# Preconditioning techniques based on the Birkhoff-von Neumann decomposition 

Michele Benzi, Bora Uçar

## To cite this version:

Michele Benzi, Bora Uçar. Preconditioning techniques based on the Birkhoff-von Neumann decomposition. [Research Report] RR-8914, Inria - Research Centre Grenoble - Rhône-Alpes. 2016. hal01318486v3

## HAL Id: hal-01318486 <br> https://inria.hal.science/hal-01318486v3

Submitted on 13 Nov 2016 (v3), last revised 5 Jan 2017 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Preconditioning techniques based on the Birkhofi-von Neumann decomposition 

Michele Benzi, Bora Uçar

## RESEARCH

REPORT
$\mathrm{N}^{\circ} 8914$
May 2016
Project-Team ROMA

# Preconditioning techniques based on the Birkhoff-von Neumann decomposition 

Michele Benzi* ${ }^{*}$ Bora Uçar ${ }^{\dagger}$<br>Project-Team ROMA<br>Research Report n 8914 - May 2016 - 21 pages


#### Abstract

We introduce a class of preconditioners for general sparse matrices based on the Birkhoff-von Neumann decomposition of doubly stochastic matrices. These preconditioners are aimed primarily at solving challenging linear systems with highly unstructured and indefinite coefficient matrices. We present some theoretical results and numerical experiments on linear systems from a variety of applications.


Key-words: preconditioning, doubly stochastic matrix, Birkhoff-von Neumann decomposition

