

# Algorithms for symmetric Birkhoff-von Neumann decomposition of symmetric doubly stochastic matrices

Damien Lesens  
joint work with Bora Uçar and Jérémy E. Cohen

ROMA team, LIP, ENS de Lyon

March 16, 2026

# Plan

1. Background
2. Generalisation of the BvN decomposition
3. Existing work and contributions
4. Overview of the algorithm
5. Conclusion

## Definition

An  $n \times n$  matrix  $A$  is **doubly stochastic** if  $a_{ij} \geq 0$ , row sums and column sums are 1

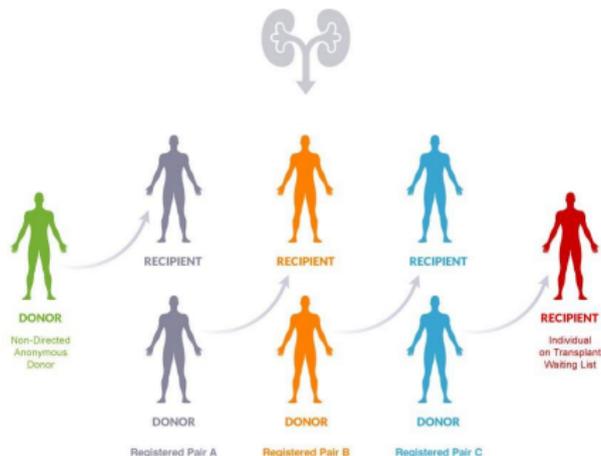
## Theorem (Birkhoff, von Neumann)

For a doubly stochastic matrix  $\mathbf{A}$ , there exist  $\alpha_1, \alpha_2, \dots, \alpha_k \in (0, 1]$  with  $\sum_{i=1}^k \alpha_i = 1$  and  $n \times n$  permutation matrices  $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_k$  such that

$$\mathbf{A} = \alpha_1 \mathbf{P}_1 + \alpha_2 \mathbf{P}_2 + \dots + \alpha_k \mathbf{P}_k$$

# Applications

- ▶ Most matrices can be scaled to be doubly-stochastic  
→ numerical linear algebra applications
- ▶ Routing traffic in data centers (circuit switches)
- ▶ Assignment problems and economics (e.g. kidney exchange problem)



Picture source: <https://livablesoftware.com/open-source-matching-market/>

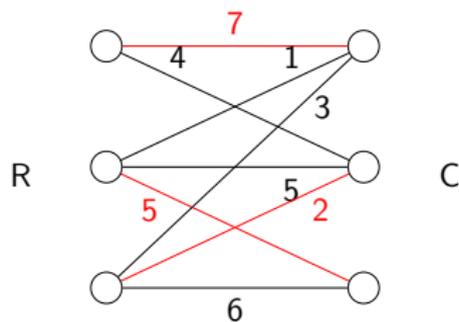
# Algorithm

- ▶ represent a doubly stochastic matrix as a bipartite graph
- ▶ Algorithm: *by Birkhoff*
  - ▶ find a perfect matching in this graph
  - ▶ the coefficient is the minimum entry in the permutation
  - ▶ update and continue until all entries are zero

$$R \begin{matrix} & C \\ \begin{pmatrix} 7 & 4 & 0 \\ 1 & 5 & 5 \\ 3 & 2 & 6 \end{pmatrix} \end{matrix}$$

$$= 2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 5 & 4 & 0 \\ 1 & 5 & 3 \\ 3 & 0 & 6 \end{pmatrix}$$

