Orthogonal matching pursuit-based algorithms for the Birkhoff-von Neumann decomposition

Damien Lesens, Jérémy E Cohen, Bora Uçar

Orthogonal matching pursuit-based algorithms for the Birkhoff–von Neumann decomposition

Damien Lesens, Jérémy E. Cohen, Bora Uçar

RESEARCH REPORT
N° 9543
March 2024
Project-Team ROMA

ISSN 0249-6399   ISRN INRIA/RR--9543--FR+ENG

Orthogonal matching pursuit-based algorithms for the Birkhoff–von Neumann decomposition

Damien Lesens*, Jérémy E. Cohen†, Bora Uçar‡

Project-Team ROMA

Research Report n° 9543 — March 2024 — 12 pages

Abstract: Birkhoff-von Neumann (BvN) decomposition writes a doubly stochastic matrix as a convex combination of permutation matrices. For a given doubly stochastic matrix, the decomposition in general is not unique. In many applications a sparsest decomposition, that is with the smallest number of permutation matrices is of interest. This problem is known to be NP-complete, and heuristics are used to obtain sparse solutions. We propose heuristics based on the well-known orthogonal matching pursuit for sparse BvN decomposition. We experimentally compare our heuristics with the state of the art from the literature and show how our methods advance the known heuristics.

Key-words: Sparse Coding, Birkhoff-von Neumann Decomposition, Orthogonal Matching Pursuit

* Univ Lyon, CNRS, ENS de Lyon, Inria, Université Claude Bernard Lyon 1, LIP UMR 5668, F-69007 LYON, France

† Univ Lyon, INSA-Lyon, UCBL, UJM, CNRS, Inserm, CREATIS UMR 5220, U1294, F-69621, Lyon, France

‡ Univ Lyon, CNRS, ENS de Lyon, Inria, Université Claude Bernard Lyon 1, LIP UMR 5668, F-69007 LYON, France

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Algorithmes basés sur l'OMP pour la décomposition de Birkhoff-von Neumann

Résumé : La décomposition de Birkhoff-von Neumann (BvN) écrit une matrice doublement stochastique comme une combinaison convexe de matrices de permutation. Pour une matrice doublement stochastique donnée, la décomposition n'est généralement pas unique. Dans de nombreuses applications, il est intéressant d'obtenir la décomposition la plus parcimonieuse, c'est-à-dire avec le plus petit nombre de matrices de permutation. Ce problème est connu pour être NP-complet, et des heuristiques sont utilisées pour obtenir des décompositions assez parcimonieuses. Nous proposons des heuristiques basées sur la célèbre méthode "Orthogonal Matching Pursuit" pour la décomposition BvN clairsemée. Nous comparons expérimentalement nos heuristiques avec l'état de l'art de la littérature et montrons comment nos méthodes font progresser les heuristiques connues.

Mots-clés : Approximation parcimonieuse, decomposition de Birkhoff-von Neumann, Orthogonal Matching Pursuit

Contents

1 Introduction

A square matrix is called doubly stochastic if it has nonnegative entries, and the sum of entries in each row and in each column is one. Birkhoff Theorem states that any doubly stochastic matrix can be written as a convex combination of permutation matrices [3]. That is, for an $n \times n$ doubly stochastic matrix $\mathbf{A}$, there exist coefficients $\alpha_1, \alpha_2, \ldots, \alpha_k \in (0, 1]$ with $\sum_{i=1}^{k} \alpha_i = 1$ and $n \times n$ permutation matrices $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_k$ such that

$$\mathbf{A} = \alpha_1 \mathbf{P}_1 + \alpha_2 \mathbf{P}_2 + \cdots + \alpha_k \mathbf{P}_k \, . \tag{1}$$

This representation is called Birkhoff-von Neumann (BvN) decomposition. A given doubly stochastic matrix in general has multiple BvN decompositions, and there are various applications in which a decomposition with a small number of permutation matrices is required [1, 7, 16, 17]. Dufossé and Uçar [10] show that the problem of finding a BvN decomposition of a given matrix with the minimum number of permutation matrices is NP-complete.

Heuristics approaches to obtaining BvN decompositions with small number of permutation matrices have been proposed and analysed [10, 11]. According to the results reported in these papers, a greedy approach called BvNG performs very well in general but has limitations. In particular, BvNG obtains a much smaller number of permutation matrices than the known algorithm resulting from Birkhoff's proof on matrices arising in various applications, in many cases close to a lower bound. More importantly there are optimal BvN decompositions which neither BvNG nor its generalizations can reach. Our aim in this paper is to design a set of heuristics based on the well-known greedy orthogonal matching pursuit algorithm with the following characteristics:

- they obtain optimal solutions on some instances in which BvNG cannot access to optimal solutions;
- they perform as good as BvNG on problem instances in which BvNG performs well, while improving it on hard cases.