[^0]```
RESEARCH CENTRE
GRENOBLE - RHÔNE-ALPES
Inovallée
6 5 5 \text { avenue de l'Europe Montbonnot}
38334 Saint Ismier Cedex
```


## Une méthode de préconditionnement basée sur la décomposition de Birkhoff-von Neumann des matrices bistochastiques

Résumé : Nous introduisons une classe de préconditionnement pour matrices creuses générales fondées sur la décomposition de Birkhoff-von Neumann des matrices bistochastiques. Ces préconditionnements visent principalement à résoudre des systèmes linéaires difficiles dont les matrices sont nonstructurées et non-définies. Nous présentons quelques résultats théoriques et des expériences numériques pour des systèmes linéaires issus de tout un ensemble d'applications.

Mots-clés : préconditionnement, matrices bistochastiques, décomposition de Birkhoff-von Neumann

## 1 Introduction

We consider the solution of linear systems $A x=b$ where $A=\left[a_{i j}\right] \in \mathbb{R}^{n \times n}, b$ is a given vector and $x$ is the unknown vector. Our aim is to develop and investigate preconditioners for Krylov subspace methods for solving such linear systems, where $A$ is highly unstructured and indefinite.

For a given matrix $A$, we first preprocess it to get a doubly stochastic matrix (whose row and column sums are one). Then using this doubly stochastic matrix, we select some fraction of some of the nonzeros of $A$ to be included in the preconditioner. Our main tools are the well-known Birkhoff-von Neumann (BvN) decomposition (this will be discussed in Section 2 for completeness), and a splitting of the input matrix in the form $A=M-N$ based on its BvN decomposition. When such a splitting is defined, $M^{-1}$ or solvers for $M y=z$ are required. We discuss sufficient conditions when such a splitting is convergent and discuss specialized solvers for $M y=z$ when these conditions are met. We discuss how to build preconditioners meeting the sufficiency conditions. In case the preconditioners become restrictive in practice, their LU decomposition can be used. Our motivation is that the preconditioner $M^{-1}$ can be applied to vectors via a number of highly concurrent steps, where the number of steps is controlled by the user. Therefore, the preconditioners (or the splittings) can be advantageous for use in many-core computing systems. In the context of splittings, the application of $N$ to vectors also enjoys the same property. These motivations are shared by recent work on ILU preconditioners, where their fine-grained computation [9] and approximate application [2] are investigated for GPU-like systems.

The paper is organized as follows. We first give necessary background (Section 2) on doubly stochastic matrices and the BvN decomposition. We then develop splittings for doubly stochastic matrices (Section 3), where we analyze convergence properties and discuss algorithms to construct the preconditioners. Later in the same Section, we discuss how the preconditioners can be used for arbitrary matrices by some preprocessing. Here our approach results in a generalization of the Birkhoff-von Neumann decomposition for matrices with positive and negative entries where the sum of the absolute values of the entries in any given row or column is one. This generalization could be of interest in other areas. Then, we give experimental results (Section 4) with nonnegative and also arbitrary matrices, and then conclude the paper.

## 2 Background and definitions

Here we define several properties of matrices: irreducible, full indecomposable, and doubly stochastic matrices.

An $n \times n$ matrix $A$ is reducible if there exists a permutation matrix $P$ such that

$$
P A P^{T}=\left[\begin{array}{cc}
A_{1,1} & A_{1,2} \\
O & A_{2,2}
\end{array}\right]
$$

where $A_{1,1}$ is an $r \times r$ submatrix, $A_{2,2}$ is an $(n-r) \times(n-r)$ submatrix, and $1 \leq r<n$. If such a permutation matrix does not exist, then $A$ is irreducible [25, Ch. 1]. When $A$ is reducible, either $A_{1,1}$ or $A_{2,2}$ can be reducible as well, and we can recursively identify their diagonal blocks, until all diagonal blocks are irreducible. That is, we can obtain

$$
P A P^{T}=\left[\begin{array}{cccc}
A_{1,1} & A_{1,2} & \cdots & A_{1, s}  \tag{1}\\
0 & A_{2,2} & \cdots & A_{2, s} \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & A_{s, s}
\end{array}\right]
$$

where each $A_{i, i}$ is square and irreducible. This block upper triangular form, with square irreducible diagonal blocks is called Frobenius normal form [19, p. 532].

An $n \times n$ matrix $A$ is fully indecomposable if there exists a permutation matrix $Q$ such that $A Q$ has a zero-free diagonal and is irreducible [7, Chs. 3 and 4]. If $A$ is not fully indecomposable, but nonsingular, it can be permuted into the block upper triangular form

$$
P A Q^{T}=\left[\begin{array}{cc}
A_{1,1} & A_{1,2} \\
O & A_{2,2}
\end{array}\right],
$$

where each $A_{i, i}$ is fully indecomposable or can be further permuted into the block upper triangular form. If the coefficient matrix of a linear system is not fully indecomposable, the block upper triangular form should be obtained, and only the small systems with the diagonal blocks should be factored for simplicity and efficiency [12, Ch. 6]. We therefore assume without loss of generality that matrix $A$ is fully indecomposable.

An $n \times n$ matrix $A$ is doubly stochastic if $a_{i j} \geq 0$ for all $i, j$ and $A e=A^{T} e=e$, where $e$ is the vector of all ones. This means that the row sums and column sums are equal to one. If these sums are less than one, then the matrix $A$ is doubly substochastic. A doubly stochastic matrix is fully indecomposable or is block diagonal where each block is fully indecomposable. By Birkhoff's Theorem [4], there exist coefficients $\alpha_{1}, \alpha_{2}, \ldots, \alpha_{k} \in(0,1)$ with $\sum_{i=1}^{k} \alpha_{i}=1$, and permutation matrices $P_{1}, P_{2}, \ldots, P_{k}$ such that

$$
\begin{equation*}
A=\alpha_{1} P_{1}+\alpha_{2} P_{2}+\cdots+\alpha_{k} P_{k} . \tag{2}
\end{equation*}
$$

Such a representation of $A$ as a convex combination of permutation matrices is known as a Birkhoffvon Neumann decomposition ( BvN ); in general, it is not unique. The Marcus-Ree Theorem states that there are $\operatorname{BvN}$ decompositions with $k \leq n^{2}-2 n+2$ for dense matrices; Brualdi and Gibson [6] and Brualdi [5] show that for a fully indecomposable sparse matrix with $\tau$ nonzeros, we have BvN decompositions with $k \leq \tau-2 n+2$.

An $n \times n$ nonnegative, fully indecomposable matrix $A$ can be uniquely scaled with two positive diagonal matrices $R$ and $C$ such that $R A C$ is doubly stochastic [24].

## 3 Splittings of doubly stochastic matrices

### 3.1 Definition and properties

Let $b \in \mathbb{R}^{n}$ be given and consider solving the linear system $A x=b$ where $A$ is doubly stochastic. Hereafter we assume that $A$ is invertible. After finding a representation of $A$ in the form (2), pick an integer $r$ between 1 and $k-1$ and split $A$ as

$$
\begin{equation*}
A=M-N, \tag{3}
\end{equation*}
$$

where

$$
\begin{align*}
M & =\alpha_{1} P_{1}+\cdots+\alpha_{r} P_{r}  \tag{4}\\
N & =-\alpha_{r+1} P_{r+1}-\cdots-\alpha_{k} P_{k}
\end{align*}
$$

Note that $M$ and $-N$ are doubly substochastic matrices.
Definition 1. A splitting of the form (3) with $M$ and $N$ given by (4) is said to be a doubly substochastic splitting.

Definition 2. A doubly substochastic splitting $A=M-N$ of a doubly stochastic matrix $A$ is said to be standard if $M$ is invertible. We will call such a splitting an SDS splitting.

In general, it is not easy to guarantee that a given doubly substochastic splitting is standard, except for some trivial situation such as the case $r=1$, in which case $M$ is always invertible. We also have a characterization for invertible $M$ when $r=2$.

Theorem 1. Let $M=\alpha_{1} P_{1}+\alpha_{2} P_{2}$. Then, $M$ is invertible if (i) $\alpha_{1} \neq \alpha_{2}$, or (ii) $\alpha_{1}=\alpha_{2}$ and all the fully indecomposable blocks of $M$ have an odd number of rows (and columns). If any such block is of even order, $M$ is singular.

Proof. We investigate the two cases separately.
Case (i): Without loss of generality assume that $\alpha_{1}>\alpha_{2}$. We have

$$
M=\alpha_{1} P_{1}+\alpha_{2} P_{2}=P_{1}\left(\alpha_{1} I+\alpha_{2} P_{1}^{T} P_{2}\right)
$$

The matrix $\alpha_{1} I+\alpha_{2} P_{1}^{T} P_{2}$ is nonsingular. Indeed, its eigenvalues are of the form $\alpha_{1}+\alpha_{2} \lambda_{j}$, where $\lambda_{j}$ is the generic eigenvalue of the (orthogonal, doubly stochastic) matrix $P_{1}^{T} P_{2}$, and since $\left|\lambda_{j}\right|=1$ for all $j$ and $\alpha_{1}>\alpha_{2}$, it follows that $\alpha_{1}+\alpha_{2} \lambda_{j} \neq 0$ for all $j$. Thus, $M$ is invertible.

Case (ii): This is a consequence of the Perron-Frobenius Theorem. To see this, observe that we need to show that under the stated conditions the sum $P_{1}+P_{2}=P_{1}\left(I+P_{1}^{T} P_{2}\right)$ is invertible, i.e., $\lambda=-1$ cannot be an eigenvalue of $P_{1}^{T} P_{2}$. Since both $P_{1}$ and $P_{2}$ are permutation matrices, $P_{1}^{T} P_{2}$ is also a permutation matrix and the Frobenius normal form $T=\Pi\left(P_{1}^{T} P_{2}\right) \Pi^{T}$ of $P_{1}^{T} P_{2}$, i.e., the block triangular matrix

$$
T=\left[\begin{array}{cccc}
T_{1,1} & T_{1,2} & \cdots & T_{1, s}  \tag{5}\\
0 & T_{2,2} & \cdots & T_{2, s} \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & T_{s, s}
\end{array}\right]
$$

has $T_{i, j}=0$ for $i \neq j$. The eigenvalues of $P_{1}^{T} P_{2}$ are just the eigenvalues of the diagonal blocks $T_{i, i}$ of $T$. Note that there may be only one such block, corresponding to the case where $P_{1}^{T} P_{2}$ is irreducible. Each diagonal block $T_{i, i}$ is also a permutation matrix. Thus, each $T_{i, i}$ is doubly stochastic, orthogonal, and irreducible. Any matrix of this kind corresponds to a cyclic permutation and has its eigenvalues on the unit circle. If a block has size $r>1$, its eigenvalues are the $p$ th roots of unity, $\varepsilon^{h}, h=0,1, \ldots, p-1$, with $\varepsilon=e^{\frac{2 \pi i}{p}}$, e.g., by the Perron-Frobenius Theorem (see [17, page 53]). But $\lambda=-1$ is a $p$ th root of unity if and only if $p$ is even. Since $M$ is a scalar multiple of $P_{1}+P_{2}=P_{1}\left(I+P_{1}^{T} P_{2}\right)$, this concludes the proof.