(a) Matrix representation



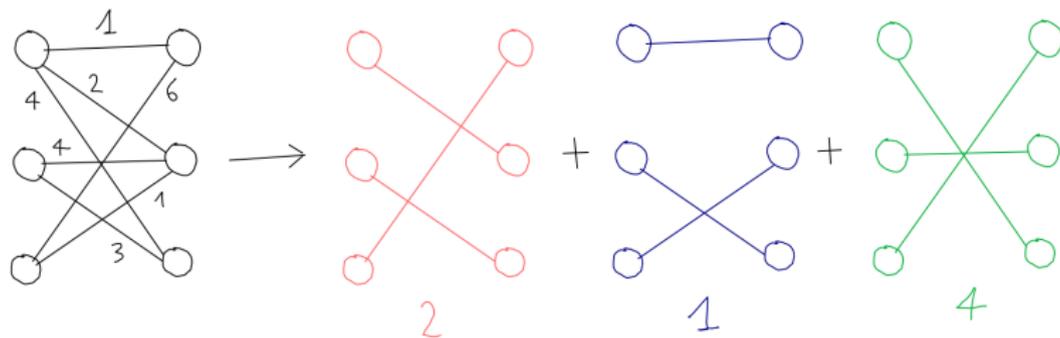
(b) Graph representation

# Generalisation of the BvN decomposition

1. Background
2. Generalisation of the BvN decomposition
3. Existing work and contributions
4. Overview of the algorithm
5. Conclusion

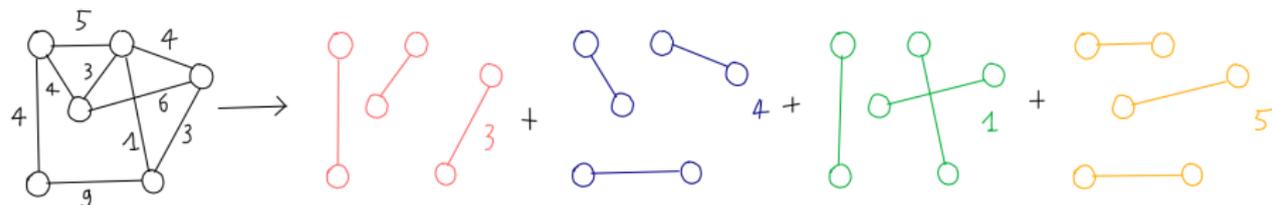
# "Classical" BvN decomposition

Given **bipartite graph** and a weighting in the convex hull of its perfect matchings,  
→ find a decomposition



# Generalisation of the BvN decomposition

Given **GENERAL graph** and a weighting in the convex hull of its perfect matchings,  
→ find a decomposition



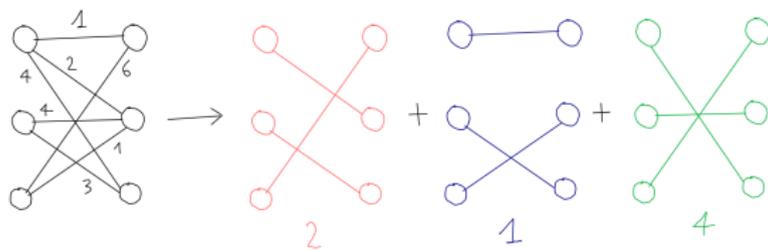
Decomposition on general graphs extends all applications of the bipartite case (non bipartite matching markets...)

# Graph formulation

For a bipartite graph  $G = (R \cup C, E)$ , the Birkhoff theorem states that the polytope

$$\sum_{e \in \delta(v)} x_e = 1 \quad \text{for all } v \in V = R \cup C$$
$$x_e \geq 0 \quad \text{for all } e \in E$$

is exactly the polytope whose vertices are the perfect matchings in  $G$ .



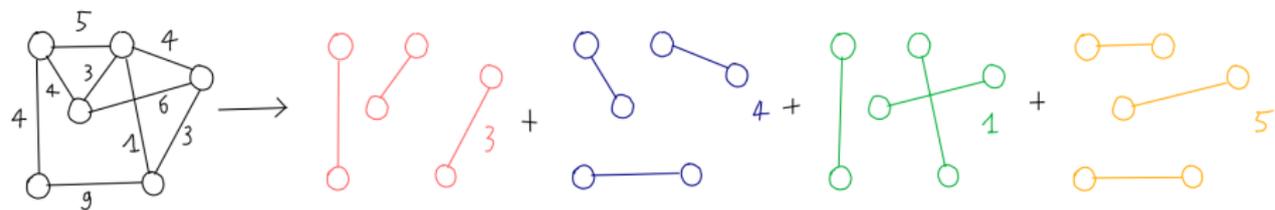
# Generalisation of the BvN decomposition