The work presented here lies at the intersection of graph algorithms, matrix computations, and the sparse coding problem. We give a brief, required background in these areas in the next section. Then, in Section 3 we present the heuristics. Section 4 contains experiments to document the characteristics listed above, before concluding the paper in Section 5 with a summary and future work.

2 Background and notation

We use bold upper-case letters to refer to matrices, and bold lower-case letters to refer to vectors. We use $\mathrm{vec}(\cdot)$ to convert a matrix to a vector, by stacking the columns, e.g., for an $n \times n$ matrix $\mathbf{A}$, the vector $\mathbf{a} = \mathrm{vec}(\mathbf{A})$ is of size n^2.

A *permutation matrix* is a square matrix consisting of zeros and ones, with exactly a single one in each column and in each row. Let $\mathbf{A}$ be an $n \times n$ matrix and $\mathbf{P}$ be an $n \times n$ permutation matrix. We use the notation $\mathbf{P} \subseteq \mathbf{A}$ to denote that the entries of $\mathbf{A}$ at the positions corresponding to the nonzero entries of $\mathbf{P}$ are also nonzero. That is, the nonzero pattern of $\mathbf{P}$ is included in the nonzero pattern of $\mathbf{A}$. We use $\mathbf{P} \odot \mathbf{A}$ to denote the entry-wise product of $\mathbf{P}$ and $\mathbf{A}$, which selects the entries of $\mathbf{A}$ at the positions corresponding to the nonzero entries of $\mathbf{P}$. We also use $\min\{\mathbf{P} \odot \mathbf{A}\}$ to denote the minimum entry of $\mathbf{A}$ at the nonzero positions of $\mathbf{P}$.

A square matrix $\mathbf{A}$ is called *fully indecomposable*, when $\mathbf{AQ}$ has nonzeros on the diagonal for a permutation matrix $\mathbf{Q}$, and no permutation matrix $\mathbf{P}$ exists such that $\mathbf{PAQP}^T$ is block upper diagonal [2, Ch.2].

2.1 State of the art heuristics

Birkhoff's original proof [3] of the existence of a BvN decomposition of the form (1) is constructive, and leads to the following family of heuristics. Let $\mathbf{A}^{(0)} = \mathbf{A}$. At every step $i \geq 1$, find a permutation matrix $\mathbf{P}_i \subseteq \mathbf{A}^{(i-1)}$, use the minimum nonzero entry of $\mathbf{A}^{(i-1)}$ at the positions identified by $\mathbf{P}_i$ as α_i, set $\mathbf{A}^{(i)} = \mathbf{A}^{(i-1)} - \alpha_i \mathbf{P}_i$, and repeat the computations in the next step $i+1$ until $\mathbf{A}^{(i)}$ becomes void. The structure of any heuristic of this type is called *generalized Birkhoff heuristic* and is shown in Algorithm 1.

Algorithm 1: Generalized Birkhoff heuristics

Input : $\mathbf{A}$: a doubly stochastic matrix
Output: $\alpha_1, \ldots, \alpha_k$ and $\mathbf{P}_1, \ldots, \mathbf{P}_k$, where $\mathbf{A} = \sum_{i=1}^{k} \alpha_i \mathbf{P}_i$

Let $\mathbf{A}^{(0)} \leftarrow \mathbf{A}$ and $i \leftarrow 1$
while $\mathbf{A}^{(i-1)} \neq 0$ **do**
1   find a permutation matrix $\mathbf{P}_i \subseteq \mathbf{A}^{(i-1)}$
     $\alpha_i \leftarrow \min\{\mathbf{P}_i \odot \mathbf{A}^{(i-1)}\}$
2    $\mathbf{A}^{(i)} \leftarrow \mathbf{A}^{(i-1)} - \alpha_i \mathbf{P}_i$
     $j \leftarrow i+1$
$k \leftarrow i-1$

The heuristics from this family differ by the way in which a permutation matrix is chosen at Step 1 of Algorithm 1. The original argument in Birkhoff's proof selects a permutation matrix $\mathbf{P}_i$ at step i which contains a one at the position of the smallest nonzero entry of $\mathbf{A}^{(i-1)}$. Dufossé and Uçar [10] propose choosing a permutation matrix $\mathbf{P}_i$ where the minimum nonzero entry of $\mathbf{A}^{(i-1)}$ identified by $\mathbf{P}_i$ is maximum—this is the mentioned BvNG heuristic. Experiments reported in the earlier work [10] comparing these two approaches show that BvNG is a very effective in general.