Note that the fully indecomposable blocks of $M$ mentioned in the theorem are just connected components of its bipartite graph.

It is possible to generalize the condition (i) in the previous theorem as follows.
Theorem 2. A sufficient condition for $M=\sum_{i=1}^{r} \alpha_{i} P_{i}$ to be invertible is that one of the $\alpha_{i}$ with $1 \leq i \leq r$ be greater than the sum of the remaining ones.

Proof. Indeed, assuming (without loss of generality) that $\alpha_{1}>\alpha_{2}+\cdots+\alpha_{r}$, we have

$$
\begin{aligned}
M & =\alpha_{1} P_{1}+\alpha_{2} P_{2}+\cdots+\alpha_{r} P_{r} \\
& =P_{1}\left(\alpha_{1} I+\alpha_{2} P_{1}^{T} P_{2}+\cdots+\alpha_{r} P_{1}^{T} P_{r}\right)
\end{aligned}
$$

This matrix is invertible if and only if the matrix $\alpha_{1} I+\sum_{i=2}^{r} \alpha_{i} P_{1}^{T} P_{i}$ is invertible. Observing that the eigenvalues $\lambda_{j}$ of $\sum_{i=2}^{r} \alpha_{i} P_{1}^{T} P_{i}$ satisfy

$$
\left|\lambda_{j}\right| \leq\left\|\alpha_{2} P_{1}^{T} P_{2}+\cdots+\alpha_{r} P_{1}^{T} P_{r}\right\|_{2} \leq \sum_{i=2}^{r} \alpha_{i}<\alpha_{1}
$$

(where we have used the triangle inequality and the fact that the 2-norm of an orthogonal matrix is one), it follows that $M$ is invertible.

Again, this condition is only a sufficient one. It is rather restrictive in practice.

### 3.2 Convergence conditions

Let $A=M-N$ be an SDS splitting of $A$ and consider the stationary iterative scheme

$$
\begin{equation*}
x^{k+1}=H x^{k}+c, \quad H=M^{-1} N, \quad c=M^{-1} b \tag{6}
\end{equation*}
$$

where $k=0,1, \ldots$ and $x^{0}$ is arbitrary. As is well known, the scheme (6) converges to the solution of $A x=b$ for any $x^{0}$ if and only if $\rho(H)<1$. Hence, we are interested in conditions that guarantee that the spectral radius of the iteration matrix

$$
H=M^{-1} N=-\left(\alpha_{1} P_{1}+\cdots+\alpha_{r} P_{r}\right)^{-1}\left(\alpha_{r+1} P_{r+1}+\cdots+\alpha_{k} P_{k}\right)
$$

is strictly less than one. In general, this problem appears to be difficult. We have a necessary condition (Theorem 3), and a sufficient condition (Theorem 4) which is simple but restrictive.

Theorem 3. For the splitting $A=M-N$ with $M=\sum_{i=1}^{r} \alpha_{i} P_{i}$ and $N=-\sum_{i=r+1}^{k} \alpha_{i} P_{i}$ to be convergent, it must hold that $\sum_{i=1}^{r} \alpha_{i}>\sum_{i=r+1}^{k} \alpha_{i}$.

Proof. First, observe that since $P_{i} e=e$ for all $i$, both $M$ and $N$ have constant row sums:

$$
M e=\sum_{i=1}^{r} \alpha_{i} P_{i} e=\beta e, \quad N e=-\sum_{i=1}^{r} \alpha_{i} P_{i} e=(\beta-1) e,
$$

with $\beta:=\sum_{i=1}^{r} \alpha_{i} \in(0,1)$. Next, observe that $M^{-1} N e=\lambda e$ is equivalent to $N e=\lambda M e$ or, since we can assume that $\lambda \neq 0$, to $M e=\frac{1}{\lambda} N e$. Substituting $\beta e$ for $M e$ and $(\beta-1) e$ for $N e$ we find

$$
\beta e=\frac{1}{\lambda}(\beta-1) e,
$$

or

$$
\lambda=\frac{\beta-1}{\beta} .
$$

Hence, $\frac{\beta-1}{\beta}$ is an eigenvalue of $H=M^{-1} N$ corresponding to the eigenvector $e$. Since $|\lambda|=\frac{1-\beta}{\beta}$, we conclude that $\rho(H) \geq 1$ for $\beta \in\left(0, \frac{1}{2}\right]$. This concludes the proof.

Theorem 4. Suppose that one of the $\alpha_{i}$ appearing in $M$ is greater than the sum of all the other $\alpha_{i}$. Then $\rho\left(M^{-1} N\right)<1$ and the stationary iterative method (6) converges for all $x^{0}$ to the unique solution of $A x=b$.

Proof. Assuming (without loss of generality) that

$$
\begin{equation*}
\alpha_{1}>\alpha_{2}+\cdots+\alpha_{k} \tag{7}
\end{equation*}
$$

(which, incidentally, insures that the matrix $M$ is invertible) we show below that

$$
\begin{equation*}
\|H\|_{2}=\left\|\left(\alpha_{1} P_{1}+\cdots+\alpha_{r} P_{r}\right)^{-1}\left(\alpha_{r+1} P_{r+1}+\cdots+\alpha_{k} P_{k}\right)\right\|_{2}<1 \tag{8}
\end{equation*}
$$

This, together with the fact that $\rho(H) \leq\|H\|_{2}$, ensures convergence. To prove (8) we start by observing that

$$
\begin{aligned}
M & =\alpha_{1} P_{1}+\cdots+\alpha_{r} P_{r} \\
& =\alpha_{1} P_{1}\left(I+\frac{\alpha_{2}}{\alpha_{1}} Q_{2}+\cdots+\frac{\alpha_{r}}{\alpha_{1}} Q_{r}\right),
\end{aligned}
$$

where $Q_{i}=P_{1}^{T} P_{i}$. Thus, we have

$$
M^{-1}=\left(\alpha_{1} P_{1}+\cdots+\alpha_{r} P_{r}\right)^{-1}=\frac{1}{\alpha_{1}}(I-G)^{-1} P_{1}^{T}
$$

where we have defined

$$
G=-\frac{1}{\alpha_{1}} \sum_{i=2}^{r} \alpha_{i} Q_{i}
$$

Next, we observe that $\|G\|_{2}<1$, since

$$
\|G\|_{2} \leq \frac{1}{\alpha_{1}} \sum_{i=2}^{r} \alpha_{i}\left\|Q_{i}\right\|_{2}=\frac{\alpha_{2}+\cdots+\alpha_{r}}{\alpha_{1}}<1
$$

as a consequence of (7). Hence, the expansion

$$
(I-G)^{-1}=\sum_{\ell=0}^{\infty} G^{\ell}
$$

is convergent, and moreover

$$
\begin{aligned}
\left\|(I-G)^{-1} P_{1}^{T}\right\|_{2} & =\left\|(I-G)^{-1}\right\|_{2} \leq \frac{1}{1-\|G\|_{2}} \\
& \leq \frac{1}{1-\left(\frac{\alpha_{2}}{\alpha_{1}}+\cdots+\frac{\alpha_{r}}{\alpha_{1}}\right)}
\end{aligned}
$$

The last inequality follows from the fact that $\|G\|_{2} \leq \sum_{i=2}^{r} \frac{\alpha_{i}}{\alpha_{1}}$, as can be seen using the triangle inequality and the fact that $\left\|Q_{i}\right\|_{2}=1$ for all $i=2, \ldots, r$.

Hence, we have

$$
\left\|M^{-1} N\right\|_{2} \leq \frac{1}{\alpha_{1}} \frac{1}{1-\left(\frac{\alpha_{2}}{\alpha_{1}}+\cdots+\frac{\alpha_{r}}{\alpha_{1}}\right)}\left\|\alpha_{r+1} P_{r+1}+\cdots+\alpha_{k} P_{k}\right\|_{2}
$$

Using once again the triangle inequality (applied to the last term on the right of the foregoing expression) we obtain

$$
\left\|M^{-1} N\right\|_{2} \leq \frac{\alpha_{r+1}+\cdots+\alpha_{k}}{\alpha_{1}-\left(\alpha_{2}+\cdots+\alpha_{r}\right)}
$$

Using the condition (7) we immediately obtain that

$$
\frac{\alpha_{r+1}+\cdots+\alpha_{k}}{\alpha_{1}-\left(\alpha_{2}+\cdots+\alpha_{r}\right)}<1
$$

therefore $\left\|M^{-1} N\right\|_{2}<1$ and the iterative scheme (6) is convergent.
Note that if condition (7) is satisfied, then the value of $r$ in (4) can be chosen arbitrarily; that is, the splitting will converge for any choice of $r$ between 1 and $k$. In particular, the splitting $A=M-N$ with $M=\alpha_{1} P_{1}$ and $N=-\sum_{i=2}^{k} \alpha_{i} P_{i}$ is convergent. It is an open question whether adding more terms to $M$ (that is, using $M=\sum_{i=1}^{r} \alpha_{i} P_{i}$ with $r>1$ ) will result in a smaller spectral radius of $H$ (and thus in faster asymptotic convergence); notice that adding terms to $M$ will make application of the preconditioner more expensive.

Note that the condition (7) is very restrictive. It implies that $\alpha_{1}>1 / 2$, a very strong restriction. It is clear that given a doubly substochastic matrix, in general it will have no representation (2) with $\alpha_{1}>1 / 2$. On the other hand, it is easy to find examples of splittings of doubly substochastic matrices for which $\alpha_{1}=1 / 2$ and the splitting (4) with $r=1$ is not convergent. Also, we have found examples with $k=3, r=2$ and $\alpha_{1}+\alpha_{2}>\alpha_{3}$ where the splitting did not converge. It is an open problem to identify other, less restrictive conditions on the $\alpha_{i}$ (with $1 \leq i \leq r$ ) that will ensure convergence, where the pattern of the permutations could also be used.

### 3.3 Solving linear systems with $M$

The stationary iterations (6) for solving $A x=b$ or Krylov subspace methods using $M$ as a preconditioner require solving linear systems of the form $M z=y$. Assume that $M=\alpha_{1} P_{1}+\alpha_{2} P_{2}$ and $\alpha_{1}>\alpha_{2}$. The stationary iterations $z^{k+1}=\frac{1}{\alpha_{1}} P_{1}^{T}\left(y-\alpha_{2} P_{2} z^{k}\right)$ are convergent for any starting point, with the rate of $\frac{\alpha_{2}}{\alpha_{1}}$. Therefore, if $M=\alpha_{1} P_{1}+\alpha_{2} P_{2}$ and $M$ is as described in Theorem 1(i), then we use the above iterations to apply $M^{-1}$ (that is, solve linear systems with $M$ ). If $M$ is as described in Theorem 4, then we can still solve $M z=y$ by stationary iterations, where we use the splitting $M=\alpha_{1} P_{1}-\sum_{r=2}^{k} \alpha_{r} P_{r}$. We note that application of $\sum_{r=2}^{k} \alpha_{r} P_{r}$ to a vector $y$, that is $z=\left(\sum_{r=2}^{k} \alpha_{r} P_{r}\right) y$ can be effectively computed in $k-1$ steps, where at each step, we perform $z \leftarrow z+\left(\alpha_{r} P_{r}\right) y$. This last operation takes $n$ input entries, scale them and add to $n$ different positions in $z$. As there is no read or write dependencies between these $n$ operations, each step is trivially parallel; especially in shared memory environments, the only parallelization overhead is the creation of threads. Either input can be read in order, or the output can be written in order (by using the inverse of the permutation $P_{r}$ ).

As said before, splitting is guaranteed to work when $\alpha_{1}>1 / 2$. Our experience showed that it does not work in general when $\alpha_{1}<1 / 2$. Therefore, we suggest using $M$ as a preconditioner in Krylov subspace methods. There are two ways to do that. The first is to solve linear systems with $M$ with a direct method by first factorizing $M$. Considering that the number of nonzeros in $M$ would be much less than that of $A$, factorization of $M$ could be much more efficient than factorization of $A$. The second alternative, which we elaborate in Section 3.4, is to build $M$ in such a way that one of the coefficients is larger than the sum of the rest.

### 3.4 Algorithms for constructing the preconditioners

It is desirable to have a small number $k$ in the Birkhoff-von Neumann decomposition while designing the preconditioner. This is because of the fact that if we use splitting, then $k$ determines the number of steps in which we compute the matrix vector products. If we do not use all $k$ permutation matrices, having a few with large coefficients should help to design the preconditioner. The problem of obtaining a Birkhoff-von Neumann decomposition with the minimum number $k$ is a difficult one, as pointed out by Brualdi [5] and has been shown to be NP-complete [15]. This last reference also discusses a heuristic which delivers a small number of permutation matrices. We summarize the heuristic in Algorithm 1.