Edmonds ('65) showed that the convex hull of all perfect matchings in a general graph  $G = (V, E)$ , with an even number of vertices, is defined by the polytope

$$\sum_{e \in \delta(v)} x_e = 1 \quad \text{for all } v \in V$$

$$\sum_{e \in \delta(S)} x_e \geq 1 \quad \text{for all odd set } S \subset V$$

$$x_e \geq 0 \quad \text{for all } e \in E.$$



# Existing work and contributions

- ▶ V. Vazirani (2020) gives an algorithm
  - ▶ more a proof than the problem is in  $\mathcal{P}$  rather than an algorithm
  - ▶ very costly, some minor bugs, not implementable easily
- ▶ Padberg and Wolsey (1984) already provided an algorithm in a more general framework
  - ▶ their paper is mostly focused on forest decomposition
  - ▶ outdated notation and formalism

Our contribution:

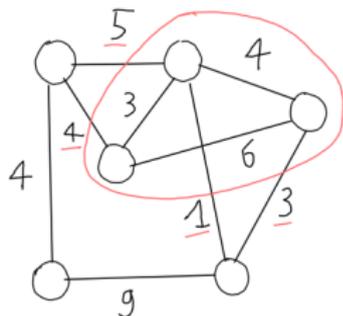
- ▶ we take the better from the two by providing a clear and detailed algorithm in a modern formalism
- ▶ minor complexity improvement
- ▶ heuristic speed up by maximizing the chosen weight at each iteration
- ▶ error analysis for quasi doubly stochastic graph/matrices
- ▶ C++ implementation and experiments

- ▶ Polytope has odd cut constraints

$$\sum_{e \in \delta(v)} x_e = 1 \quad \text{for all } v \in V$$

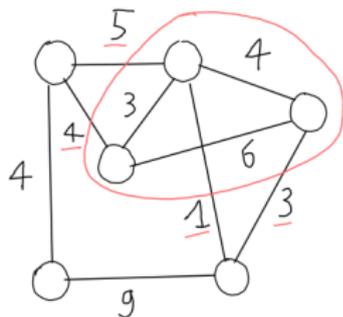
$$\sum_{e \in \delta(S)} x_e \geq 1 \quad \text{for all odd set } S \subset V$$

$$x_e \geq 0 \quad \text{for all } e \in E.$$



- ▶ One can check that a graph is in the polytope with a minimum odd cut computation (Padberg and Rao '82)
- ▶ Incremental algorithm: starts with 1's in the polytope, then  $1 - \alpha_1$ ,  $1 - \alpha_1 - \alpha_2$  and so on until 0

$$\sum_{e \in \delta(S)} x_e \geq 1 \quad \text{for all odd set } S \subset V$$

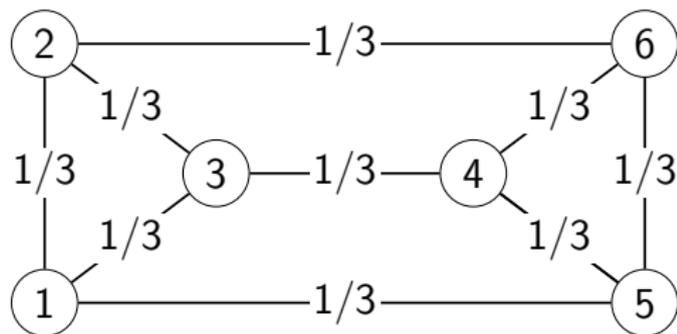


The algorithm is much more intricate than for the bipartite case

- ▶ We cannot choose any matching
  - a matching should cross every tight odd cut exactly once
- ▶ We cannot use the minimum edge weight as a coefficient
  - sometimes we have to choose a smaller coefficient

