prog. Rep.*, vol. 42, no. 154, pp. 42–154, 2003.

[29] V. R. Aravind, P. K. Sarvepalli, and A. Thangaraj, “Lifting constructions of PDAs for coded caching with linear subpacketization,” 2020, arXiv:2007.07475.

Pradeep Kiran Sarvepalli received the B.Tech. degree in electrical engineering from the Cochin University of Science and Technology, Kochi, India. He is currently pursuing the doctoral degree with the Department of Electrical Engineering, IIT Madras. He was a Software Engineer at Compro Technologies, New Delhi, India. His research interests include information theory, error correcting codes, and coded caching and privacy.

Andrew Thangaraj (Senior Member, IEEE) received the B.Tech. degree in electrical engineering from IIT Madras and the master's degree in electrical engineering and the Ph.D. degree in computer science from Texas A&M University. He was a Post-Doctoral Fellow at the University of British Columbia and the Georgia Institute of Technology. He was an IC Design Engineer at Texas Instruments India, Bengaluru. He is currently an Associate Professor with the Department of Electrical Engineering, IIT Madras. His research interests include quantum and classical error correcting codes, quantum cryptography, quantum computation, and coding for distributed storage.

V. R. Aravind received the B.Tech. degree in electronics and communication engineering from the Cochin University of Science and Technology, Kochi, India. He is currently pursuing the doctoral degree with the Department of Electrical Engineering, IIT Madras. He was a Software Engineer at Compro Technologies, New Delhi, India. His research interests include information theory, error correcting codes, and coded caching and privacy.