```
Algorithm 1 A greedy algorithm for obtaining a BvN decomposition.
Input: \(A\), a doubly stochastic matrix.
Output: A BvN decomposition (2) of \(A\) with \(k\) permutation matrices.
    \(k \leftarrow 0\)
    while \(\mathrm{nnz}(A)>0\) do
        \(k \leftarrow k+1\)
        \(P_{k} \leftarrow\) the pattern of a bottleneck perfect matching \(M\) in \(A\)
        \(\alpha_{k} \leftarrow \min M_{i, P_{k}(i)}\)
        \(A \leftarrow A-\alpha_{k} P_{k}\)
    end while
```

As seen in Algorithm 1, the heuristic proceeds step by step, where at each step $j$, a bottleneck matching is found to determine $\alpha_{j}$ and $P_{j}$. A bottleneck matching can be found using MC64 [13, 14]. In this heuristic, at Line $4 M$ is a bottleneck perfect matching; that is, the minimum weight of an edge in $M$ is the maximum among all minimum elements of perfect matchings. Also, at Line 5, $\alpha_{k}$ is equal to the bottleneck value of the perfect matching $M$. A nice property of this heuristic is that it delivers $\alpha_{i}$ s in a non-increasing order; that is $\alpha_{j} \geq \alpha_{j+1}$ for all $1 \leq j<k$. The worst case running time of a step of this heuristic can be bounded by the worst case running time of a bottleneck matching algorithm. For matrices where nnz $(A)=O(n)$, the best known algorithm is of time complexity $O(n \sqrt{n \log n})$ [16]. We direct the reader to [8, p. 185] for other cases.

This heuristic could be used to build an $M$ such that it satisfies the sufficiency condition presented in Theorem 4. That is, we can have $M$ with $\frac{\alpha_{1}}{\sum_{i=1}^{k} \alpha_{i}}>1 / 2$, and hence $M^{-1}$ can be applied with splitting iterations. For this, we start by initializing $M$ to $\alpha_{1} P_{1}$. Then, when the $\alpha_{j}$ where $j \geq 2$ is obtained at Line 5 , we add $\alpha_{j} P_{j}$ to $M$ if $\alpha_{1}$ is still larger than the sum of the other $\alpha_{j}$ included in $M$. In practice, we iterate the while loop until $k$ is around 10 and collect $\alpha_{j}$ 's as described above as long as $\frac{\alpha_{1}}{\sum_{P_{j} \in M^{\alpha}}}>\frac{1}{1.9} \approx 0.53$.

### 3.5 Arbitrary coefficient matrices

Here we discuss how to apply the proposed preconditioning technique to all fully indecomposable sparse matrices.

Let $A \geq 0$ be an $n \times n$ fully indecomposable matrix. The first step is to scale the input matrix with two positive diagonal matrices $R$ and $C$ so that $R A C$ is doubly stochastic, or nearly so. For this step, there are different approaches [20, 21, 24]. Next the linear system $A x=b$ can be solved by solving $R A C x^{\prime}=R b$ and recovering $x=C x^{\prime}$, where we have a doubly stochastic coefficient matrix.

Suppose $A$ is an $n \times n$ fully indecomposable matrix with positive and negative entries. Then, let $B=\operatorname{abs}(A)$ and consider $R$ and $C$ making $R B C$ doubly stochastic, which has a Birkhoff-von Neumann decomposition $R B C=\sum_{i}^{k} \alpha_{i} P_{i}$. Then, $R A C$ can be expressed as

$$
R A C=\sum_{i=1}^{k} \alpha_{i} Q_{i}
$$

where $Q_{i}=\left[q_{j k}^{(i)}\right]_{n \times n}$ is obtained from $P_{i}=\left[p_{j k}^{(i)}\right]_{n \times n}$ as follows:

$$
q_{j k}^{(i)}=\operatorname{sign}\left(a_{j k}\right) p_{j k}^{(i)} .
$$

That is, we can use a convex combination of a set of signed permutation matrices to express any fully indecomposable matrix $A$ where abs $(A)$ is doubly stochastic. We can then use the same construct to use $r$ permutation matrices to define $M$ (for splitting or for defining the preconditioner).

We note that the Theorems 2 and 4 remain valid without changes, since the only property we use is the orthogonality of the $P_{i}$ (not the nonnegativity). Theorem 1 , on the other hand, needs some changes. All we can prove in this more general setting is that if $\alpha_{1} \neq \pm \alpha_{2}$, then $M=\alpha_{1} P_{1}+\alpha_{2} P_{2}$ is nonsingular. We could not find a simple condition for the case $\alpha_{1}= \pm \alpha_{2}$ since we cannot use the Perron-Frobenius Theorem to conclude anything about 1 (or -1 ) being an eigenvalue of $P_{1}^{T} P_{2}$.

## 4 Experiments

We are interested in testing our preconditioner on challenging linear systems that pose difficulties for standard preconditioned Krylov subspace methods. We conducted experiments with the preconditioned GMRES of Matlab, without restart. We also use an implementation of the flexible GMRES (FGMRES) [23] when needed.

We used a large set of matrices which come from three sources. The first set contains 22 matrices which were used by Benzi et al. [3]. Another six matrices were used by Manguoglu et al. [22]; they have experiments with larger matrices but we chose only those with $n \leq 20000$. These matrices are shown in Table 1. All these matrices, except slide, two-dom, watson4a, and watson5a, are available from the UFL Sparse Matrix Collection [10]. To this set, we add all real, square matrices from the UFL Sparse Matrix Collection which contain "chemical" as the keyword. These matrices pose challenges to Krylov subspace methods. There were a total of 70 such matrices; taking the union with the previously described 28 matrices yield 90 matrices in total. Table 1 shows the size and the number of nonzeros of the largest fully indecomposable block (these are the largest square blocks in the Dulmage-Mendelsohn decomposition of the original matrices) of the first two sets of

| matrix | $n_{A}$ | $\mathrm{nnz}_{A}$ | $n$ | nnz | $k$ |
| :--- | ---: | ---: | ---: | ---: | ---: |
| slide | 20191 | 1192535 | 19140 | 1191421 | 900 |
| two-dom | 22200 | 1188152 | 20880 | 1186500 | 938 |
| watson5a | 1854 | 8626 | 1765 | 6387 | 28 |
| watson4a | 468 | 2459 | 364 | 1480 | 134 |
| bp_200 | 822 | 3802 | 40 | 125 | 47 |
| gemat11 | 4929 | 33108 | 4578 | 31425 | 234 |
| gemat12 | 4929 | 33044 | 4552 | 31184 | 260 |
| lns_3937 | 3937 | 25407 | 3558 | 24002 | 205 |
| mahindas | 1258 | 7682 | 589 | 4744 | 722 |
| orani678 | 2529 | 90158 | 1830 | 47823 | 6150 |
| sherman2 | 1080 | 23094 | 870 | 19256 | 334 |
| west0655 | 655 | 2808 | 452 | 1966 | 307 |
| west0989 | 989 | 3518 | 720 | 2604 | 315 |
| west1505 | 1505 | 5414 | 1099 | 3988 | 345 |
| west2021 | 2021 | 7310 | 1500 | 5453 | 376 |
| circuit_3 | 12127 | 48137 | 7607 | 34024 | 322 |
| bayer09 | 3083 | 11767 | 1210 | 6001 | 300 |
| bayer10 | 13436 | 71594 | 10803 | 62238 | 4 |
| lhr01 | 1477 | 18427 | 1171 | 15914 | 473 |
| lhr02 | 2954 | 36875 | 1171 | 15914 | 476 |
| appu | 14000 | 1853104 | 14000 | 1853104 | 1672 |
| raefsky4 | 19779 | 1316789 | 19779 | 1316789 | 774 |
| venkat25 | 62424 | 1717763 | 62424 | 1717763 | 375 |
| utm5940 | 5940 | 83842 | 5794 | 83148 | 291 |
| bundle1 | 10581 | 770811 | 10581 | 770811 | 9508 |
| fp | 7548 | 834222 | 7548 | 834222 | 8008 |
| dw8192 | 8192 | 41746 | 8192 | 41746 | 155 |
| FEM_3D_th | 17880 | 430740 | 17880 | 430740 | 329 |

Table 1 - Test matrices, their original size $n_{A}$ and the size of the largest fully indecomposable block $n$. The experiments were done using the largest fully indecomposable block ( $n$ is the effective size in the following experiments). The last matrix's full name is FEM_3D_thermal1.
matrices. The experiments are conducted with those largest blocks; from now on a matrix refers to its largest block.