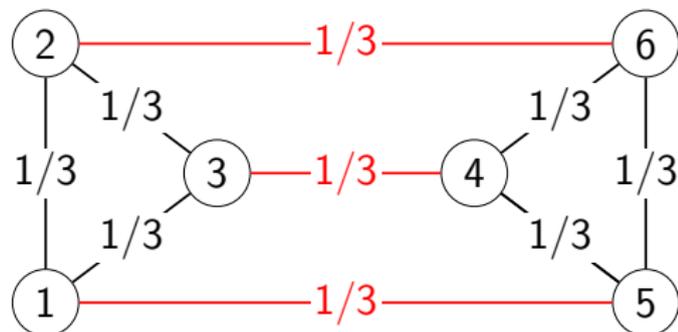
# Example

$$\mathbf{A}_2 = \begin{bmatrix} & 1/3 & 1/3 & & 1/3 \\ 1/3 & & 1/3 & & \\ 1/3 & 1/3 & & 1/3 & \\ & & 1/3 & & 1/3 \\ 1/3 & & & 1/3 & 1/3 \\ & 1/3 & & 1/3 & 1/3 \end{bmatrix}$$



# Example

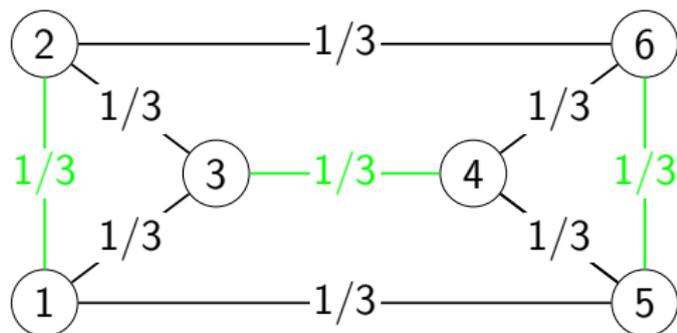
$$\mathbf{A}_2 = \begin{bmatrix} & 1/3 & 1/3 & & 1/3 & \\ 1/3 & & 1/3 & & & 1/3 \\ 1/3 & 1/3 & & 1/3 & & \\ & & 1/3 & & 1/3 & 1/3 \\ 1/3 & & & 1/3 & 1/3 & \\ & 1/3 & & 1/3 & 1/3 & \end{bmatrix}$$



- ▶ We cannot pick the **red** matching, because we would be left with two triangles  $\rightarrow$  not decomposable

# Example

$$\mathbf{A}_2 = \begin{bmatrix} & 1/3 & 1/3 & & 1/3 \\ 1/3 & & 1/3 & & \\ 1/3 & 1/3 & & 1/3 & \\ & & 1/3 & & 1/3 \\ 1/3 & & & 1/3 & 1/3 \\ & 1/3 & & 1/3 & 1/3 \end{bmatrix}$$



- ▶ We cannot pick the **red** matching, because we would be left with two triangles  $\rightarrow$  not decomposable
- ▶ The **green** matching respects all odd cut constraints

# Experiments

- ▶ Algorithm can be used to decompose *symmetric doubly stochastic matrices* as a sum of *symmetric permutation matrices* by interpreting them as the adjacency matrix of a graph
- ▶ Running the algorithm on matrices from the Suite Sparse collection  
→ Can produce less terms than classical BvN decomposition

Matrix name	V	E	Number of terms	
			BvN	symBvN
662.bus	662	2474	34	40
bcsstm34	588	24270	171	159
Si2	769	17801	86	93
EX1	560	8736	57	56
Trefethen_500	500	8478	69	70
L	956	3640	57	138
dynamicSoaringProblem_1	647	5367	319	302
spaceShuttleEntry_1	560	6891	258	292
dendrimer	730	63024	567	486