The heuristics from the generalized Birkhoff family have the following distinctive characteristic. At Step 2 of Algorithm 1, they annihilate at least one nonzero from $\mathbf{A}^{(i-1)}$ in obtaining the next matrix iterate $\mathbf{A}^{(i)}$. Based on this observation, Marcus–Ree Theorem [19] states that for a dense matrix, any algorithm from the generalized Birkhoff family will obtain at most $k \leq n^2 - 2n + 2$ permutation matrices. Brualdi and Gibson [5] and Brualdi [4] show that for a sparse, fully indecomposable doubly stochastic matrix with τ nonzeros, the relation $k \leq \tau - 2n + 2$ holds.

Motivated by a question of Brualdi [4], Dufossé et al. [11] show that the mentioned characteristic of the generalized Birkhoff heuristics prevents all algorithms from this family to explore the whole solution space of BvN decompositions. In particular, no algorithm from this family, nor similar algorithms making sequential greedy decisions about coefficients without revising them, can always find optimal solutions, even if they test all available permutations. One class of matrices are constructed [11] by first associating each letter from a to j with 2^p for $p = 0, \ldots, 9$ in the lexicographic order. Then, the matrix

$$\mathbf{A} = \frac{1}{1023} \begin{pmatrix} a+b & d+i & c+h & e+j & f+g \\ e+g & a+c & b+i & d+f & h+j \\ f+j & e+h & d+g & b+c & a+i \\ d+h & b+f & a+j & g+i & c+e \\ c+i & g+j & e+f & a+h & b+d \end{pmatrix}$$

is doubly stochastic, as is

$$\begin{pmatrix} \mathbf{I}_{n\times n} & 0_{n\times 5} \\ 0_{5\times n} & \mathbf{A} \end{pmatrix}, \tag{2}$$

for any $n \geq 0$, where $\mathbf{I}_{n\times n}$ is the $n \times n$ identity matrix and $0_{n\times 5}$ and $0_{n\times 5}$ are matrices of zeros. An optimal decomposition of (2) requires ten permutation matrices corresponding to the ten letters, and any algorithm annihilating an entry in the first iteration without revising the coefficients cannot find it.

2.2 BvN decomposition as a sparse coding problem

Dufossé et al. [11] formulate the BvN decomposition of a given doubly stochastic matrix $\mathbf{A}$ as a linear system of equations. As this formulation underlines our contributions, we review it shorty.

Let Ω_n be the set of all $n \times n$ permutation matrices. The $n!$ matrices in Ω_n are, without loss of generality, ordered and referred to as $\mathbf{P}_1, \ldots, \mathbf{P}_{n!}$. Dufossé et al. define an incidence matrix $\mathbf{M}$ of size $n^2 \times n!$, which encodes the inclusion of each nonzero position of $\mathbf{A}$ in the permutation matrices in Ω_n. Matrix $\mathbf{M}$ is therefore a dictionary of permutations. Consequently, a BvN decomposition with the smallest number of permutation matrices is the following sparse coding problem

$$\min_{\mathbf{x}\in\mathbb{R}_+^{n!}} \|\mathbf{x}\|_0 \text{ such that } \mathbf{M}\mathbf{x} = \mathbf{a}\ , \tag{3}$$

where $\mathbf{a} = \mathrm{vec}(\mathbf{A})$, and $\mathbf{x}$ is a vector of $n!$ nonnegative elements, x_j corresponding to the permutation matrix $\mathbf{P}_j$. The dictionary is dramatically large even for tiny values of $n \geq 11$, making this sparse coding problem very challenging in theory (the dictionary coherence is high) and in practice (the whole dictionary must not be stored).