In Sections 4.1 and 4.2, we present two sets of experiments: with nonnegative matrices and with general matrices. Each time, we compare three preconditioners. The first is $\operatorname{ILU}(0)$, which forms a basis of comparison. We created ILU(0) preconditioners by first permuting the coefficient matrices to have a maximum product matching in the diagonal (using MC64), as suggested by Benzi et al. [3], and then calling ilu of Matlab with the option nofill. The second set of preconditioners is obtained by a (generalized) BvN decomposition and taking a few different values of " $r$ " in defining the preconditioner. We use $\mathrm{BvN}_{r}$ to denote these preconditioners, where the first $r$ permutation matrices and the corresponding coefficients are taken from the output of Algorithm 1. Since we cannot say anything special about these preconditioners, we use their LU decomposition to apply the preconditioner. For this purpose, we used MATLAB's LU with four output arguments (which permutes the matrices for numerical stability and fill-in). The third set of preconditioners is obtained by the method proposed in Section 3.4 so that the preconditioners satisfy the sufficiency condition of Theorem 4, and hence we can use the splitting based specialized solver. We use $\mathrm{BvN}_{*}$ to denote these preconditioners. In the experiments with the $\operatorname{ILU}(0)$ and $\operatorname{BvN}_{r}$ preconditioners, we used MATLAB's GMRES, whereas with $\mathrm{BvN}_{*}$ we used FGMRES, since we apply BvN ${ }_{*}$ with the specialized splitting based solver. In all cases, we asked a relative residual of $10^{-6}$ from (F)GMRES and run them without restart with at most 3000 iterations. We checked the output of (F)GMRES, and marked those with $f l a g \neq 0$ as unsuccessful. For the successful runs, we checked the relative residual $\frac{\|A x-b\|}{\|b\|}$ for a linear system $A x=b$, and deemed the runs whose relative residual larger than $10^{-4}$ as unsuccessful.

### 4.1 Nonnegative matrices

The first set of experiments are conducted on nonnegative matrices. Let $A$ be a matrix from the data set (Table 1 and another 62 matrices) and $B=\operatorname{abs}(A)$, that is $b_{i j}=\left|a_{i j}\right|$. We scaled $B$ to a doubly stochastic form with the method of Knight and Ruiz [20]. We asked a tolerance of $10^{-8}$ (so that row and column sums can deviate from 1 by $10^{-8}$ ). We then obtained the Birkhoff-von Neumann decomposition by using Algorithm 1. When the bottleneck value found at a step was smaller than $10^{-10}$, we stopped the decomposition process-hence we obtain an "approximate" Birkhoff-von Neumann decomposition of an "approximately" doubly stochastic matrix. This way we obtained $B \approx \alpha_{1} P_{1}+\alpha_{2} P_{2}+\cdots+\alpha_{k} P_{k}$. Then, let $x^{\star}$ be a random vector whose entries are from the uniform distribution on the open interval $(0,1)$, generated using rand of Matlab. We then defined $b=B x^{\star}$ to be the right hand side of $B x=b$.

We report our experiments with GMRES and FGMRES using different permutation matrices on nonnegative matrices in Table 2. The first part of the table gives the number of GMRES iterations with the $\operatorname{ILU}(0)$ preconditioner, and with the proposed Birkhoff-von Neumann based preconditioner, using $r=1,2,4,8,16,32$, and 64 permutation matrices and their coefficients. When there were less than $r$ permutation matrices in the $\operatorname{BvN}$ decomposition, we used all available permutation matrices. Note that this does not mean that we have an exact inverse, as the BvN decomposition is only approximative. We also report results with FGMRES for the $\mathrm{BvN}_{*}$ preconditioners. For this case, we give the number $r$ of permutation matrices used and the number of FGMRES iterations (under the column "it"). In the second part (the last row of the table), we give the number of successful runs with different preconditioners; here we also give the average number of permutation


Figure 1 - Performance profiles for the number of iterations with $\operatorname{ILU}(0), \operatorname{BvN}_{r}$ with $r=2,4,8$, and $\mathrm{BvN}_{*}$.
matrices in $\mathrm{BvN}_{*}$ under the column " $r$ ".
Some observations are in order. As seen in Table 2, $\operatorname{ILU}(0)$ results in 11 successful runs on the matrices listed in the first part of the table, and 32 successful runs in all 90 matrices. $\mathrm{BvN}_{r}$ preconditioners with differing number of permutation matrices result in at least 53 successful runs in total, where $\mathrm{BvN}_{*}$ obtained the highest number of successful runs, but it has in only a few cases the least number of iterations. In 20 cases in the first part of the table, we see that adding more permutation matrices usually helps in reducing the number of iterations. This is not always the case though. We think that this is due to the preconditioner becoming badly conditioned with the increasing number of permutation matrices; if we use the full BvN decomposition, we will have the same conditioning as in $A$.

In order to clarify more, we present the performance profiles [11] for $\operatorname{ILU}(0), \mathrm{BvN}_{r}$ with $r=$ $2,4,8$, and $\mathrm{BvN}_{*}$ in Fig. 1a. A performance profile for a preconditioner shows the fraction of the test cases in which the number of (F)GMRES iterations with the preconditioner are within $\tau$ times the smallest number of (F)GMRES iterations observed (with the mentioned set of preconditioners). Therefore, the higher the profile of a preconditioner, the better is its performance. As seen in this figure, $\mathrm{BvN}_{*}$ and $\mathrm{BvN}_{8}$ manifest themselves as the best preconditioners; they are better than others after around $\tau=1.1$, while $\mathrm{BvN}_{8}$ being always better than others. $\mathrm{BvN}_{*}$ is a little behind $\operatorname{ILU}(0)$ at the beginning but then catches up with $\mathrm{BvN}_{8}$ at around 1.2 and finishes as the best alternative.

Although we are mostly concerned with robustness of the proposed preconditioners and their potential for parallel computing, we give a few running time results on a sequential Matlab environment (on a core of an Intel Xeon E5-2695 with 2.30 GHz clock speed). We present the complexity of the preconditioners and the running time of (F)GMRES (total time spent in the iterations) for the cases where $\operatorname{ILU}(0)$ resulted in convergence in Table 3. This set of matrices are chosen to give the running time of (F)GMRES with all preconditioners under study. Otherwise, it is not very meaningful to use sophisticated preconditioners when $\operatorname{ILU}(0)$ is effective. Furthermore, the applications of $\operatorname{ILU}(0)$ and $\mathrm{BvN}_{r}$ require triangular solves, which are efficiently implemented in MATLAB. On the other hand, the application of $\mathrm{BvN}_{*}$ requires permutations and scaling, which should be very efficient in parallel. The running time given in the right side of Table 3 should not be taken at face

| matrix | ILU(0) | $\mathrm{BvN}_{r}$ (iterations) |  |  |  |  |  |  | $\mathrm{BvN}_{*}$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 1 | 2 | 4 | 8 | 16 | 32 | 64 | $r$ | it. |
| slide | - | 1102 | 1158 | 946 | 787 | 567 | 566 | 131 | 10 | 884 |
| two-dom | 1328 | - | - | - | 2774 | 2088 | 1719 | 1429 | - | - |
| watson5a | - | 453 | 516 | 610 | 806 | 908 | 908 | 908 | 5 | 621 |
| watson4a | 49 | 146 | 133 | 139 | 124 | 93 | 32 | 5 | 12 | 129 |
| bp_200 | - | 36 | 26 | 25 | 12 | 7 | 3 | 3 | 8 | 25 |
| gemat11 | 176 | - | - | 2857 | 2118 | 1406 | 664 | 169 | - | - |
| gemat12 | 263 | - | - | 2906 | 2237 | 1591 | - | 145 | - | - |