# Conclusion

- ▶ Provide the first implementation for the generalized Birkhoff von-Neumann decomposition
- ▶ Can be used to write a symmetric doubly-stochastic matrix as a sum of symmetric permutation matrices
- ▶ Paper accepted to SIAM Journal on Matrix Analysis and Applications

- ▶ Provide the first implementation for the generalized Birkhoff von-Neumann decomposition
- ▶ Can be used to write a symmetric doubly-stochastic matrix as a sum of symmetric permutation matrices
- ▶ Paper accepted to SIAM Journal on Matrix Analysis and Applications

Thank you for your attention

# Tight odd cuts

Odd sets  $S$  such that  $\sum_{e \in \delta(S)} x_e = 1$  are annoying for two reasons.

We will call them tight odd cuts.

- ▶ First reason: a perfect matching in the decomposition must cross each tight cut exactly once

We want to go from  $\sum_{e \in \delta(S)} x_e = 1$  to  $\sum_{e \in \delta(S)} x'_e = 1 - \beta$

But for all  $\beta > 0$  and  $k \geq 3$ ,  $1 - k\beta < 1 - \beta$

# Tight odd cuts

- ▶ The coefficient is not always the minimum edge weight in the matching (call it  $\beta_{\max}$ ):

$$\text{Even if } \sum_{e \in \delta(S)} x_e > 1$$

$$\text{It might happen that } \sum_{e \in \delta(S)} x_e - k\beta_{\max} < 1 - \beta_{\max}$$

There is potentially an exponential number of those cuts

→ we cannot check them all.

We maintain a collection of tight odd cuts  $\mathcal{H}$  (starting with  $\mathcal{H} = \emptyset$ )

While there are edges in the graph

- ▶ Find a matching  $M$  crossing all tight cuts in  $\mathcal{H}$  exactly once (with minimum weight perfect matching algorithm)
- ▶ Compute a valid coefficient  $\beta$ :
  - ▶ if  $\beta = 0$  :  $M$  is not valid because it crosses a tight odd cut more than once.

We know one of these cut by the coefficient finding algorithm  
→ add it to  $\mathcal{H}$

- ▶ if  $\beta > 0$ : add  $\beta \cdot M$  to the decomposition

# Example

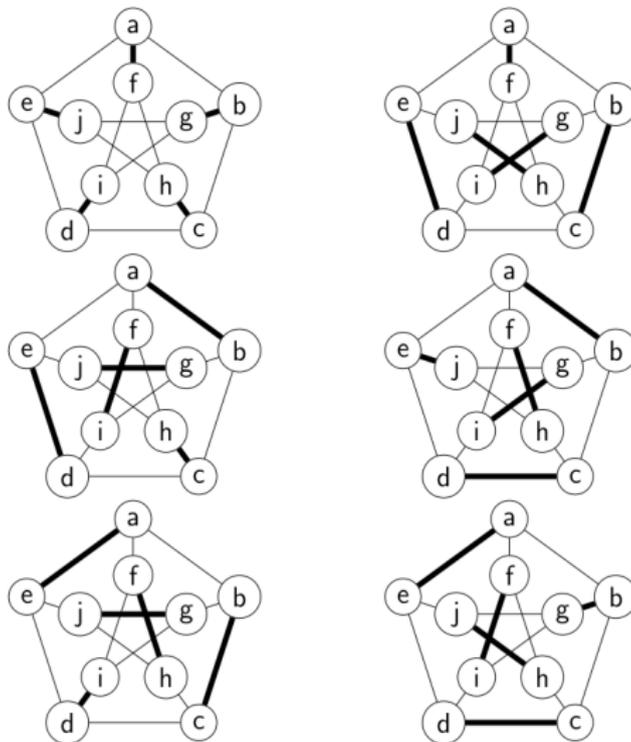


Fig. 2: The Petersen graph with edge weights  $1/3$  and its decomposition into six perfect matchings. The matching edges are shown with thick lines. Each edge is used in two perfect matchings in the decomposition, and the coefficient of each term in the decomposition is  $1/6$ .