2.3 Birkhoff heuristics, matching pursuit and Frank-Wolfe algorithms

The generalized Birkhoff family is closely related to the Matching Pursuit (MP) algorithm from the sparse coding literature [18]. MP is a greedy iterative algorithm that finds patterns in a known dictionary that sum up to a given observation. It first chooses a maximally correlated pattern with the residuals $\mathbf{r}^{(i)} = \mathbf{a} - \mathbf{M}\mathbf{x}^{(i)}$ at current iteration i by solving the selection problem $\arg\max_{j\leq n!} |\mathbf{M}_j\mathbf{r}^{(i)}|$, then removes this pattern by projecting the residual on the orthogonal hyperplane, computing coefficients $\mathbf{x}^{(i+1)}$. MP is therefore similar to the generalized Birkhoff heuristics. However the update rule for the coefficients in the MP algorithm may not guarantee that the residuals $\mathbf{a}-\mathbf{M}\mathbf{x}^{(i)}$ are nonnegative. When this happens, the matrix in the next iteration cannot be a (scaled) doubly stochastic matrix and hence the decomposition cannot correspond to (1). We elaborate on this to develop novel algorithms in the next section.

A similar observation is used by Valls et al. [22], who connect the Frank-Wolfe algorithm [14] to the Birkhoff's heuristics. In this context, Frank-Wolfe has the same selection rule as MP, but a different coefficient update method that guarantees that the solution stays in the space of doubly stochastic matrices at all times throughout the iterative algorithm.

2.4 Orthogonal matching pursuit

A significant issue with generalized Birkhoff heuristics, MP or Frank-Wolfe is that they always estimate permutation weights sequentially, leading to suboptimality on instances (2). A solution to this issue, also quickly explored in [22], is to improve upon the coefficients update rules. To this end, we propose to use the framework of Orthogonal Matching Pursuit (OMP). In Algorithm 2, we reproduce the standard OMP algorithm from the literature [12, p. 65], but in the next Section we will part from the usual formulation to derive a class of generalized OMP heuristics for the BvN decomposition. In Step OMP_1 of Algorithm 2, similarly to MP heuristics, we choose the index of the largest absolute value in the vector $\mathbf{M}^T(\mathbf{a}-\mathbf{M}\mathbf{x}^{(i)})$, and add that to the solution

Algorithm 2: Orthogonal matching pursuit

Input : a given matrix $\mathbf{M}$ and a given vector $\mathbf{a}$
Output: the vector $\mathbf{x}$, a sparse approximate solution to $\mathbf{Mx} = \mathbf{a}$

Let $S^{(0)} \leftarrow \emptyset$ and $\mathbf{x}^{(0)} \leftarrow 0$
$i \leftarrow 0$
while *not converged* **do**
OMP$_1$ $\quad S^{(i+1)} \leftarrow S^{(i)} \cup \{j\}$ where $j = \arg\max_j \{|\mathbf{M}_j^T(\mathbf{a} - \mathbf{Mx}^{(i)})|\}$
OMP$_2$ $\quad \mathbf{x}^{(i+1)} \leftarrow \arg\min_{\mathbf{z}} \{\|\mathbf{a} - \mathbf{Mz}\|_2$ where $\operatorname{supp}(\mathbf{z}) \subseteq S^{(i+1)}\}$
$\quad i \leftarrow i + 1$

support S. However in Step OMP$_2$, we now find the best solution vector whose nonzero indices are in the current set S, therefore updating all coefficients at each iteration.

It is worthwhile to note that variants of OMP have been studied rigorously in the literature. Of particular relevance to our work is the Nonnegative OMP heuristic [20], which imposes nonnegativity in Step OMP$_2$ and has similar performances and guarantees as the standard OMP.

2.5 Bipartite matching problems

Permutation matrices correspond to perfect matchings in bipartite graphs, and are useful in our context. Therefore, we give an overview of bipartite matching problems.

A *matching* in a graph is a set of edges no two of which share a common vertex. A vertex is said to be matched if there is an edge in the matching incident on the vertex. In a *perfect matching*, all vertices are matched.

Let $\mathbf{A}$ be an $n \times n$ doubly stochastic matrix. The standard bipartite graph $G = (R \cup C, E)$ associated with $\mathbf{A}$ has n vertices in the set R and another n vertices in the set C such that each $a_{i,j} \neq 0$ uniquely defines an edge $(r_i, c_j) \in E$ in G. For a given matrix $\mathbf{A}$ then, a permutation matrix $\mathbf{P}$ with $\mathbf{P} \subseteq \mathbf{A}$ corresponds to a unique perfect matching in the bipartite graph of $\mathbf{A}$. Because of this connection, we are concerned with perfect matchings in bipartite graphs. The values of the nonzero entries of the matrix can be used as the weight of the corresponding edges in the bipartite graph.