| lns_3937 | 349 | 2067 | 1286 | 666 | 229 | 56 | 14 | 5 | 10 | 1138 |
| mahindas | - | 303 | 283 | 245 | 173 | 87 | 40 | 21 | 7 | 278 |
| orani678 | - | 371 | 360 | 334 | 303 | 283 | 264 | 221 | 5 | 335 |
| sherman2 | 19 | 324 | 238 | 157 | 96 | 51 | 21 | 7 | 9 | 235 |
| west0655 | 71 | 331 | 275 | 188 | 139 | - | - | 31 | 9 | 270 |
| west0989 | - | 194 | 167 | 114 | 63 | 35 | 19 | 9 | 8 | 165 |
| west1505 | - | 335 | 254 | 166 | 91 | 49 | 27 | 10 | 9 | 241 |
| west2021 | - | 406 | 323 | 217 | - | 47 | 34 | 13 | 8 | 315 |
| circuit_3 | - | - | - | - | - | - | - | 1426 | - | - |
| bayer09 | 124 | 521 | 434 | 291 | 204 | 110 | 28 | 7 | 7 | 393 |
| bayer10 | - | - | - | - | - | - | - | - | - | - |
| lhr01 | - | 574 | 350 | 216 | 130 | 86 | 42 | 18 | 7 | 313 |
| lhr02 | - | 565 | 351 | 219 | 136 | 87 | 45 | 19 | 8 | 314 |
| appu | 31 | 50 | 50 | 52 | 56 | 62 | 74 | 82 | 7 | 50 |
| raefsky4 | - | - | 2728 | 2735 | 2444 | 1559 | 1739 | 1230 | 6 | 2370 |
| venkat25 | - | - | - | - | - | - | - | - | - | - |
| utm5940 | - | 2633 | 2115 | - | - | - | 153 | 34 | 7 | 2069 |
| bundle1 | 25 | 163 | 150 | 139 | 149 | 174 | 158 | 139 | 5 | 149 |
| fp | - | - | - | - | - | - | - | - | - | - |
| dw8192 | - | - | - | 2398 | 1516 | 650 | 81 | 8 | - | - |
| FEM_3D_th | 7 | 37 | 35 | 38 | 72 | 58 | 25 | 6 | 9 | 33 |
| Number of successful runs with all 90 matrices |  |  |  |  |  |  |  |  |  |  |
|  | 32 | 58 | 71 | 60 | 57 | 53 | 58 | 61 | 7 | 77 |

Table 2 - The number of GMRES iterations with $\operatorname{ILU}(0)$ and $\mathrm{BvN}_{r}$ with different number of permutation matrices, and the number of permutation matrices and FGMRES iterations with $\mathrm{BvN}_{*}$. (F)GMRES are run with tolerance $10^{-6}$, without restart and with at most 3000 iterations. The symbol "-" in the table flags the cases where (F)GMRES were unsuccessful. All matrices are nonnegative.

|  | Complexity of $M$ |  |  |  | Running time |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $\mathrm{BvN}_{r}$ |  |  | $\mathrm{BvN}_{*}$ | ILU(0) | $\mathrm{BvN}_{r}$ |  |  | $\mathrm{BvN}_{*}$ |
|  | 2 | 4 | 8 |  |  | 2 | 4 | 8 |  |
| two-dom | 0.03 | 0.04 | 1.33 | 0.08 | 3219.99 | - | - | 6108.59 | - |
| watson4a | 0.28 | 0.36 | 0.44 | 0.66 | 0.06 | 0.24 | 0.26 | 0.21 | 6.99 |
| gemat11 | 0.23 | 0.35 | 0.65 | 0.61 | 8.31 | - | 851.16 | 458.18 | - |
| gemat12 | 0.21 | 0.31 | 0.58 | 0.61 | 10.93 | - | 825.32 | 492.63 | - |
| lns_3937 | 0.29 | 0.55 | 1.67 | 0.63 | 27.28 | 131.73 | 31.42 | 3.84 | 519.83 |
| sherman2 | 0.06 | 0.10 | 0.14 | 0.26 | 0.02 | 0.83 | 0.39 | 0.16 | 20.13 |
| west0655 | 0.42 | 0.69 | 1.11 | 0.62 | 0.08 | 0.94 | 0.46 | 0.26 | 16.84 |
| bayer09 | 0.32 | 0.52 | 0.91 | 0.51 | 0.41 | 2.89 | 1.34 | 0.70 | 42.96 |
| appu | 0.01 | 0.02 | 7.05 | 0.04 | 1.59 | 1.07 | 1.18 | 3.13 | 31.99 |
| bundle1 | 0.01 | 0.01 | 0.02 | 0.03 | 0.94 | 4.70 | 4.13 | 4.31 | 88.82 |
| FEM_3D_th | 0.05 | 0.08 | 0.78 | 0.27 | 0.36 | 0.73 | 0.83 | 2.23 | 23.40 |

Table 3 - The complexity of the preconditioners and the running time of the solver. The complexities of $\mathrm{BvN}_{r}$ are given as the ratio $(\mathrm{nnz}(L+U)-n) / \mathrm{nnz}$, where $L$ and $U$ are the factors of the preconditioner $(\operatorname{ILU}(0)$ has a complexity of 1.0$)$, and that of $\mathrm{BvN}_{*}$ is given as nnz $(M) / \mathrm{nnz}(A)$. A sign of "-" in the running time column flags the cases where (F)GMRES were unsuccessful. All matrices are nonnegative.
value. The complexities of $\mathrm{BvN}_{r}$ are given as the ratio (nnz $\left.(L+U)-n\right) / \mathrm{nnz}(A)$, where $L$ and $U$ are the factors of the preconditioner. In this setting, $\operatorname{ILU}(0)$ has always a complexity of 1.0. The complexity of $\mathrm{BvN}_{*}$ is given as $\mathrm{nnz}(M) / \mathrm{nnz}(A)$. As seen in this table, the LU-factors of $\mathrm{BvN}_{r}$ with $r \leq 8$ have reasonable number of nonzeros; the complexity of the preconditioners are less than one in all cases except two-dom, lns_3937, west0655, and appu. In some preliminary experiments, we have seen large numbers when $r \geq 16$. This was especially important for appu where the full LU factorization of $A$ contained $85.67 \cdot \mathrm{nnz}(A)$ nonzeros and $M$ suffered large fill-in. We run the solver with a large error tolerance of $1.0 \mathrm{e}-1$, as this was enough to get the outer FGMRES to converge. The outer FGMRES iterations would change if a lower error tolerance is used in the inner solver, but we do not dwell into this issue here. With the specified parameters, an application of $\mathrm{BvN}_{*}$ required, for the data shown in Table 3, between 128 and 162 iterations.

We now comment on the running time of the preconditioner set up phase. The first step is to apply the scaling method of Knight and Ruiz [20]. The dominant cost in this step is sparse matrixvector and sparse matrix transpose-vector multiply operations. The second step is to obtain a BvN decomposition, or a partial one, using the algorithms for the bottleneck matching problem from MC64 [13, 14]; these algorithms are highly efficient. The number of sparse matrix-vector and sparse matrix transpose-vector multiply operations (MVP) in the scaling algorithm, the running time of the scaling algorithm with the previous setting, and the running time of the partial BvN decomposition with 16 permutation matrices are shown for the same set of matrices in Table 4. Since an application of $\mathrm{BvN}_{*}$ required between 128 and 162 iterations, and those iterations are no more costly then sparse matrix vector operations, we could relate the running time of FGMRES with $\mathrm{BvN}_{*}$ in Table 3 to the scaling algorithm. For example, for $\operatorname{lns} 3937$ a matrix vector multiplication should not cost more than $0.27 / 3732$ seconds, while the application of $\mathrm{BvN}_{*}$ required a total of $1138 \cdot 152$ steps of inner solver. In principle the total cost of the inner solver should be around 12.5 , but in Table3, we see 519.83 seconds of total FGMRES time. As seen in Table 4, for these set of matrices the scaling algorithm and the partial BvN decomposition algorithms are fast. There

|  |  | running time (.s) |  |
| :--- | ---: | ---: | ---: |
| matrix | MVP | Scaling | BvN |
| two-dom | 438 | 0.67 | 3.03 |
| watson4a | 40368 | 0.44 | 0.00 |
| gemat11 | 2286 | 0.28 | 0.04 |
| gemat12 | 843216 | 1.01 | 0.04 |
| lns_3937 | 3732 | 0.27 | 0.04 |
| sherman2 | 1966 | 0.06 | 0.01 |
| west0655 | 1846 | 0.03 | 0.00 |
| bayer09 | 3626 | 0.10 | 0.01 |
| appu | 46 | 0.15 | 1.57 |
| bundle1 | 116 | 0.13 | 0.78 |
| FEM_3D_th | 52 | 0.04 | 0.81 |

Table 4 - The total number of sparse matrix-vector and sparse matrix transpose-vector multiply operations (MVP) in the scaling algorithm, the running time of the scaling algorithm and the partial BvN decomposition algorithm in seconds.
are efficient parallel scaling algorithms [1], and the one that we used [20] can also be efficiently parallelized by tapping into the parallel sparse matrix vector multiplication algorithms.

### 4.2 General matrices

In this set of experiments, we used the matrices listed before, while retaining the sign of the nonzero entries. We used the same scaling algorithm as in the previous section and the proposed generalized BvN decomposition (Section 3.5) to construct the preconditioners. As in the previous subsection, we report our experiments with GMRES and FGMRES using different permutation matrices in Table 5. In particular, we report the number of GMRES iterations with the $\operatorname{ILU}(0)$ preconditioner, and with the proposed Birkhoff-von Neumann based preconditioner, using $r=1,2,4,8,16,32$, and 64 permutation matrices and their coefficients. We also report results with FGMRES for the $\mathrm{BvN}_{*}$ preconditioners. Again, the number $r$ of permutation matrices used and the number of FGMRES iterations (under the column "it.") are given for $\mathrm{BvN}_{*}$. The last row of the table gives the number of successful runs with different preconditioners in all 90 matrices; here the average number of permutation matrices in $\mathrm{BvN}_{*}$ is also given under the column " $r$ ". We again use the performance profiles shown in Fig. 1b to see the effects of the preconditioners. As seen in this table and the associated performance profile, the preconditioners behave much like they do in the nonnegative case. In particular, $\mathrm{BvN}_{*}$ and $\mathrm{BvN}_{8}$ deliver the best performance in terms of number of iterations, followed by ILU(0).

We also give the complexity of the preconditioners and the running time of (F)GMRES (total time spent in the iterations) for the cases where $\operatorname{ILU}(0)$ resulted in convergence in Table 6. The complexities of the preconditioners remain virtually the same, as expected. As before, the running time is just to give a rough idea in a sequential MATLAB environment. We note that $\mathrm{BvN}_{*}$ required, on average, between 125 and 151 inner iterations. Finally note that nothing changes for the scaling and BvN decomposition algorithms with respect to the nonnegative case (Table 4 remains as is for the general case).

| matrix | $\mathrm{ILU}(0)$ | $\mathrm{BvN}_{r}$ it. |  |  |  |  |  |  | $\mathrm{BvN}_{*}$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 1 | 2 | 4 | 8 | 16 | 32 | 64 | $r$ | it. |
| slide | 208 | 1051 | 999 | 745 | 685 | 490 | 783 | 659 | 10 | 663 |
| two-dom | 149 | 579 | 553 | - | 610 | 607 | 432 | 360 | 7 | 573 |
| watson5a | - | 805 | 839 | 883 | 1094 | 1068 | 1068 | 1068 | 5 | 947 |
| watson4a | 48 | 160 | 149 | 135 | 120 | 81 | 34 | 5 | 12 | 124 |
| bp_200 | - | 35 | 26 | 22 | 11 | 6 | 3 | 3 | 8 | 24 |
| gemat11 | 239 | - | - | - | 2486 | 1588 | 750 | 152 | - | - |
| gemat12 | 344 | - | - | - | 2607 | 1615 | - | - | - | - |
| lns_3937 | 134 | 1710 | 969 | - | 121 | 23 | 9 | 4 | 10 | 879 |
| mahindas | - | 259 | 232 | 180 | 121 | 51 | - | 12 | 7 | 232 |
| orani678 | - | 356 | 341 | 320 | 288 | - | 225 | 196 | 5 | 316 |
| sherman2 | 18 | 327 | 234 | 116 | 65 | 31 | 16 | 8 | 9 | 225 |
| west0655 | 50 | 298 | 229 | 164 | 101 | 64 | 30 | - | 9 | 222 |
| west0989 | - | 199 | 165 | 113 | 63 | 37 | 19 | 8 | 8 | 166 |
| west1505 | - | 343 | 262 | 170 | 92 | 49 | 27 | 12 | 9 | 256 |
| west2021 | - | 411 | 323 | 225 | 115 | 48 | 28 | 12 | 8 | 316 |
| circuit_3 | - | - | - | - | - | - | - | 1178 | - | - |
| bayer $9^{-9}$ | 39 | 312 | 193 | 105 | 58 | - | 11 | 5 | 7 | 213 |
| bayer10 | - | - | 2624 | 2517 | 2517 | 2517 | 2517 | 2517 |  | 2594 |
| lhr01 | - | 495 | 314 | - | 117 | 65 | 28 | 14 | 7 | 279 |
| lhr02 | - | 508 | 315 | - | 119 | 67 | 28 | 15 | 8 | 277 |
| appu | 31 | 50 | 50 | 52 | 56 | 62 | 74 | 82 | 7 | 51 |
| raefsky 4 | 499 | 1207 | 722 | 472 | 592 | 403 | 860 | 617 | 6 | 656 |
| venkat25 | 163 | - | - | 1863 | - | - | 258 | - | 8 | 2493 |
| utm5940 | - | 1972 | 1588 | 1173 | 787 | 536 | 196 | 34 | 7 | 1520 |
| bundle1 | 18 | 130 | 123 | 115 | 127 | 137 | 129 | 102 | 5 | 125 |
| fp | - | - | 2953 | - | - | - | 120 | 185 | 6 | 2779 |
| dw8192 | - | - | - | 2450 | 1536 | 661 | 82 | 11 | - | - |
| FEM_3D_th | 7 | 42 | 40 | 40 | 74 | 62 | 27 | 6 | 9 | 38 |
| Number of successful runs with all 90 matrices |  |  |  |  |  |  |  |  |  |  |
|  | 35 | 65 | 72 | 51 | 57 | 53 | 56 | 58 | 7 | 82 |

Table 5 - The number of GMRES iterations with different number of permutation matrices in $M$ with tolerance $10^{-6}$. We ran GMRES (without restart) for at most 3000 iterations; the symbol "-" in the Table flags cases where GMRES did not converge to the required tolerance. Matrices have negative and positive entries.

|  | Complexity of $M$ |  |  |  | Running time |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $\mathrm{BvN}_{r}$ |  |  | $\mathrm{BvN}_{*}$ | ILU(0) | $\mathrm{BvN}_{r}$ |  |  | $\mathrm{BvN}_{*}$ |
|  | 2 | 4 | 8 |  |  | 2 | 4 | 8 |  |
| slide | 0.03 | 0.05 | 4.04 | 0.10 | 38.67 | 300.65 | 176.02 | 148.79 | 1017.34 |
| two-dom | 0.03 | 0.04 | 1.31 | 0.08 | 56.40 | 183.60 | - | 205.53 | 1122.33 |
| watson4a | 0.28 | 0.36 | 0.44 | 0.66 | 0.05 | 0.28 | 0.24 | 0.19 | 6.58 |
| gemat11 | 0.23 | 0.35 | 0.65 | 0.61 | 16.57 |  |  | 1548.86 | - |
| gemat12 | 0.21 | 0.31 | 0.58 | 0.61 | 34.15 | - | - | 1578.89 | - |
| lns_3937 | 0.29 | 0.55 | 1.66 | 0.63 | 4.68 | 186.25 | - | 3.12 | 359.80 |
| sherman2 | 0.06 | 0.10 | 0.14 | 0.26 | 0.02 | 0.80 | 0.25 | 0.09 | 18.03 |
| west0655 | 0.42 | 0.69 | 1.08 | 0.62 | 0.06 | 0.65 | 0.38 | 0.15 | 13.53 |
| bayer09 | 0.32 | 0.52 | 0.90 | 0.51 | 0.05 | 0.62 | 0.21 | 0.08 | 21.12 |
| appu | 0.01 | 0.02 | 7.05 | 0.04 | 1.64 | 2.06 | 2.16 | 4.64 | 32.71 |
| raefsky4 | 0.02 | 0.02 | 0.29 | 0.07 | 206.02 | 262.48 | 116.39 | 182.10 | 1027.58 |
| venkat25 | 0.05 | 0.53 | 3.61 | 0.23 | 94.29 | - | 5462.64 | - | 20382.59 |
| bundle1 | 0.01 | 0.01 | 0.02 | 0.03 | 0.50 | 4.81 | 4.84 | 5.46 | 76.47 |
| FEM_3D_th | 0.05 | 0.08 | 0.78 | 0.27 | 0.34 | 1.10 | 1.10 | 3.11 | 30.07 |

Table 6 - The complexity of the preconditioners and the running time of the solver. The complexities of $\mathrm{BvN}_{r}$ are given as the ratio $(\mathrm{nnz}(L+U)-n) / \mathrm{nnz}$, where $L$ and $U$ are the factors of the preconditioner $(\operatorname{ILU}(0)$ has a complexity of 1.0$)$, and that of $\mathrm{BvN}_{*}$ is given as $\mathrm{nnz}(M) / \mathrm{nnz}(A)$. A sign of "-" in the running time column flags the cases where (F)GMRES were unsuccessful. Matrices have negative and positive entries.

### 4.3 Further investigations

Here, we give some further experiments to shed light into the behavior of the proposed preconditioners. We compare $\mathrm{BvN}_{*}$ with $\mathrm{BvN}_{r}$ having the same number of permutation matrices. For this purpose, we created a $\mathrm{BvN}_{*}$ preconditioner and used the number $r$ of its permutation matrices to create another preconditioner $\mathrm{BvN}_{r}$ by just taking the first $r$ permutation matrices from a BvN decomposition obtained by Algorithm 1. The results are shown in Table 7, where we give the sum of the coefficients used in constructing the preconditioners and the number of (F)GMRES iterations for a subset of matrices from Table 6 , where $\mathrm{BvN}_{*}$ led to a short running time.

As seen in Table $7, \mathrm{BvN}_{r}$ has a larger sum of coefficients than $\mathrm{BvN}_{*}$ with the same number $r$ of permutation matrices. This is expected as the Algorithm 1 finds coefficients in a non-increasing order; hence the first $r$ coefficients are the largest $r$ ones. The number of (F)GMRES iterations is usually inversely proportional to the sum of the coefficients; a higher sum of coefficients usually results in a smaller number of iterations for the same matrix.