Given the weighted bipartite graph of a doubly stochastic matrix, one can define various perfect matching problems. One problem of interest is the *maximum weighted perfect matching* problem (MWPM). In this problem, the weight of a perfect matching (PM) is defined as the sum of the weights of its edges, and the aim is to find a PM with the maximum weight. A second problem of interest is the *bottleneck perfect matching* problem (BPM). In this problem, the bottleneck value of a PM is defined as the minimum of the weights of its edges, and the aim is to find a perfect matching with the maximum bottleneck value. The BvNG thus solves BPM at each iteration. A comprehensive coverage of algorithms for the mentioned bipartite matching problems can be found in the book by Burkhard et al. [6]. There are efficient solvers for MWPM and BPM [9, 21].

3 GompBvn: Greedy orthogonal matching pursuit based algorithms for BvN decomposition

Here, we propose OMP-based methods to solve (3) in order to obtain effective heuristics for the BvN decomposition. The principal traits of the proposed approach are as follows: (i) it is based on the solid OMP-framework to obtain sparse solutions; (ii) it generalizes and improves the BvNG heuristic [10] in multiple aspects, including obtaining optimum solutions that cannot be computed by any generalized Birkhoff heuristic.

Let us start by examining (3) and Steps OMP_1 and OMP_2 of Algorithm 2. As the matrix $\mathbf{M}$ has $n!$ columns, it cannot be stored when carrying out the Step OMP_1. We will exploit the special structure of $\mathbf{M}$ for this step. As highlighted before, one needs to guarantee that the matrix entries are nonnegative to obtain a BvN decomposition (1). We will present approaches to do so in the OMP framework for OMP_2, to obtain a suitable heuristic for sparse BvN decompositions.

Step OMP_1 of Algorithm 2 finds the column of $\mathbf{M}$ having the largest inner product with the residual $\mathbf{a} - \mathbf{M}\mathbf{x}^{(i)}$. We first realize that computing the residual translates to $\mathbf{A}^{(i+1)} \leftarrow \mathbf{A} - \sum_{j \in S^{(i)}} x_j \mathbf{P}_j$, therefore it can be computed efficiently. We then recognize that $\mathbf{M}^T$ contains n nonzeros in each row, which are all one, and that those nonzeros in a row define a permutation matrix. Therefore $\mathbf{M}^T(\mathbf{a} - \mathbf{M}\mathbf{x}^{(i)})$ translates to computing the value $\sum \mathbf{P} \odot \mathbf{A}^{(i+1)}$ for all permutation matrices $\mathbf{P}$ and storing them in a vector. Obviously, the $\arg\max$ in Step OMP_1 selects the maximum of this vector.

As we have seen in Section 2.5, $\sum \mathbf{P} \odot \mathbf{A}^{(i+1)}$ is the weight of the perfect matching associated with $\mathbf{P}$ in the bipartite graph of $\mathbf{A}^{(i+1)}$. Therefore, Step OMP_1 of Algorithm 2 can be solved by finding a MWPM in the bipartite graph of $\mathbf{A}^{(i+1)}$.

Step OMP_2 of Algorithm 2 solves a least squares (LS) problem which is efficiently solvable by off-the-shelf methods. As a solution to an LS problem can contain both negative and positive components, and the BvN decomposition asks for positive only, we should solve a nonnegative LS problem here, as in nonnnegative OMP [20]. Furthermore, one needs to ensure that the residual is nonnegative, that is $\mathbf{a} - \mathbf{M}\mathbf{x}^{(i)} \geq 0$, at the ith iteration. This is necessary for continuing with finding perfect matchings, and also for making sure that the remaining matrix when multiplied with $\frac{1}{\sum_{j=1}^{i} \alpha_j}$ is doubly stochastic. Therefore, Step OMP_2 is reformulated as

$$\underset{\mathbf{z} \geq 0,\ \mathbf{a} - \mathbf{M}\mathbf{z} \geq 0}{\arg\min} \{\|\mathbf{a} - \mathbf{M}\mathbf{z}\|_2 \text{ where } \operatorname{supp}(\mathbf{z}) \subseteq S^{(i+1)}\} . \tag{QP}$$

With this formulation in Step OMP_2, we have thus a heuristic based on the OMP methodology to find a BvN decomposition of a given doubly stochastic matrix. Note that (QP) can be solved via quadratic program solvers.