## 5 Conclusions and open questions

We introduced a class of preconditioners for general sparse matrices based on the Birkhoff-von Neumann decomposition of doubly stochastic matrices. These preconditioners are aimed primarily at solving challenging linear systems with highly unstructured and indefinite coefficient matrices in parallel computing environments. We presented some theoretical results and numerical experiments on linear systems from a variety of applications. We regard this work to be a proof of concept realization; many challenging questions remain to be investigated to render the proposed

| matrix | $\sum \alpha_{i}$ |  | it. |  |
| :--- | ---: | ---: | ---: | ---: |
|  | $\mathrm{BvN}_{*}$ | $\mathrm{BvN}_{r}$ | $\mathrm{BvN}_{*}$ | $\mathrm{BvN}_{r}$ |
| watson4a | 0.43 | 0.57 | 124 | 111 |
| lns_3937 | 0.29 | 0.80 | 881 | 68 |
| sherman2 | 0.24 | 0.61 | 226 | 59 |
| west0655 | 0.12 | 0.45 | 222 | 97 |
| bayer09 | 0.19 | 0.50 | 216 | 59 |
| appu | 0.09 | 0.13 | 50 | 55 |
| bundle1 | 0.01 | 0.01 | 125 | 110 |
| FEM_3D_th | 0.35 | 0.55 | 39 | 79 |

Table 7 - Comparing $\mathrm{BvN}_{*}$ with $\mathrm{BvN}_{r}$ having the same number of permutation matrices in terms of the sum of the coefficient of permutation matrices and the number of (F)GMRES iterations (it.). Matrices have negative and positive entries.
preconditioners competitive with the standard approaches.
Based on our current theoretical findings, we suggest the use of proposed preconditioners within a Krylov subspace method. There are two ways to go about this. In the first one, the preconditioners are built to satisfy a sufficiency condition that we identified (Theorem 4). This way, the application of the preconditioner requires a small number of highly concurrent steps, where there is no data dependency within a step. In the second alternative, one obtains the LU decomposition of the preconditioners that are built arbitrarily. Here, the proposed preconditioner is therefore constructed as a complete factorization of an incomplete matrix. We demonstrated that using around eight matrices is good enough for this purpose. Beyond that number, the LU decomposition of the preconditioner can become a bottleneck (but remains always cheaper than that of the original matrix). Is there a special way to order these preconditioners for smaller fill-in?

The construction of the preconditioners needs an efficient doubly stochastic scaling algorithm. The known algorithms for this purpose are iterative schemes whose computational cores are sparse matrix-vector multiply operations whose efficient parallelization is well known. For constructing these preconditioner, we then need a bottleneck matching algorithm. The exact algorithms for this purpose are efficient on sequential execution environments, but hard to parallelize efficiently. There are efficiently parallel heuristics for matching problems [18], and more research is needed to parallelize the heuristic for obtaining a BvN decomposition while keeping an eye on the quality of the preconditioner.

## Acknowledgements

The work of Michele Benzi was supported in part by NSF grant DMS-1418889. Bora Uçar was supported in part by French National Research Agency (ANR) project SOLHAR (ANR-13-MONU0007). We thank Alex Pothen for his contributions to this work. This work resulted from the collaborative environment offered by the Dagstuhl Seminar 14461 on High-performance Graph Algorithms and Applications in Computational Science (November 9-14, 2014).

## References

[1] P. R. Amestoy, I. S. Duff, D. Ruiz, and B. U̧̧ar, A parallel matrix scaling algorithm, in VECPAR 2008, J. M. Palma, P. R. Amestoy, M. Daydé, M. Mattoso, and J. C. Lopes, eds., Springer Berlin Heidelberg, 2008, pp. 301-313.
[2] H. Anzt, E. Chow, and J. Dongarra, Iterative sparse triangular solves for preconditioning, in Euro-Par 2015, J. L. Träff, S. Hunold, and F. Versaci, eds., Springer Berlin Heidelberg, 2015, pp. 650-661.
[3] M. Benzi, J. C. Haws, and M. Tuma, Preconditioning highly indefinite and nonsymmetric matrices, SIAM Journal on Scientific Computing, 22 (2000), pp. 1333-1353.
[4] G. Birkhoff, Tres observaciones sobre el algebra lineal, Univ. Nac. Tucumán Rev. Ser. A, (1946), pp. 147-150.
[5] R. A. Brualdi, Notes on the Birkhoff algorithm for doubly stochastic matrices, Canadian Mathematical Bulletin, 25 (1982), pp. 191-199.
[6] R. A. Brualdi and P. M. Gibson, Convex polyhedra of doubly stochastic matrices: I. Applications of the permanent function, Journal of Combinatorial Theory, Series A, 22 (1977), pp. 194-230.
[7] R. A. Brualdi and H. J. Ryser, Combinatorial Matrix Theory, vol. 39 of Encyclopedia of Mathematics and its Applications, Cambridge University Press, Cambridge, UK; New York, USA; Melbourne, Australia, 1991.
[8] R. Burkard, M. Dell'Amico, and S. Martello, Assignment Problems, SIAM, Philadelphia, PA, USA, 2009.
[9] E. Chow and A. Patel, Fine-grained parallel incomplete LU factorization, SIAM Journal on Scientific Computing, 37 (2015), pp. C169-C193.
[10] T. A. Davis and Y. Hu, The University of Florida sparse matrix collection, ACM Transactions on Mathematical Software, 38 (2011), 1:1-1:25.
[11] E. D. Dolan and J. J. Moré, Benchmarking optimization software with performance profiles, Mathematical Programmonf, 91 (2002), pp. 201-213.
[12] I. S. Duff, A. M. Erisman, and J. K. Reid, Direct Methods for Sparse Matrices, Oxford University Press, London, 1986.
[13] I. S. Duff and J. Koster, The design and use of algorithms for permuting large entries to the diagonal of sparse matrices, SIAM Journal on Matrix Analysis and Applications, 20 (1999), pp. 889-901.
[14] I. S. Duff and J. Koster, On algorithms for permuting large entries to the diagonal of a sparse matrix, SIAM Journal on Matrix Analysis and Applications, 22 (2001), pp. 973-996.
[15] F. Dufossé and B. UÇar, Notes on Birkhoff-von Neumann decomposition of doubly stochastic matrices, Linear Algebra and its Applications, 497 (2016), pp. 108-115.
[16] H. N. Gabow and R. E. Tarjan, Algorithms for two bottleneck optimization problems, J. Algorithms, 9 (1988), pp. 411-417.
[17] F. R. Gantmacher, The Theory of Matrices, vol. 2, Chelsea Publishing Co.,, New York, NY, 1959.
[18] M. Halappanavar, A. Pothen, A. Azad, F. Manne, J. Langquth, A. M. Khan, IEEE Computer, 48 (2015), pp. 46-55.
[19] R. A. Horn and C. R. Johnson, Matrix Analysis, Cambridge University Press, second ed., 2013.
[20] P. A. Knight and D. Ruiz, A fast algorithm for matrix balancing, IMA Journal of Numerical Analysis, 33 (2013), pp. 1029-1047.
[21] P. A. Knight, D. Ruiz, and B. Uçar, A symmetry preserving algorithm for matrix scaling, SIAM Journal on Matrix Analysis and Applications, 35 (2014), pp. 931-955.
[22] M. Manguoglu, M. Koyutürk, A. H. Sameh, and A. Grama, Weighted matrix ordering and parallel banded preconditioners for iterative linear system solvers, SIAM Journal on Scientific Computing, 32 (2010), pp. 1201-1216.
[23] Y. SAAD, A flexible inner-outer preconditioned GMRES algorithm, SIAM Journal on Scientific Computing, 14 (1993), pp. 461-469.
[24] R. Sinkhorn and P. Knopp, Concerning nonnegative matrices and doubly stochastic matrices, Pacific J. Math., 21 (1967), pp. 343-348.
[25] R. S. Varga, Matrix Iterative Analysis, Springer, Berlin, Heidelberg, New York, second (revised and expanded edition of Prentice-Hall, Englewood Cliffs, New Jersey, 1962) ed., 2000

RESEARCH CENTRE
GRENOBLE - RHÔNE-ALPES
Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105-78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399


[^0]:    * Department of Mathematics and Computer Science, Emory University, Atlanta, USA (benzi@mathcs.emory.edu).
    $\dagger$ CNRS and LIP (UMR5668 CNRS-ENS Lyon-INRIA-UCBL), 46, allée d'Italie, ENS Lyon, Lyon F-69364, France.