While the above development follows the OMP-framework, we need to part from this strict interpretation. This is so, as the additional conditions breaks the original design of OMP where the problem solved by Steps OMP_1 and OMP_2 are derived from the same cost and constraints: Step OMP_2 fixes the solution support, while Step OMP_1 finds a single entry in the solution, which yield favorable properties on the reduction of the cost function [12, Lemma 3.3]. However by introducing the nonnegative residuals constraints, the cost function in Step OMP_2 can only use a scaled down version of the inner product between the residual and the dictionary to retain nonnegativity. We therefore propose another loss function for which the two OMP steps are coherent.

Let $\mathbf{z}$ be a feasible vector, that is $\mathbf{z} \geq 0$ and $\mathbf{a} - \mathbf{M}\mathbf{z} \geq 0$. Then, $\|\mathbf{z}\|_1 = \sum z_i$, and $\|\mathbf{a} - \mathbf{M}\mathbf{z}\|_1 = n(1 - \sum z_i)$. This is so, as each row/column of $\mathbf{A}$ adds up to 1 and $\mathbf{a} - \mathbf{M}\mathbf{z}$ translates to subtracting a total of $\sum z_i$ from each row/column using the permutation matrices identified by the support of $\mathbf{z}$. Minimizing the quantity $1 - \sum z_i$, or maximizing $\sum_i z_i$, in Step OMP_2 therefore will minimize an upper bound on $\|\mathbf{a} - \mathbf{M}\mathbf{z}\|_2$, since $\|\mathbf{v}\|_1 \geq \|\mathbf{v}\|_2$ for any vector $\mathbf{v}$. Once we choose $\sum z_i$ as the objective function, we should modify Step OMP_1 to be relevant to this objective function. Let $\mathbf{x}^{(i)}$ be the current solution where $\operatorname{supp}(\mathbf{x}^{(i)}) \subseteq S^{(i)}$, and j be the permutation matrix selected in Step OMP_1. Since the vector $\mathbf{z}' = [\mathbf{x}^{(i)}, z'_j]$ for $0 \leq z'_j \leq \min\{\operatorname{vec}(\mathbf{P}_j) \odot \mathbf{a} - \mathbf{M}\mathbf{x}^{(i)}\}$ is supported by $S^{(i)} \cup \{j\}$, and it holds that $\mathbf{z}' \geq 0$, and

$\mathbf{a}-\mathbf{M}\mathbf{z}' \geq 0$, choosing j which maximizes $\min\{\text{vec}(\mathbf{P}_j) \odot \mathbf{a} - \mathbf{M}\mathbf{x}^{(i)}\}$ guarantees an improvement of at least $\min\{\text{vec}(\mathbf{P}_j) \odot \mathbf{a} - \mathbf{M}\mathbf{x}^{(i)}\}$ in the objective function $\sum z_i$. We thus propose using

$$\underset{\mathbf{z}\geq 0,\ \mathbf{a}-\mathbf{M}\mathbf{z}\geq 0}{\arg\max} \{\|\mathbf{z}\|_1 \text{ where } \text{supp}(\mathbf{z}) \subseteq S^{(i+1)}\} \tag{LP}$$

as the objective function in Step OMP_2, with a BPM solver for Step OMP_1. As its name indicate, (LP) is a linear program for which very efficient solvers exist.

We use GompBvN(BPM,LP) to denote the proposed OMP-based solver, where the first parameter designates the problem solved in Step OMP_1, and the second one designates the problem solved in Step OMP_2. The strict OMP-inspired method is similarly referred to as GompBvN(MWPM,QP). Obviously, the other combinations GompBvN(BPM,QP) and GompBvN(MWPM,LP) are possible.

4 Experiments

We present a selection of experimental results to compare the proposed GompBvN heuristics with each other and with the state of the art. We have implemented all algorithms in Python, used Gurobi [13] for (LP) and (QP), MC64 [9] for MWPM, and Bottled [21] for BPM.

We use matrices from the SuiteSparse Matrix collection [8] that arise in diverse applications. As this collection has many matrices, we automatically selected those with the following properties: square, between 500 and 1000 rows, fully indecomposable, and with at most 50 nonzeros per row. This selection yields 58 matrices from 11 different groups. We retained up to two matrices per group to remove any bias that might be arising from the group. This resulted in 18 matrices given in Table 2. (The name dynamicSoaringProblem_1 is shortened.) In this table, the column n lists the number of rows, and the column τ gives the number of nonzeros for each matrix. The matrices are preprocessed by taking the absolute values of their nonzeros and scaling them to be doubly stochastic by the method by Knight and Ruiz [15], where the maximum deviation of the row and column sums from 1 were less than 3.27E-7. We run different algorithms until the coefficients obtained add up to at least 0.999.

In many problem instances, GompBvN(MWPM, QP) obtained a larger number of permutation matrices with much more computing time than GompBvN(BPM, LP). Therefore, we give only a few results with GompBvN(MWPM, QP) for the sake of completeness. For example on matrices 662_bus and EX1, GompBvN(MWPM, QP) obtained 116 and 90 permutation matrices, which are larger than other numbers given in Table 2. On some matrices, for example on bcsstk34, the run time was large. On this matrix, GompBvN(BPM, LP) runs in less than 20 seconds on a laptop with 2,5 GHz Intel Core i7 with 16GB memory and obtains 118 permutation matrices, but GompBvN(MWPM, LP) does not deliver a result within 10 minutes (we have stopped it at 81st iteration).

We first note that all four variants of GompBvN solve the special $(n+5) \times (n+5)$ case (2) optimally with 10 permutation matrices. This outcome clearly separates the GompBvN solvers from the generalized Birkhoff heuristics, in particular from BvNG which finds 12 permutation matrices. As BvNG and GompBvN(BPM, $\cdot$) apply the same algorithm for selecting the permutation matrices, their first selection is the same. Step OMP_2 then helps GompBvN(BPM, $\cdot$), to revise the coefficients to deliver optimal results.

Next, another family of matrices are constructed to give further insights into the GompBvN's performance. This set of matrices are created by two parameters n, k. First, an $n \times n$ permutation matrix $\mathbf{P}$ is created. Then, another k permutation matrices are created, each of which has distinct n/k common elements with $\mathbf{P}$, while having random permutations for the remaining $n - n/k$ elements. $\mathbf{P}$ is multiplied by 2^k, and other permutation matrices with 2^p, for $p =$

Table 1: Performance of heuristics on a set of constructed matrices

		GompBvN		GompBvN	
(n,k)	BvNG	(BPM,LP)	(BPM,QP)	(MWPM,LP)	(MWPM,QP)
(100,10)	19	11	11	11	11
(200,15)	29	16	16	16	16
(500,20)	39	21	21	21	21

Table 2: Performance of heuristics on a set of matrices from the SuiteSparse matrix collection

Matrix				GompBvN	
Name	n	τ	BvNG	(BPM,LP)	(BPM,QP)
EX1	560	8736	56	56	64
EX2	560	8736	67	68	70
cdde1	961	4681	22	19	19
cdde2	961	4681	23	16	18
bcsstk34	588	21418	120	118	119
bcsstm34	588	24270	174	174	170
ex21	656	18964	166	165	168
Trefethen_500	500	8478	69	69	69
ex22	839	22460	173	171	177
L	956	3640	59	58	58
ch5-5-b3	600	2400	4	4	4
dynamicSoaringP_1	647	5367	305	316	312
685_bus	685	3249	41	40	40
662_bus	662	2474	35	34	34
spaceShuttleEntry_1	560	6891	274	280	259
Trefethen_700	700	12654	73	73	73
netz4504_dual	615	2342	30	28	28
Si2	769	17801	87	86	87

$\{0, \dots, k-2\}$ in arbitrary order, and all $k+1$ scaled permutation matrices are added to produce a doubly stochastic matrix after a straightforward normalization. By construction, a matrix from the (n,k)-family can be decomposed with $k+1$ permutation matrices. BvNG will find $\mathbf{P}$ in the first step and will use the minimum weight 2^k+1 (before normalization) and thus cannot find the said decomposition. The same is true of a variant of BvNG which uses MWPM for selecting permutations. We compare BvNG and the four variants of GompBvN in Table 1 for three (n,k) pairs. As seen from this table, GompBvN variants find decompositions with $k+1$ permutation matrices, but BvNG obtains nearly twice larger number of permutation matrices. These experiments thus show that the proposed methods improve the state of the art in hard cases.

Last, we observe from Table 2 that in cases where BvNG is effective so is GompBvN(BPM, $\cdot$). This is of course not a surprise, as at a given iteration the solution space in Step OMP$_2$ of GompBvN(BPM, $\cdot$) includes the coefficients found by BvNG, should they both contain the same set of permutation matrices. The near identical solution quality between BvNG and GompBvN(BPM, $\cdot$) thus not only confirms that GompBvN(BPM, $\cdot$) is effective but also sheds light on the good performance of BvNG.

5 Conclusion

This paper proposes a set of heuristics for obtaining a sparse Birkhoff-von Neumann (BvN) decomposition of doubly stochastic matrices. The proposed heuristics are based on the well-known greedy orthogonal matching pursuit (OMP) algorithm, and advances the state of the art in the BvN decomposition problem. Experimental results show that the proposed heuristics overcome the innate limitation of the state of the art approaches in the literature and are competitive with them on general instances. There are a number of follow-up questions: (i) what other selection and optimization approaches can be used inside the OMP framework; (ii) what are the limitations of the proposed heuristics; (iii) what is the best way to hybridize the existing BvNG and the proposed heuristics.

Acknowledgment

J. E. Cohen is supported in part by ANR JCJC project LoRAiA ANR-20-CE23-0010.

References

[1] M. Benzi and B. Uçar. Preconditioning Techniques Based on the Birkhoff–von Neumann Decomposition. *Computational Methods in Applied Mathematics*, Dec. 2016.

[2] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences.* SIAM, Philadelphia, PA, USA, 1994.

[3] G. Birkhoff. Tres observaciones sobre el algebra lineal. *Univ. Nac. Tucumán Rev. Ser. A*, (5):147–150, 1946.

[4] R. A. Brualdi. Notes on the Birkhoff algorithm for doubly stochastic matrices. *Can. Math. Bulletin*, 25(2):191–199, 1982.

[5] 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(2): 194–230, 1977.

[6] R. Burkard, M. Dell'Amico, and S. Martello. *Assignment Problems.* SIAM, Philadelphia, PA, USA, 2009.

[7] C.-S. Chang, W.-J. Chen, and H.-Y. Huang. On service guarantees for input-buffered crossbar switches: A capacity decomposition approach by birkhoff and von neumann. In *Quality of Service, 1999. IWQoS '99. 1999 Seventh International Workshop on*, pages 79–86, 1999.

[8] T. A. Davis and Y. Hu. The University of Florida sparse matrix collection. *ACM Trans. Math. Softw.*, 38(1):1:1–1:25, 2011.

[9] 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*, 22:973–996, 2001.

[10] F. Dufossé and B. Uçar. Notes on Birkhoff–von Neumann decomposition of doubly stochastic matrices. *Linear Algebra and its Applications*, 497:108–115, 2016. ISSN 0024-3795. doi: http://dx.doi.org/10.1016/j.laa.2016.02.023.

[11] F. Dufossé, K. Kaya, I. Panagiotas, and B. Uçar. Further notes on Birkhoff–von Neumann decomposition of doubly stochastic matrices. *Linear Algebra and its Applications*, 554:68–78, 2018. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2018.05.017.

[12] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Birkhäuser Basel, 2013. ISBN 0817649476.

[13] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL `https://www.gurobi.com`.

[14] M. Jaggi. Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization. In *30th ICML*, pages 427–435. PMLR, Feb. 2013.

[15] P. A. Knight and D. Ruiz. A fast algorithm for matrix balancing. *IMA Journal of Numerical Analysis*, 33(3):1029–1047, 2013.

[16] J. Kulkarni, E. Lee, and M. Singh. Minimum Birkhoff-von Neumann decomposition. preliminary version which is available at `http://www.cs.cmu.edu/~euiwoonl/sparsebvn.pdf`, June 2017.

[17] L. Liu, J. J. Xu, and L. Fortnow. Quantized BvND: A better solution for optical and hybrid switching in data center networks. In *2018 IEEE/ACM 11th International Conference on Utility and Cloud Computing (UCC)*, pages 237–246, Dec 2018.

[18] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[19] M. Marcus and R. Ree. Diagonals of doubly stochastic matrices. *The Quarterly Journal of Mathematics*, 10(1):296–302, 1959.

[20] T. T. Nguyen, C. Soussen, J. Idier, and E.-H. Djermoune. Exact recovery analysis of non-negative orthogonal matching pursuit. In *SPARS*, 2019.

[21] I. Panagiotas, G. Pichon, S. Singh, and B. Uçar. Engineering fast algorithms for the bottleneck matching problem. In *The 31st Annual European Symposium on Algorithms*, Amsterdam, Netherlands, 2023.

[22] V. Valls, G. Iosifidis, and L. Tassiulas. Birkhoff's decomposition revisited: Sparse scheduling for high-speed circuit switches. *IEEE/ACM Transactions on Networking*, 29(6):2399–2412, 2021. doi: 10.1109/TNET.2021.3088327.

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